Grażyna Koba

TERAZBAJTY

Informatyka

dla szkół ponadpodstawowych

Zakres podstawowy

Klasa

Oznaczenie nieobowiązkowego materiału w związku z usunięciem odpowiednich treści z podstawy programowej zgodnie z rozporządzeniem MEN z 28.06.2024 r.

MiGra

Współpraca autorska:
Katarzyna Koba-Gołaszewska
Marta Skała-Kowalczyk
Michał Łętowski
Piotr Babij
Bolesław Gremblewski

Redakcja i korekta: dr Halina Kubicka

Opracowanie graficzne: Roman Jankowski, Grażyna Koba

Projekt okładki, layout i skład: Studio Grafiki INCOLOR Roman Jankowski

Cyfrowe odzwierciedlenie podręcznika dopuszczonego do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisanego do wykazu podręczników przeznaczonych do kształcenia ogólnego do nauczania informatyki, na podstawie opinii rzeczoznawców: prof. dr hab. Liliany Byczkowskiej-Lipińskiej, dr. Leszka Rudaka, dr Ewy Kozioł-Chrzanowskiej.

Zakres kształcenia: podstawowy.
Etap edukacyjny: III.
Typ szkoły: liceum ogólnokształcące i technikum.
Rok dopuszczenia: 2019.

Numer ewidencyjny w wykazie: 1042/1/2019

Zastrzeżonych nazw firm i produktów użyto w tej książce wyłącznie w celu identyfikacji.

ISBN 978-83-964794-6-4

Copyright © by MIGRA Sp. z o.o., Wrocław 2024

Wydanie I

Fotografie: Michał Koba, Marcin Mytych; Joan Cros/NurPhoto via Getty Images – str. 30 (rys. 8.); bombuscreative/Getty Images – str. 30 (rys. 9.).

Wydawnictwo: MIGRA Sp. z o.o.
ul. Świeradowska 51/57
50-559 Wrocław
tel.: 71 75 06 230
faks: 71 75 06 235
e-mail: biuro@migra.pl

Spis treści

Moduł A Wokół komputera 7

Temat A1. Komputer i urządzenia peryferyjne 8
1. Wyjaśnienie podstawowych pojęć 8
2. Klasyfikacja środków technologii informacyjnej 10
3. Jakie parametry komputera warto znać? 11
   3.1. Co można zamontować na płycie głównej? 11
   3.2. Jakie są parametry dysku komputera? 15
   3.3. Jaki wybrać monitor? 17
4. Dodatkowe urządzenia pamięci masowej 17
   4.1. Napędy optyczne 17
   4.2. Pamięci flash 18
   4.3. Pamięci taśmowe (streamery) 18
5. Wybieranie parametrów komputera 18
6. Przykłady urządzeń peryferyjnych 19

Temat A2. Systemy operacyjne i inne oprogramowanie 23
1. Początki systemów operacyjnych 23
2. Funkcje i struktura systemu operacyjnego 24
3. Wybrane systemy operacyjne 25
   3.1. MS DOS 25
   3.2. Microsoft Windows 25
   3.3. Mac OS/macOS 28
   3.4. Unix 29
   3.5. Linux 29
   3.6. Systemy operacyjne na urządzenia mobilne 30
4. Oprogramowanie użytkowe, narzędziowe i języki programowania 31
5. Przykładowe typy plików 33

Moduł B Wokół dokumentów komputerowych 35

Temat B1. Opracowywanie dokumentów tekstowych o rozbudowanej strukturze 36
1. Nagłówek i stopka dokumentu 36
2. Style tekstu 37
   2.1. Style standardowe 37
   2.2. Style niestandardowe 39
3. Tworzenie konspektu dokumentu 42
4. Odwołania w dokumencie tekstowym 43
   4.1. Spis treści 44
   4.2. Spis ilustracji, tabel i wykresów 45
   4.3. Przypisy 47

Temat B2. Dzielenie dokumentu tekstowego i praca w trybie recenzji 50
1. Podział dokumentu tekstowego na strony, sekcje i kolumny 50
   1.1. Podział dokumentu na strony 50
   1.2. Podział dokumentu na sekcje 51
   1.3. Rozmieszczanie tekstu w kolumnach 52
2. Szablony dokumentów 52

3. Praca w trybie recenzji 53
3.1. Śledzenie zmian 54
3.2. Wstawianie komentarzy 56
3.3. Porównywanie dokumentów 57
4. Praktyczny poradnik 57

Temat B3. Opracowywanie grafiki rastrowej 64
1. Rodzaje grafiki komputerowej 64
2. Formaty plików graficznych 65
2.1. Formaty plików grafiki rastrowej 65
2.2. Formaty plików grafiki wektorowej 67
3. Tworzenie obrazu w programie GIMP 67
4. Praca z warstwami w programie GIMP 70
5. Stosowanie narzędzi selekcji w programie GIMP 72
6. Edycja zdjęć w programie GIMP 76
6.1. Wybrane filtry i efekty 76
6.2. Zmiana kontrastu i nasycenia kolorów 77
6.3. Kadrowanie i skalowanie 77

Temat B4. Zasady tworzenia prezentacji multimedialnej 81
1. Typy prezentacji multimedialnych 81
2. Etapy przygotowywania prezentacji multimedialnej 83
2.1. Planowanie prezentacji 85
2.2. Tworzenie prezentacji 86
2.3. Dodawanie dźwięków i filmów 88
2.4. Zapisywanie prezentacji 90
2.5. Drukowanie materiałów informacyjnych i notatek 92
2.6. Ustalenie parametrów pokazu i pokaz 93
3. Tworzymy prezentację multimedialną o Fryderyku Chopinie 94
4. Importowanie konspektu prezentacji z dokumentu tekstowego 97

Moduł C Wokół algorytmiki i programowania 99

Temat C1. Wprowadzenie do programowania i rozwiązywania problemów z wykorzystaniem komputera 100
1. Sytuacje problemowe 100
2. Określanie specyfikacji zadania (problemu) 103
3. Języki programowania 106
4. Na czym polega programowanie? 107

Temat C2. Tworzenie programów w języku C++ 111
1. Środowisko programistyczne języka C++ 111
1.1. Struktura programu w języku C++ 112
1.2. Etapy tworzenia programu w języku C++ 112
2. Stosowanie zmiennych 115
2.1. Deklarowanie zmiennych 116
2.2. Nadawanie zmiennym wartości 117
2.3. Wykonywanie obliczeń 118
3. Wyprowadzanie komunikatów i wyników na ekran monitora 119
4. Zapisywanie rozwiązania problemu w języku C++ 122

Temat C3. Tworzenie programów w języku Python 126
1. Środowisko programistyczne języka Python 126
1.1. Tryb interaktywny i skryptowy 127
1.2. Pisanie programu w trybie skryptowym 128

2. Stosowanie zmiennych 131
2.1. Nazywanie zmiennych 131
2.2. Nadawanie zmiennym wartości 132
2.3. Wykonywanie obliczeń 133
3. Wyprowadzanie komunikatów i wyników na ekran monitora 134
4. Zapisywanie rozwiązania problemu w języku Python 137

Temat C4. Stosowanie instrukcji warunkowych w językach C++ i Python 141
1. Sytuacje warunkowe 141
2. Algorytmy z warunkami w językach C++ i Python 142
2.1. Proste warunki logiczne 142
2.2. Złożone warunki logiczne 145
3. Sprawdzanie poprawności danych 148
4. Algorytmy z warunkami zagnieżdżonymi 150

Temat C5. Stosowanie instrukcji iteracyjnych w językach C++ i Python 155
1. Iteracja – podstawowa technika algorytmiczna 155
2. Stosowanie iteracji do sumowania liczb 156
3. Algorytmy iteracyjne w językach C++ i Python 158
3.1. Instrukcja iteracyjna `for` w języku C++ 158
3.2. Instrukcja iteracyjna `for` w języku Python 159
3.3. Zapisywanie rozwiązania problemu iteracyjnego w językach C++ i Python 161
4. Instrukcje iteracyjne zagnieżdżone 162

Moduł D Wokół Internetu i projektów 167

Temat D1. Internet jako ocean informacji 168
1. Piramida rozwoju Internetu 168
2. Organizacja informacji w WWW 171
3. Przykłady wyszukiwania informacji 173
3.1. Korzystamy z encyklopedii i słowników 173
3.2. Wyszukujemy informacje zapisane w innych językach 173
3.3. Szukamy szybciej i sprawniej 176
3.4. Oglądamy mapy i podróżujemy z Internetem 177
4. Praktyczny poradnik 179

Temat D2. Korzystanie z wybranych e-usług 184
1. E-nauczanie 184
2. E-dziennik 185
3. E-praca 186
4. E-bank 187
5. E-zakupy 191
6. E-aukcje 193
7. E-podpis 194

Temat D3. Zadania projektowe 197
1. Przygotowanie projektu 197
1.1. Etapy tworzenia projektu grupowego 197
1.2. Debata ZA i PRZECIW 198
2. Zadania projektowe 199

Oznaczenia przyjęte w podręczniku



Warto powtórzyć – wykaz zagadnień do powtórzenia z poprzednich lekcji



Metoda – opis istotnych metod, pojęć i zasad



Sposób postępowania – sposób postępowania prowadzący do osiągnięcia określonego celu



Przykłady – pokazanie przykładowego rozwiązania



Ćwiczenia – praktyczne sprawdzenie i utrwalenie poznawanych zagadnień



Słownik – wyjaśnienie ważnych terminów



Czy wiesz, że – dodatkowe informacje na temat zagadnień dotyczących informatyki

Na końcu tematów znajdują się działy:



Warto zapamiętać – zestawienie najważniejszych informacji podsumowujących temat



Pytania i polecenia – sprawdzające zdobytą wiedzę



Zadania – sprawdzające stopień opanowania wiedzy i zdobytych umiejętności

Dla zainteresowanych – zadania rozwijające zainteresowania i kreatywność



Przykłady innych projektów grupowych – tematy projektów do wykonania w grupach



Przeczytaj, jeśli chcesz wiedzieć więcej… – rozszerzenie wiedzy na temat interesujących zagadnień z dziedziny informatyki

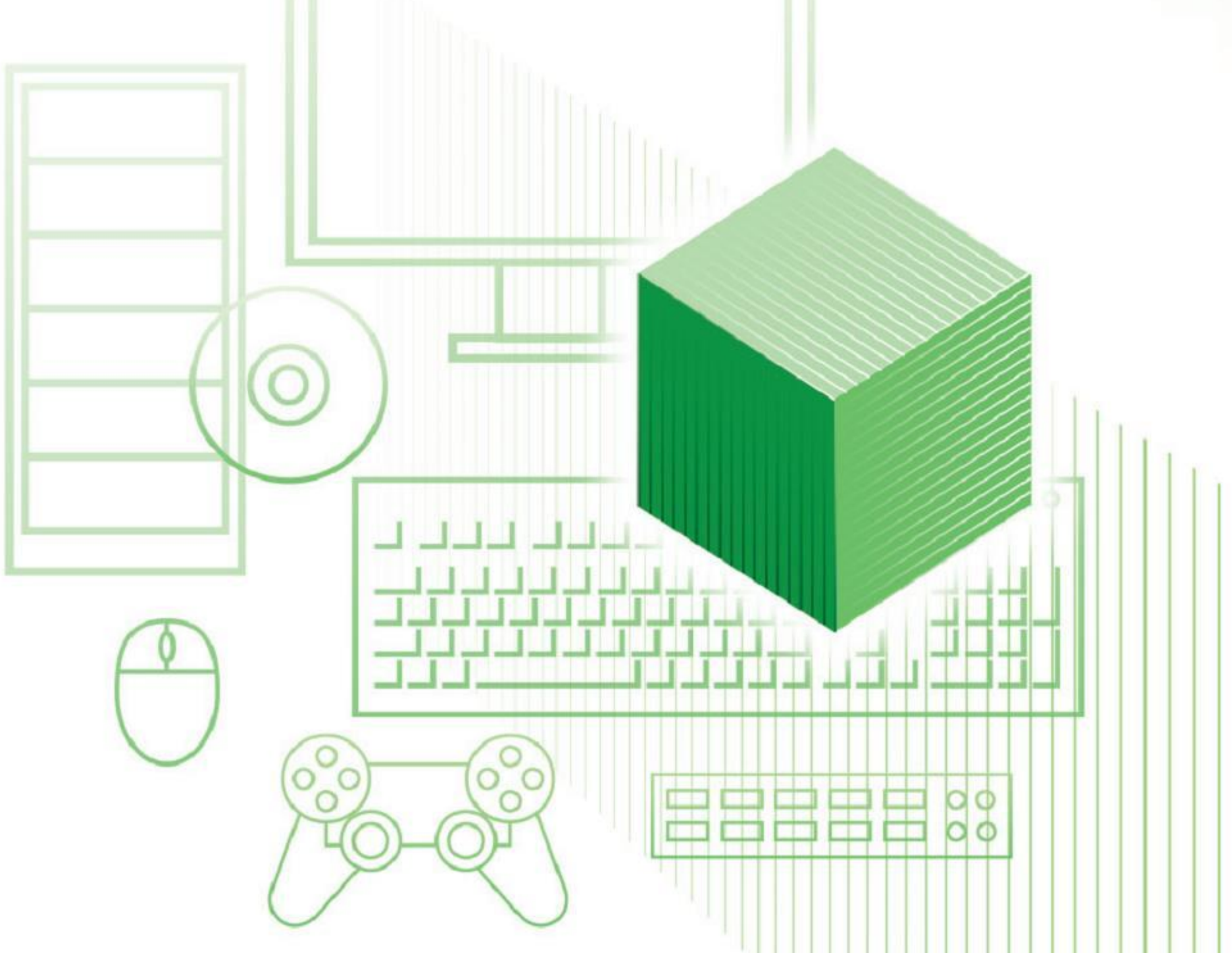
Wokół komputera

A

Moduł

Temat A1. Komputer i urządzenia peryferyjne

Temat A2. Systemy operacyjne
i inne oprogramowanie



Temat A1. Komputer i urządzenia peryferyjne

1. Wyjaśnienie podstawowych pojęć
2. Klasyfikacja środków technologii informacyjnej
3. Jakie parametry komputera warto znać?
   - 3.1. Co można zamontować na płycie głównej?
   - 3.2. Jakie są parametry dysku komputera?
   - 3.3. Jaki wybrać monitor?
4. Dodatkowe urządzenia pamięci masowej
   - 4.1. Napędy optyczne
   - 4.2. Pamięci flash
   - 4.3. Pamięci taśmowe (streamery)
5. Wybieranie parametrów komputera
6. Przykłady urządzeń peryferyjnych

Warto powtórzyć

1. Z jakich elementów składa się zestaw komputerowy?
2. Jaką rolę pełni w komputerze pamięć operacyjna (RAM)? Jakie są jej właściwości?
3. Jakie znasz typy komputerów?
4. Podaj przykładowe zastosowania komputera.

1. Wyjaśnienie podstawowych pojęć

Na lekcjach informatyki będziemy poznawać metody rozwiązywania problemów z wykorzystaniem komputera i oprogramowania komputerowego. Będą to problemy algorytmiczne, do których rozwiązania będziemy wykorzystywać środowiska programistyczne (moduł C), i inne problemy, które będziemy rozwiązywać, stosując oprogramowanie użytkowe (moduły B i D). W trakcie nauki informatyki będziemy dążyć do rozwijania **myślenia komputacyjnego** określającego umiejętności stosowania metod wywodzących się z informatyki do rozwiązywania problemów z różnych dziedzin.

Informatyka obejmuje zagadnienia związane z:

- teoriami informatycznymi, czyli przedstawianiem informacji, ich uzyskiwaniem, przechowywaniem, porządkowaniem i przetwarzaniem;
- systemami informatycznymi, w tym z programowaniem;
- budową i działaniem komputera;
- zastosowaniem metod informatyki w różnych dziedzinach gospodarki i działalności człowieka.

Wraz z wchodzeniem ludzkości w erę społeczeństwa informacyjnego, w której źródłem bogactwa i postępu są wiedza i informacja, nastąpił rozwój sprzętu i oprogramowania komputerowego. Połączenie informatyki z innymi technologiami, np. telekomunikacją, spowodowało kolosalną zmianę w wykorzystaniu nie tylko komputerów i oprogramowania, lecz także urządzeń zewnętrznych. Nastąpił rozwój sieci komputerowych, zwłaszcza Internetu. Pojawiły się ogromne możliwości wykorzystania komputera do zdobywania i wymiany informacji.

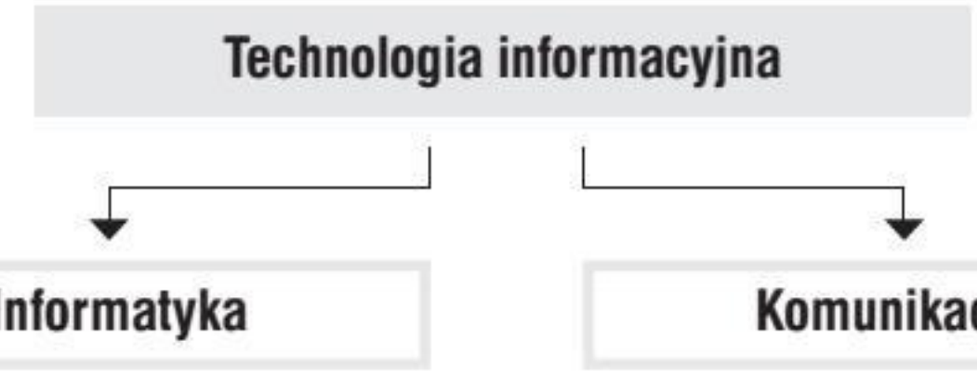
Powstała **technologia informacyjna – IT** (z ang. *Information Technology*), która mieści w swym zakresie, poza informatyką, również inne technologie związane z informacją, przede wszystkim telekomunikację.

Już od dawna telekomunikacja nie polega wyłącznie na przesyłaniu informacji w postaci głosu. Dzięki połączeniu komputerów z urządzeniami telekomunikacyjnymi i wynalezieniu Internetu możliwe stało się przesyłanie nie tylko głosu, lecz także danych, obrazów, filmów. Można dzięki temu m.in. zdobywać informacje z najbardziej odległych miejsc, komunikować się z ludźmi z całego świata czy przeprowadzać wideokonferencje.



Technologia informacyjna obejmuje:

- środki (komputery, ich urządzenia zewnętrzne i sieci komputerowe),
- narzędzia (oprogramowanie),
- inne technologie, przede wszystkim komunikacyjne.





Środki technologii informacyjnej to komputery, ich urządzenia peryferyjne, urządzenia sieci komputerowych i telekomunikacyjnych oraz cyfrowe urządzenia elektroniki użytkowej.

Urządzenia peryferyjne to takie, które nie mogą funkcjonować bez komputera, np.: monitor, drukarka, mysz, skaner, kamera internetowa. Urządzenia sieci komputerowych i telekomunikacyjnych to m.in.: routery, przełączniki, modemy, telefony. Cyfrowe urządzenia elektroniki użytkowej to m.in.: aparaty i kamery cyfrowe, tablety, czytniki e-booków.



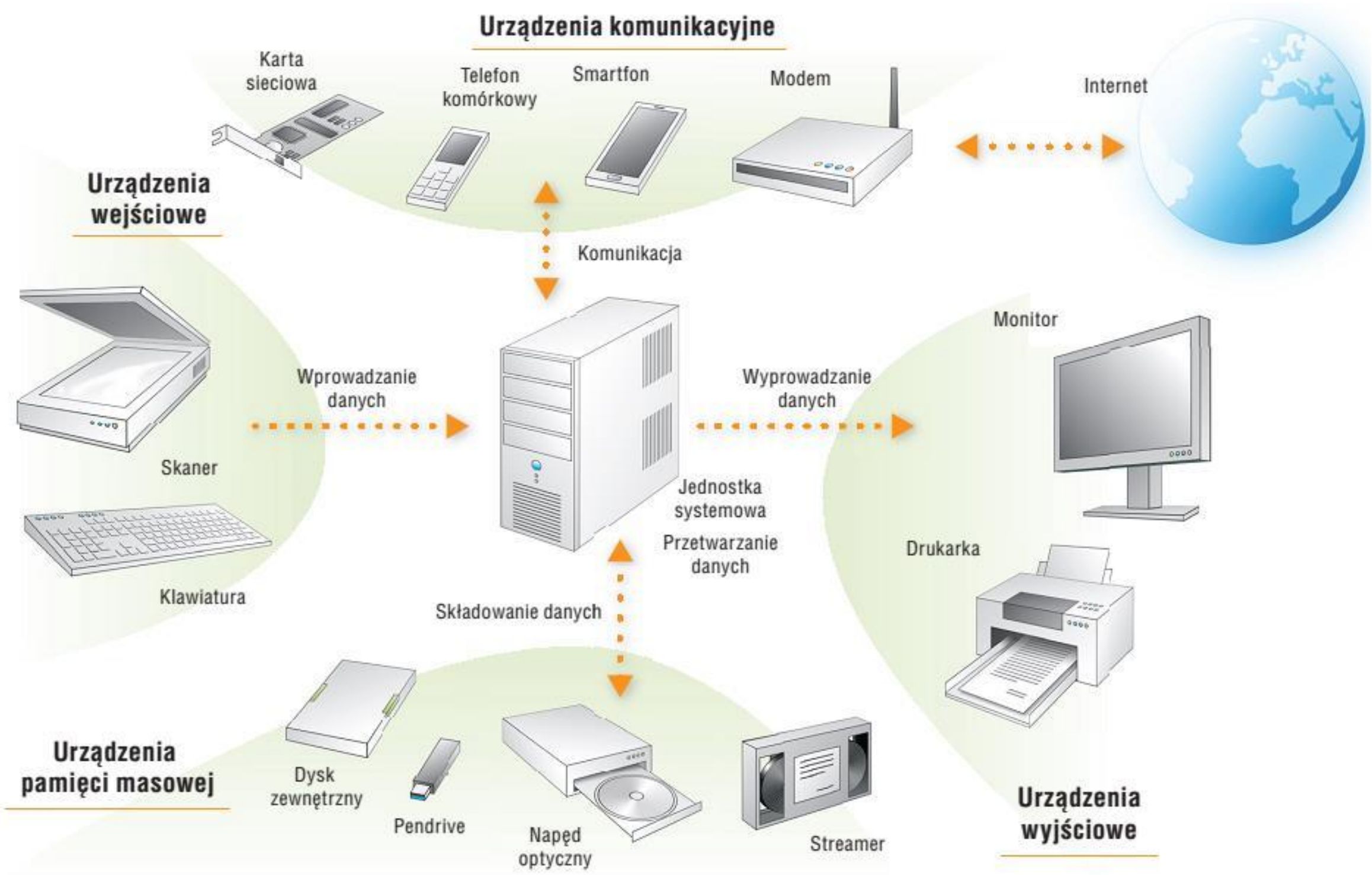
Narzędzia technologii informacyjnej to różnego rodzaju oprogramowanie – systemy operacyjne, programy użytkowe i narzędziowe (w tym gry), języki programowania.

2. Klasyfikacja środków technologii informacyjnej

Klasyczny podział środków IT nie jest już tak klarowny, jak na początku ery komputerów. Rozwój techniki sprawia, że różne funkcje są integrowane w jednym urządzeniu, np. telefon komórkowy może być wyposażony w cyfrowy aparat fotograficzny i posiadać przeglądarkę internetową.

Przeznaczenie	Przykłady
Przetwarzanie danych	procesor (CPU – z ang. *Central Processing Unit*)
Wprowadzanie informacji	klawiatura, mysz, ekran dotykowy, mikrofon
Wyprowadzanie informacji	monitor, drukarka, ploter
Przechowywanie informacji	dysk twardy, nośnik optyczny (CD, DVD, Blu-ray), streamer, pendrive, dysk zewnętrzny
Przetwarzanie obrazu	skaner, kamera cyfrowa, cyfrowy aparat fotograficzny
Wymiana informacji	karta sieciowa przewodowa, karta sieciowa bezprzewodowa, modem do linii telefonicznej analogowej, cyfrowej lub DSL, modem GSM

Tabela 1. *Podział środków technologii informacyjnej ze względu na ich przeznaczenie*



Rys. 1. *Ogólny schemat przepływu informacji z wykorzystaniem środków IT*

3. Jakie parametry komputera warto znać?

{ Planujemy kupić nowy komputer. Jakie jego elementy składowe powinniśmy znać i jakie dobrać parametry tych elementów, zanim wybierzemy komputer? }

Zanim kupimy komputer, powinniśmy poznać funkcje i parametry jego podstawowych elementów.



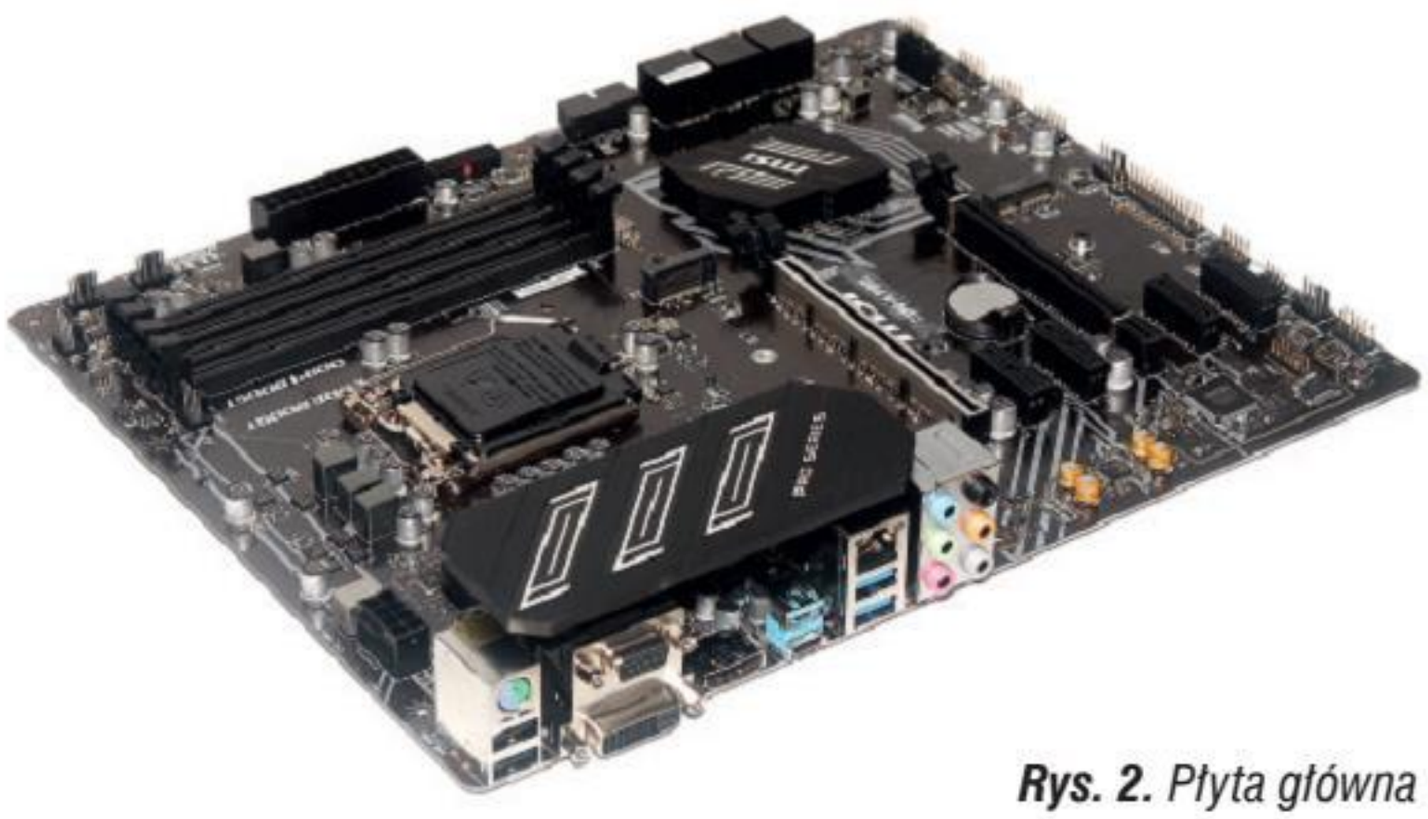
Niezależnie od typu komputera (PC, Mac) w skład podstawowego zestawu komputerowego wchodzą: **jednostka systemowa**, monitor, mysz i klawiatura.

Jednostka systemowa to zasadnicza część zestawu komputerowego, zawierająca w jednej obudowie najważniejsze elementy komputera, przede wszystkim **płytę główną** z **procesorem** i **pamięcią operacyjną** oraz urządzenie **pamięci masowej** (np. **dysk twardy**).

3.1. Co można zamontować na płycie głównej?

Płyta główna to podstawowy podzespół jednostki systemowej. Zależnie od typu komputera (PC, Mac) i jego wersji (stacjonarny, przenośny) płyta główna może być umieszczona w oddzielnej obudowie lub połączona w jedno urządzenie z monitorem (bądź klawiaturą).

Na płycie głównej są umieszczane różne układy elektroniczne, m.in. pamięć **RAM**, pamięć **BIOS/UEFI**, tzw. **chipset**, gniazda (tzw. sloty) **kart rozszerzeń**, a przede wszystkim **procesor**. Na krawędzi płyty są umieszczone złącza **portów**: USB, szeregowego, równoległego, myszy i klawiatury, a także gniazda do podłączania urządzeń dźwiękowych.



Rys. 2. *Płyta główna*

K **Karta rozszerzenia**

Umożliwia komputerowi realizowanie dodatkowych funkcji (np. generowanie realistycznej grafiki komputerowej, odbieranie sygnału telewizyjnego).

R **RAM**
(z ang. *Random Access Memory*)

Pamięć o dostępie swobodnym do chwilowego przechowywania danych i kodów działających programów.

B **BIOS**
(z ang. *Basic Input Output System*)

Podstawowy system wejścia/wyjścia. Oprogramowanie pośredniczące pomiędzy sprzętem a systemem operacyjnym komputera.

U **UEFI**
(z ang. *Unified Extensible Firmware Interface*)

Oprogramowanie pośredniczące pomiędzy oprogramowaniem sprzętowym (ang. *firmware*) a systemem operacyjnym komputera; następca BIOS-u.

Podstawowym kryterium wyboru płyty głównej jest możliwość zamontowania na niej odpowiedniego procesora.



Rys. 3. *Procesor*

Jaki wybrać procesor?

Procesor odpowiada za wykonywanie praktycznie wszystkich operacji, które realizuje jednostka systemowa (tj. załadowanie i uruchomienie systemu operacyjnego, załadowanie sterowników obsługi zainstalowanych urządzeń, uruchamianie programów i nadzorowanie ich pracy, odczytywanie i zapisywanie danych) oraz za nadzorowanie pracy pozostałych podzespołów jednostki systemowej.

Podstawowymi parametrami **procesora** są: architektura (liczba bitów danych, które procesor może jednocześnie przetworzyć), liczba jednostek wykonawczych (rdzeni) oraz częstotliwość taktowania zegara (podawana w gigahercach – GHz; w starszych komputerach w megahercach – MHz). Większość obecnie spotykanych procesorów ma architekturę 32/64-bitową.

Procesor wybieramy w zależności od przewidywanego przeznaczenia komputera, np. do prac biurowych wystarczy procesor o niedużej wydajności, natomiast do bardziej zaawansowanych prac (graficznych, projektowych, obróbki wideo) potrzebujemy szybszego procesora. Architektura **wielordzeniowa** pozwala konstruować procesory, które w jednej obudowie zawierają więcej niż jeden rdzeń. System operacyjny uznaje każdy z tych rdzeni za osobny procesor.

Procesory wielordzeniowe umożliwiają efektywniejsze przetwarzanie danych w przypadku wykonywania jednocześnie wielu zadań. Możliwe jest na przykład równoczesne przesyłanie pliku wideo do innego komputera, edytowanie pliku graficznego oraz skanowanie dysków komputera przez program antywirusowy bez zauważalnego spowolnienia pracy komputera. Najbardziej złożone procesory (dostępne obecnie w sprzedaży) dysponują trzydziestoma dwoma rdzeniami.

Jaką wybrać wielkość pamięci RAM?

Wszystkie uruchamiane programy są ładowane w pierwszej kolejności do pamięci RAM, skąd instrukcje składające się na program są odczytywane przez procesor. W pamięci RAM są przechowywane: kod systemu operacyjnego, kody działających programów oraz dane do programów. Z pamięci RAM informacje są przesyłane do urządzeń wyjściowych.



RAM jest pamięcią ulotną – po wyłączeniu komputera jej zawartość bezpowrotnie ginie. Ważnym parametrem pamięci RAM jest jej pojemność (mierzona w gigabajtach – GB; w starszym sprzęcie w megabajtach – MB) oraz częstotliwość, z jaką pracuje (mierzona w gigahercach – GHz; w starszym sprzęcie w megahercach – MHz).

Częstotliwość taktowania pamięci powinna być zgodna ze specyfikacją płyty głównej i procesora; pojemność dobieramy w zależności od przeznaczenia komputera. Obecnie standardem są 4 GB w komputerach do prostszych zastosowań i 8-32 GB w bardziej rozbudowanych konfiguracjach. Należy zaznaczyć, że 32-bitowe systemy operacyjne (np. Windows XP) nie są w stanie wykorzystać więcej niż 4 GB pamięci operacyjnej. Możliwość wykorzystania większej ilości pamięci mają tylko systemy operacyjne 64-bitowe (odpowiednie wersje systemów Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10).

! Uwaga

Podane w tym temacie wartości parametrów są przykładowe i, biorąc pod uwagę niezwykle dynamiczny rozwój technologii komputerowych, w ciągu kilku lat mogą ulec zmianie.

Czym jest chipset płyty głównej?

Poszczególne elementy płyty głównej: procesor, pamięć, magistrala systemowa, karty rozszerzeń – muszą komunikować się ze sobą w odpowiednich relacjach czasowych i na właściwych poziomach logicznych. Komunikacją tą zarządza zestaw kilku układów scalonych, zwany **chipsetem**.

Czym jest BIOS?

BIOS (w nowych komputerach – **UEFI**) to zbiór programów i procedur odpowiedzialnych za diagnostykę i konfigurację podstawowych parametrów systemu komputerowego oraz za uruchamianie systemu operacyjnego. Program BIOS/UEFI zapisany jest w pamięci stałej typu **ROM**. Pamięć taka zachowuje swoją zawartość po wyłączeniu zasilania.

R ROM
(z ang. *Read Only Memory*)

Pamięć tylko do odczytu.



Ćwiczenie 1. Sprawdzamy parametry komputera

Ustal, w jakiego typu procesor i pamięć RAM wyposażono szkolny komputer, na którym pracujesz. Jeśli masz domowy komputer, porównaj jego parametry z parametrami szkolnego komputera.

Wskazówka: Potrzebne dane można znaleźć w oknie **Panel sterowania**. Aby otworzyć to okno, należy w menu (lub na ekranie) **Start** wybrać **Panel sterowania/System i zabezpieczenia/System** lub z menu kontekstowego ikony **Mój komputer** (**Ten komputer**) – polecenie **Właściwości** (zależnie od wersji systemu operacyjnego).

Jakie wybrać karty rozszerzeń?

Kontroler grafiki

Kontroler grafiki występuje najczęściej w postaci niezależnej **karty graficznej**, ale może być również zintegrowany z płytą główną. Jest to wyspecjalizowany układ, odpowiedzialny za przetwarzanie obrazu i jego wyświetlanie na ekranie monitora.



Rys. 4. *Karta graficzna*

Kontroler grafiki posiada zwykle własny **procesor graficzny** (często o wiele bardziej złożony od procesora głównego) oraz własną **pamięć obrazu**. Dzięki temu możliwe jest tworzenie obrazu bez udziału procesora głównego.

Do przechowywania informacji o obrazie kontrolery zintegrowane z płytą główną wykorzystują najczęściej wydzielony obszar pamięci operacyjnej.

Nowoczesne procesory graficzne, dzięki bardzo dużej mocy obliczeniowej, mogą być również wykorzystywane do przeprowadzania złożonych obliczeń, np. symulacyjnych.



Przy wyborze **kontrolera grafiki** uwzględniamy:

- typ procesora graficznego (w decydujący sposób wpływający na szybkość działania, np. gier komputerowych),
- wielkość pamięci obrazu,
- liczbę i rodzaj wejść/wyjść,
- sposób umieszczenia (np. dla komputera biurowego wystarczy kontroler umieszczony na płycie głównej),
- producenta, zwłaszcza gdy zajmujemy się profesjonalną obróbką grafiki.

Karta sieciowa lub kontroler sieciowy

Element składowy jednostki systemowej komputera, służący do włączenia komputera do sieci komputerowej. Jego główną funkcją jest zamiana danych przechowywanych w komputerze w postaci cyfrowej na sygnały fizyczne (np. elektryczne, optyczne).

Kontroler dźwięku

Coraz częściej funkcję kontrolera dźwięku spełnia **chipset** płyty głównej przy współpracy z procesorem. Większe możliwości daje jednak zastosowanie niezależnej **karty dźwiękowej**. Jest to układ odpowiedzialny za nagrywanie i odtwarzanie muzyki, głosu i innych dźwięków. Niezależna karta dźwiękowa odciąża procesor z funkcji, które przejmuje wyspecjalizowany procesor dźwięku.

Użytkownik, który tylko słucha muzyki, zadowoli się prostą, tanią kartą. Profesjonaliści używają bardziej zaawansowanych kart: wielokanałowych, wyposażonych w syntezatory dźwięku, wejścia cyfrowe, o niskim poziomie szumów własnych.



Rys. 5. *Modem zewnętrzny GSM*



Rys. 6. *Karta telewizyjna (wewnętrzna)*



Rys. 7. *Karta telewizyjna (zewnętrzna)*

Karta modemowa

Modem (rys. 5.) to wyspecjalizowany układ odpowiedzialny za przesyłanie informacji za pomocą sieci telekomunikacyjnych (analogowych, cyfrowych lub GSM). Umożliwia w szczególności łączenie się z Internetem oraz pozwala komputerowi odbierać i wysłać wiadomości, dane, faksy, a także pełnić rolę automatycznej sekretarki lub skrzynki głosowej.

Karta tunera telewizyjnego

Aby korzystać z telewizji w komputerze, potrzebna jest **karta telewizyjna** i odpowiednie oprogramowanie. Karta może być umieszczona wewnątrz komputera (rys. 6.) lub podłączona przez port USB (rys. 7.). Po zainstalowaniu karty lub włożeniu jej do portu USB należy dostarczyć do niej sygnał z anteny lub innego źródła (kamera wideo, magnetowid, DVD).

Odpowiedni program umożliwiający oglądanie telewizji (a także np. nagrywanie) jest dołączany do karty. W niektórych systemach operacyjnych jest dostępne oprogramowanie umożliwiające oglądanie telewizji, np. w Windows 7 – Windows Media Center.

Niekiedy karty telewizyjne umożliwiają także odbiór radia.

3.2. Jakie są parametry dysku komputera?

Dysk komputera (**dysk twardy** lub **dysk SSD**) to podstawowe **urządzenie pamięci masowej**, o dużej pojemności, przeznaczone do trwałego przechowywania danych. Dane zapisane na dysku pozostają na nim bez względu na to, czy jest on zasilany energią elektryczną.

Nośnikiem danych **dysku twardego** (**HDD** z ang. ***Hard Disk Drive***) jest wirujący z dużą prędkością (kilku tysięcy obrotów na minutę) talerz aluminiowy lub szklany, pokryty warstwą magnetyczną (rys. 8.). Zapis i odczyt są realizowane za pośrednictwem głowicy ferromagnetycznej, poruszającej się nad powierzchnią płyty. Położenie chwilowe głowicy, a zatem i miejsce, w którym dokonywany jest w danej chwili zapis lub odczyt informacji, są kontrolowane przez zaawansowany układ elektroniczny, zwany **kontrolerem dysku**.



Rys. 8. *Dysk twardy*



Rys. 9. *Zewnętrzny dysk twardy*

Dyski twarde osiągają pojemność do 20 TB (ok. 20 000 GB).

Do komputerów stacjonarnych i przenośnych można dodatkowo podłączyć **zewnętrzny dysk twardy** (rys. 9.). Dodatkowy dysk umożliwia m.in. wykonywanie kopii zapasowych plików. Można go szybko odłączyć od komputera i zabrać ze sobą.

Z dyskami związane są dwa ważne pojęcia: **partycja** i **formatowanie**. Na jednym dysku (fizycznym) może znajdować się jedna partycja lub kilka partycji dyskowych. Kilka dysków fizycznych można też połączyć w jeden dysk logiczny, tzw. **macierz dyskową** (**RAID** z ang. ***R**edundant **A**rray of **I**ndependent **D**isks*).

Dysk SSD (z ang. ***S**olid **S**tate **D**rive*) nie posiada elementów mechanicznych (talerzy i głowic), zbudowany jest z pamięci półprzewodnikowych typu flash (rys. 10. i 11). Dyski tego typu są odporne na wstrząsy oraz zwykle wielokrotnie szybsze od tradycyjnych dysków twardych, kosztem kilka razy mniejszej pojemności i znacznie wyższej ceny. Szybsza praca dysku SSD w porównaniu ze standardowym HDD zapewnia lepszą jakość pracy komputera.

Odmianą dysków SSD są dyski **SSD M.2** (rys. 11.). Są one o wiele mniejsze od dysków typu HDD i standardowych SSD, dzięki czemu łatwiej je zamontować, szczególnie w komputerach przenośnych (są umieszczone bezpośrednio na płycie głównej). Niektóre dyski tego rodzaju pracują w standardzie NVMe, zapewniającym jeszcze większą wydajność w porównaniu ze standardowymi dyskami SSD.

Partycja dyskowa (dysk logiczny) **P**

Wydzielony z fizycznego obszaru dysku spójny fragment, z punktu widzenia logiki komputera traktowany jako osobne urządzenie (dysk).

Formatowanie dysku **F**

Przygotowanie dysku do współpracy z wybranym systemem plików. Formatowanie dysku usuwa zapisane na nim informacje.



Rys. 10. *Przykładowy dysk SSD*



Rys. 11. *Dysk SSD M.2*

3.3. Jaki wybrać monitor?

Obecnie używa się monitorów ciekłokrystalicznych (LCD), zapewniających wysoką jakość obrazu. Monitory ciekłokrystaliczne zajmują stosunkowo niewiele miejsca, są lekkie i zużywają niewiele energii. Duże monitory panoramiczne pozwalają na jednoczesne wyświetlanie obrazów z wielu źródeł, np. kilku komputerów.



Podstawowymi parametrami monitora są: wielkość ekranu (określona przez długość jego przekątnej, podawana w calach), **proporcja obrazu** oraz **rozdzielczość**.

Obecnie najpopularniejsze są monitory o wielkości ponad 20" i proporcjach obrazu 16:9, dysponujące rozdzielczością 1920x1080 **pikseli** (tzw. standard Full HD). Coraz większą popularność zyskują monitory standardu Ultra HD (inaczej 4K) o rozdzielczości 3840x2160 pikseli lub zbliżonej.

Wielkość ekranu LCD	Rozdzielczość
15"	1024x768
17"-19"	1280x1024, 1440x900
20"-21"	1600x1200, 1680x1050
23"-26"	1920x1080, 1920x1200
27"-28"	2560x1440, 3840x2160
30"	2560x1600

Tabela 2. *Przykładowe wielkości ekranów LCD i odpowiadające im rozdzielczości*

Czy wiesz, że...

na początku ery komputerów wyniki działania programów były wyprowadzane na zewnątrz za pomocą urządzeń zbliżonych w działaniu do dalekopisów (ang. *teletype*)? Urządzenia te umożliwiały drukowanie wyłącznie znaków alfanumerycznych (liter i cyfr).

P Piksel

Najmniejszy element obrazu wyświetlanego na ekranie monitora.

R Rozdzielczość

Określa liczbę pikseli obrazu wyświetlanych na ekranie monitora. Rozdzielczość wyraża się w postaci liczby pikseli w poziomie i w pionie.

P Proporcja obrazu

Stosunek szerokości obrazu do jego wysokości.

4. Dodatkowe urządzenia pamięci masowej

4.1. Napędy optyczne

Napęd optyczny pozwala na odtwarzanie i zapisywanie różnego rodzaju płyt: CD, DVD lub Blu-ray.

Na początku lat 80. XX w. wprowadzono do użytku płyty kompaktowe (**CD** – z ang. ***Compact Disc***). Początkowo służyły one do nagrywania muzyki. Z czasem znalazły zastosowanie w technice komputerowej jako nośniki informacji o dużej pojemności. Na standardowej płycie można zmieścić 700 MB informacji.

Napęd optyczny N

Urządzenie, które za pomocą wiązki lasera odczytuje dane lub zapisuje je na tzw. nośnikach optycznych (płyty CD, DVD, Blu-ray).



***Rys. 12.** Zewnętrzny napęd optyczny CD/DVD*

Większą pojemność oferują płyty **DVD** (powszechnie używane do dystrybucji filmów). Pojemność standardowej płyty DVD to 4,7 GB, natomiast dwustronnej – 8,5 GB. Jeszcze bardziej pojemne są płyty **Blu-ray** (powszechnie używane do dystrybucji filmów w jakości HD), w których do odczytu używa się niebieskiego lasera, dzięki czemu stało się możliwe osiągnięcie pojemności od 25 GB (na płytach jednowarstwowych) do 200 GB (na płytach ośmiowarstwowych).

Płyty CD, DVD i Blu-ray przeznaczone są tylko do odczytu. Istnieją również płyty pozwalające na jednokrotny (CD-R, DVD-R, DVD+R, BD-R) lub wielokrotny zapis (CD-RW, DVD-RW, DVD+RW, DVD-RAM).

Napęd optyczny może znajdować się wewnątrz komputera lub stanowić odrębne, zewnętrzne urządzenie, podłączane do komputera za pomocą odpowiedniego złącza, np. USB.

4.2. Pamięci flash

Bardzo dynamiczny rozwój urządzeń przenośnych wywołał zapotrzebowanie na nośniki pamięci masowej o małych gabarytach, lecz o dużej pojemności. Wymogi te spełniają **pamięci półprzewodnikowe**, zwane inaczej **pamięciami flash**. Pamięci flash wykorzystuje się w urządzeniach typu **pendrive** (wyposażonych w interfejs USB, umożliwiających gromadzenie danych oraz ich przenoszenie z komputera do komputera) oraz w **kartach pamięci** (zwykle typu **SD** – z ang. ***S**ecure **D**igital*), stosowanych powszechnie w cyfrowych aparatach fotograficznych, odtwarzaczach MP3, dyktafonach czy telefonach komórkowych.

4.3. Pamięci taśmowe (streamery)

Pamięci taśmowe (**streamery**) są mało znane użytkownikom komputerów domowych, ale powszechnie wykorzystuje się je w dużych systemach informatycznych do archiwizacji danych. Urządzenia te pozwalają na zapisywanie dużych ilości danych na taśmie magnetycznej, podobnej do magnetofonowej. Streamery osiągają pojemności zapisu do 20 TB na jednej taśmie. W razie konieczności zapisania większej ilości informacji wystarczy włożyć następną taśmę i kontynuować zapis.

5. Wybieranie parametrów komputera

Przy zakupie komputera wybór konkretnych elementów i ich parametrów uzależniamy od przewidywanego przeznaczenia komputera – przykłady pokazujemy w tabeli 3.

Zastosowanie komputera	Istotne parametry – przykładowe wielkości
Komputer domowy (korzystanie z Internetu, tworzenie prostych dokumentów, okazjonalne granie w gry komputerowe)	• pamięć RAM 4-8 GB (system operacyjny 64-bitowy) • dysk twardy 1-2 TB (przechowywanie zdjęć, filmów, muzyki i innych plików o dużej objętości)
Komputer do gier	• pamięć RAM 8-16 GB (system operacyjny 64-bitowy) • wydajna karta graficzna • monitor o dużej rozdzielczości i przekątnej ekranu • dysk SSD oraz dodatkowy dysk HDD na dane
Komputer biurowy	• pamięć RAM 8-16 GB (system operacyjny 64-bitowy) • dysk SSD • system operacyjny posiadający dodatkowe mechanizmy do zarządzania, np. systemy Windows w wersji Professional, Pro lub Enterprise
Komputer profesjonalny (montaż filmów, muzyki, obróbka zdjęć cyfrowych, projektowanie inżynierskie, skład publikacji)	• procesor przynajmniej 4-rdzeniowy • pamięć RAM 16 GB (system operacyjny 64-bitowy) • dysk SSD do instalacji programów i przechowywania aktualnie opracowywanych dokumentów • dodatkowy dysk twardy 2-4 TB przeznaczony na magazyn plików • monitor o dużej rozdzielczości i przekątnej ekranu, z matrycą wiernie odwzorowującą barwy

Tabela 3. *Przykładowe parametry komputera – zależnie od jego zastosowania*

6. Przykłady urządzeń peryferyjnych

Do urządzeń związanych z technologiami informacyjno-komunikacyjnymi możemy zaliczyć urządzenia peryferyjne (tabela 4.).

Przykłady urządzeń peryferyjnych			
	Funkcja, przeznaczenie	**Działanie**	**Główny parametr**
Drukarki	**Igłowe** Obecnie używane niemal wyłącznie w firmach handlowych do wystawiania faktur.	Znak graficzny na papierze powstaje na skutek uderzenia cienkiej stalowej igły w taśmę barwiącą, umieszczoną tuż nad powierzchnią papieru.	Liczba igieł: 9 lub 24.
	Atramentowe Monochromatyczny lub kolorowy wydruk tekstów i obrazów. Popularne wśród użytkowników dzięki niskiej cenie oraz możliwości druku kolorowego.	Obraz tworzony jest przez miniaturowe kropelki atramentu rozpylane ze sterowanego elektronicznie zespołu dysz.	Rozdzielczość: typowo 1200 dpi.

Przykłady urządzeń peryferyjnych			
	Funkcja, przeznaczenie	**Działanie**	**Główny parametr**
Drukarki	**Laserowe** Monochromatyczny lub kolorowy wydruk tekstów i obrazów w bardzo wysokiej jakości.	Druk powstaje najpierw na odpowiednio elektryzowanym bębnie selenowym. Stamtąd zostaje przeniesiony na papier. W procesie druku wykorzystywany jest toner – proszek barwiący, który po podgrzaniu zostaje wtopiony w przesuwającą się kartkę papieru.	Rodzaj: monochromatyczne lub kolorowe, rozdzielczość: typowo 600 dpi.
	3D Drukowanie fizycznych obiektów przestrzennych techniką przyrostową (technika addytywna).	Trójwymiarowy wydruk powstaje na bazie przygotowanego wcześniej komputerowego modelu 3D, zapisanego do postaci pliku *stl* lub *g-code*. W domowych drukarkach 3D używa się przede wszystkim tworzyw sztucznych (technologia FDM) w postaci żyłki o różnej grubości (najczęściej 1,75 mm). Drukarki przemysłowe mogą używać innych materiałów np.: żywicy, gumy, metalu, papieru, gipsu, a nawet betonu czy szkła.	Technologia wydruku, np: • FDM, • SLM, • SLA.
Ploter	Drukowanie, wycinanie, grawerowanie na materiałach płaskich. Plotery są wykorzystywane głównie w pracowniach projektowych i agencjach reklamowych.	Rysunek powstaje dzięki pisakom, które poruszają się nad kartką. Istnieją również plotery tnące, w których zamiast pisaków używa się odpowiednich noży. Służą do wycinania różnorodnych kształtów i napisów z folii samoprzylepnej.	Format, np. A3.
Frezarka	**CNC** Wytwarzanie fizycznych obiektów przestrzennych techniką ubytkową (technika subtraktywna).	Z bryły materiału za pomocą narzędzi skrawających jest usuwany naddatek, aż do osiągnięcia docelowego modelu.	Moc silnika i twardość obrabianych materiałów.
Skaner	Zamiana obrazu drukowanego (tekst, rysunek, zdjęcie) na jego elektroniczne odwzorowanie. Skaner może występować w roli kopiarki (kopia jest wysyłana do drukarki podłączonej do komputera); można też zeskanowany obraz wysłać jako załącznik do listu elektronicznego.	Po zapisaniu obrazu w postaci pliku może on zostać poddany obróbce za pomocą specjalistycznych programów – edytorów map bitowych i programów OCR służących do zamiany obrazu na tekst.	Rozdzielczość optyczna: typowo 1200 dpi.

Projektor multimedialny	Przekazywanie obrazu bezpośrednio z karty graficznej komputera na specjalny ekran lub ścianę. Przydatny do wyświetlania np. prezentacji multimedialnych.	Obraz przesłany z komputera jest wyświetlany na miniaturowych ekranach ciekłokrystalicznych lub odbijany przez mikrolusterka, a następnie powiększany za pomocą soczewki.	Rozdzielczość obrazu: typowo od 1024x768 do 1920x1080 pikseli.
Kamera internetowa	Prowadzenie wideokonferencji za pośrednictwem Internetu, nadzorowanie i obserwacja pomieszczeń. Prowadzenie przez Internet rozmów telefonicznych (np. za pośrednictwem programu Skype).	Podobnie jak w cyfrowym aparacie fotograficznym, obraz rzutowany na światłoczułą matrycę zostaje przekształcony w sygnał cyfrowy i przesłany bezpośrednio do komputera.	Rozdzielczość: od 640x480 pikseli.

Tabela 4. *Przykłady urządzeń peryferyjnych*

Do urządzeń technologii informacyjno-komunikacyjnych zaliczamy również przenośne urządzenia cyfrowe (**urządzenia mobilne**) oraz **cyfrowe urządzenia techniki użytkowej** (np. aparaty i kamery cyfrowe).

Urządzenia mobilne umożliwiają m.in. przeglądanie stron internetowych, odbieranie i wysyłanie e-maili. Funkcję takich urządzeń pełnią m.in. **smartfony** oraz inne urządzenia, np.: komputery przenośne, **tablety**, przenośne urządzenia do nawigacji satelitarnej **GPS**. Dane urządzenie jest wyposażone w odpowiednie aplikacje mobilne, m.in. w system operacyjny, klienta poczty elektronicznej, przeglądarkę internetową, pakiet biurowy.

! Uwaga

Więcej na temat urządzeń cyfrowych (ich funkcji i towarzyszącego im oprogramowania) dowiesz się w kolejnych klasach.

Warto zapamiętać

- Komputery typu PC i Mac mogą występować w wersji stacjonarnej i przenośnej. Przykładami komputerów przenośnych są: notebooki, netbooki, tablety. Funkcję komputera pełni również smartfon (inteligentny telefon komórkowy).
- Zasadniczą częścią komputera jest jednostka systemowa. W skład podstawowego zestawu komputerowego wchodzą także monitor, mysz i klawiatura.
- Najważniejsze układy (procesor, pamięć RAM, BIOS, chipset, gniazda kart rozszerzeń) są umieszczone na płycie głównej komputera.
- Pamięć RAM, do której ładowana jest kopia programów po ich uruchomieniu, jest pamięcią ulotną – jej zawartość ginie po wyłączeniu komputera. Pamięć ROM, w której zapisane są programy odpowiedzialne za diagnostykę komputera, jest pamięcią stałą.
- Do przechowywania danych (plików, programów) wykorzystuje się urządzenia pamięci masowej, np. dyski twarde, napędy optyczne, pamięci flash.
- Poszczególne elementy zestawu komputerowego, urządzenia peryferyjne i urządzenia techniki użytkowej charakteryzują się różnymi parametrami, określającymi m.in. szybkość działania i pojemność.

- Urządzenia peryferyjne to te, które można podłączyć do komputera, m.in.: drukarki, frezarki, skanery, plotery, modemy, kamery internetowe, głośniki, mikrofony.
- Przenośne urządzenie elektroniczne, które umożliwia korzystanie z Internetu, określamy jako urządzenie mobilne (np.: tablet, smartfon, przenośne urządzenie do nawigacji satelitarnej).

Pytania i polecenia

1. Czym są środki IT? Przedstaw podział środków IT, podając ich przeznaczenie.
2. Jakie znasz rodzaje komputerów przenośnych?
3. Czym jest jednostka systemowa?
4. Co może mieścić się na płycie głównej?
5. Jaką rolę pełni procesor? Jaki parametr charakteryzuje jego jakość?
6. Na co powinniśmy zwracać uwagę, wybierając pamięć RAM?
7. Jakie karty rozszerzeń warto dodać do komputera?
8. Za co odpowiada kontroler grafiki?
9. Omów rodzaje pamięci masowych.
10. Jakiego nośnika danych należy użyć, aby zapisać wiele dużych plików graficznych?
11. Podaj przykłady urządzeń peryferyjnych. Omów ich przeznaczenie i działanie.
12. Jaki rodzaj drukarki najlepiej zakupić na domowy użytek? Uzasadnij odpowiedź.

Zadania

1. Planujesz zakup komputera do tworzenia dokumentów komputerowych, nauki programowania i do korzystania z Internetu. Przedstaw w tabeli propozycje elementów zestawu komputerowego z ich parametrami.
2. Utwórz w edytorze tekstu tabelę podobną do tabeli 1. Uzupełnij ją, wpisując przykłady środków IT. Poza przykładami omówionymi w tym temacie podaj także inne, wyszukane w Internecie.
3. Opisz w edytorze tekstu szkolny zestaw komputerowy – wymień wszystkie urządzenia peryferyjne, w które jest wyposażony.

Dla zainteresowanych

4. Planujesz zakup komputera do tworzenia dokumentów komputerowych, nauki programowania i do korzystania z Internetu. Przedstaw w tabeli propozycje elementów zestawu komputerowego z ich parametrami.
5. Twoja siostra, która zajmuje się profesjonalnie grafiką komputerową, poprosiła ciebie o pomoc w doborze parametrów nowego komputera. Przedstaw w tabeli propozycje elementów z ich parametrami.
6. Zastanów się, w jakim kierunku zmierza rozwój sprzętu komputerowego. Jaki będzie komputer przyszłości?
7. Znajdź w Internecie więcej informacji na temat macierzy dyskowych RAID.

Temat A2. Systemy operacyjne i inne oprogramowanie

1. Początki systemów operacyjnych
2. Funkcje i struktura systemu operacyjnego
3. Wybrane systemy operacyjne
   - 3.1. MS DOS
   - 3.2. Microsoft Windows
   - 3.3. Mac OS/macOS
   - 3.4. Unix
   - 3.5. Linux
   - 3.6. Systemy operacyjne na urządzenia mobilne
4. Oprogramowanie użytkowe, narzędziowe i języki programowania
5. Przykładowe typy plików

Warto powtórzyć

1. Czym są narzędzia technologii informacyjnej?
2. Jakie są zadania procesora?
3. Czym jest BIOS?
4. Wyjaśnij skrót ROM.
5. Jaką funkcję pełni w komputerze pamięć operacyjna?
6. Podaj przykładowe nazwy programów użytkowych.

1. Początki systemów operacyjnych

Początkowo programy użytkowe powstawały równocześnie z komputerem. Komputery budowano do wykonania konkretnego zadania, tworząc jednocześnie program, który miał to zadanie realizować. Oprogramowanie było zatem ściśle związane z danym komputerem i jego funkcją. Zmiana zastosowania komputera była niezwykle uciążliwa i kosztowna. Wymagała nie tylko nowego programu, ale równocześnie przebudowy fragmentów maszyny. Pomysłem na rozwiązanie tego problemu było zastosowanie programu, który z jednej strony współpracowałby ściśle ze sprzętem, a z drugiej – reagował na działanie użytkownika. Zamiast podmieniać fragmenty maszyny można podmieniać fragmenty programu pośredniczącego. Tak zrodził się pomysł **systemu operacyjnego** (z ang. **OS** – ***O**perating **S**ystem*).

Pierwsze systemy operacyjne, choć nie były związane z konkretnym zadaniem, mogły obsługiwać tylko konkretne maszyny, a dokładnie serie (rodziny) maszyn. Producenci tworzyli systemy operacyjne na użytek produkowanych przez siebie

komputerów. Musiało upłynąć jeszcze trochę czasu, zanim powstały tzw. **systemy otwarte**, związane jedynie z listą rozkazów procesora, a nie konstrukcją maszyny.

Początkowo system operacyjny umieszczano w pamięci typu ROM, zwykle z jakimś prostym językiem programowania. Tak były budowane w Polsce w latach 60. XX wieku komputery z rodziny ZAM (z językiem SAKO) i komputery z serii Odra (z językiem Jean), a w latach 80. XX wieku – komputery ZX Spectrum, Commodore C64, Atari 800XL (z językiem BASIC). Pojawienie się stacji dysków elastycznych oraz dysków twardych stało się impulsem do opracowania tzw. **dyskowych systemów operacyjnych**.

2. Funkcje i struktura systemu operacyjnego

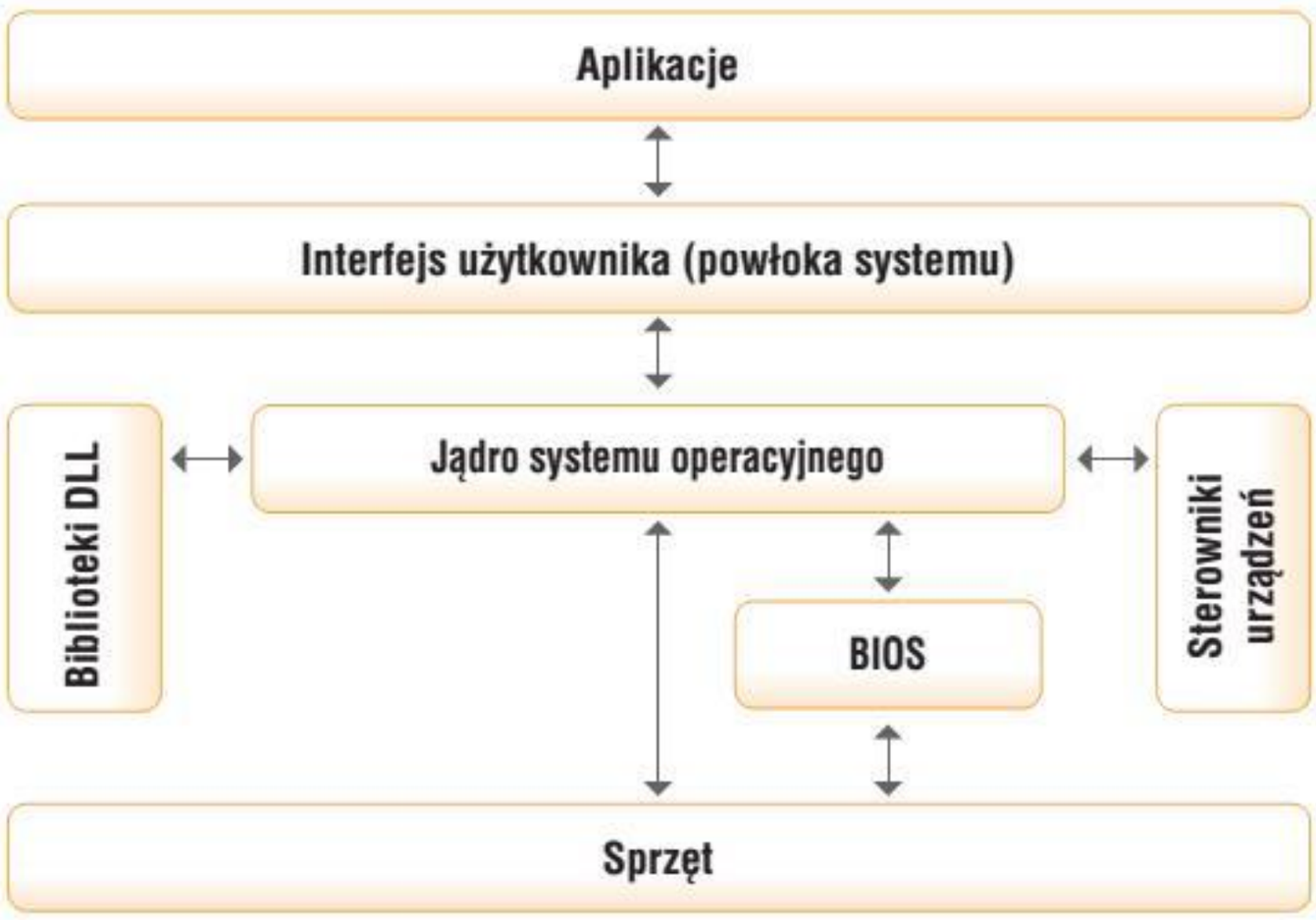


System operacyjny jest nadrzędnym programem kontrolującym pracę komputera. Jest on niezbędny dla uruchamiania innych programów i nadzoruje ich działanie.

System operacyjny:

- pośredniczy pomiędzy użytkownikiem i komputerem oraz programami użytkowymi;
- zarządza komputerem, m.in. pamięcią operacyjną i zasobami komputera, w szczególności plikami zapisanymi na dysku;
- nadzoruje instalowanie, uruchamianie i wykonywanie innych programów;
- umożliwia wykonywanie operacji na dokumentach, które tworzy się w programach komputerowych: ich nazywanie, zapisywanie, kopiowanie, drukowanie;
- sygnalizuje błędy w działaniu komputera i aplikacji.

Główną część systemu operacyjnego stanowi **jądro**. Jest to zbiór najważniejszych podprogramów systemu. Jądro komunikuje się z urządzeniami zewnętrznymi za pośrednictwem sterowników, a z programami użytkownika – przez **powłokę systemu** (rys. 1.).



Rys. 1. *Ogólna struktura systemu operacyjnego na przykładzie Microsoft Windows*

Razem z danym systemem operacyjnym instalowane są zazwyczaj: podstawowy edytor tekstu, podstawowy edytor grafiki, proste gry.



Ćwiczenie 1. Omawiamy najczęściej używane funkcje systemu operacyjnego

Wymień funkcje systemu operacyjnego, z których najczęściej korzysta użytkownik komputera. Na konkretnych przykładach przedstaw ich zastosowania.

3. Wybrane systemy operacyjne

W kolejnych punktach omówimy kilka systemów operacyjnych, które były kiedyś popularne bądź są powszechnie stosowane w obecnie używanych komputerach i urządzeniach mobilnych.

Najpopularniejszym systemem operacyjnym dla komputerów PC jest opracowany przez firmę Microsoft system **Windows** (kolejno w wersjach: 3.x, 95, 98, ME, NT, 2000, XP, 2003, Vista, 7, 8, 8.1, 10, 11). Komputery Mac wykorzystują systemy z rodziny **Mac OS** (Mac OS i nowszą wersję o nazwie **macOS**). Na prawie każdym komputerze można natomiast uruchomić system **Linux** – darmowy system operacyjny, rozwijany na licencji GPL.

3.1. MS DOS

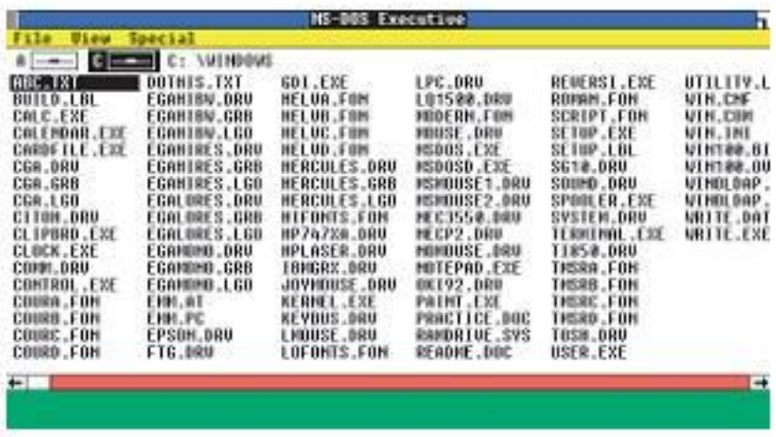
W 1981 roku w firmie Microsoft powstał system operacyjny **MS DOS**, który był rozwijany do roku 1995. Stanowił on nieodłączne wyposażenie setek milionów komputerów osobistych, opartych na konstrukcji IBM PC. Kolejne jego wersje były coraz bardziej rozbudowane i coraz bardziej funkcjonalne. MS DOS to system jednozadaniowy, przeznaczony dla jednego użytkownika. Ma ograniczone możliwości zarządzania pamięcią, wymaga ograniczenia modułów programów użytkowych do 64 KB, nie posiada prawie żadnych możliwości graficznych.

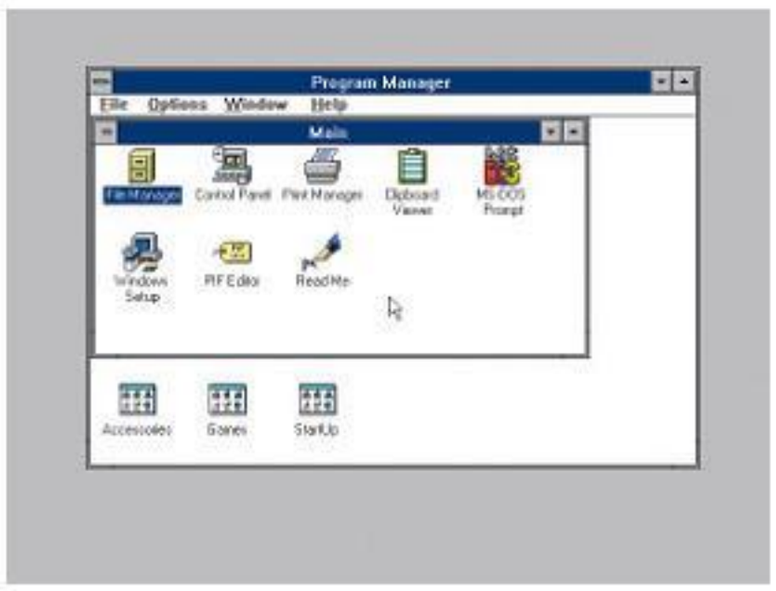
Z czasem dołożono do niego **graficzną nakładkę** i tak powstał system znany pod nazwą **Microsoft Windows**.

3.2. Microsoft Windows

System operacyjny Microsoft Windows był odpowiedzią firmy Microsoft na rosnące zapotrzebowanie na systemy zorientowane graficznie, wprowadzone wcześniej w komputerach Apple Macintosh. Chociaż pierwsze dwie wersje systemu Windows przeszły prawie niezauważone, to wersja Windows 3.0 (z roku 1990) wywołała duże zainteresowanie, a prawdziwym bestsellerem okazała się wersja Microsoft Windows 3.11 for Workgroups, oferująca możliwość pracy w sieci.

Swój sukces Microsoft Windows zawdzięcza standaryzacji wielu elementów użytkowania programów. Do tej pory każdy producent oprogramowania tworzył własne standardy czcionek, własne systemy menu i nawigacji, własne polecenia. Przyswojenie przy każdym programie tylu nowości wymagało wiele czasu. Microsoft Windows wprowadził standaryzację wielu składników programów, dzięki czemu poznanie nowego programu nie wymaga nauki od podstaw.











Rys. 2. *Zmiany graficznego interfejsu systemu Microsoft Windows (kolejno wersje: 1.0, 3.1, 98, XP, 7)*

Microsoft Windows wprowadził też nowe mechanizmy zarządzania pamięcią, dzięki czemu można wprowadzić do niej kilka programów i używać ich równocześnie. Stąd mówimy, że jest to system **wielozadaniowy**. Jeżeli któryś z programów nie mieści się w obszarze pamięci RAM, system Windows przepisuje go na dysk do obszaru tzw. pamięci wirtualnej. Program w dalszym ciągu pozostaje aktywny, chociaż odwołania do niego są wolniejsze (czas dostępu do dysku jest dużo dłuższy niż odwołania do pamięci RAM).

Microsoft Windows wprowadził ponadto takie nowości, jak **DDE** (ang. ***D**ynamic **D**ata **E**xchange* – dynamiczna wymiana danych) i **OLE** (ang. ***O**bject **L**inking and **E**mbedding* – łączenie i osadzanie obiektów). Pozwalają one na współużytkowanie danych przez kilka aplikacji.

Kolejne wersje Microsoft Windows (95, 98 i Millennium) wprowadzały dalsze ulepszenia i dodatki. Między innymi wprowadzono 32-bitowy system plików. Umożliwiono stosowanie długich (do 256 znaków) nazw plików i katalogów. Poszerzono możliwości współpracy z siecią. Dla zaawansowanych użytkowników rozwijano równolegle systemy profesjonalne: Microsoft Windows NT, Microsoft Windows 2000.

W 2002 roku do użytku weszła wersja systemu o nazwie Microsoft Windows XP, łącząca cechy systemów domowych (95, 98) i profesjonalnych (NT, 2000). Następcami systemu Windows XP były systemy: Windows Vista (2006 r.), Windows 7 (2009 r.) i Windows 8 (2012 r.). Nie wprowadzano zasadniczych zmian w funkcjonowaniu systemu, natomiast modyfikacji ulegały elementy graficznego systemu komunikacji z użytkownikiem, poprawiono również zabezpieczenia systemu.

Wersję Windows 8 przystosowano do pracy z urządzeniami przenośnymi (smartfony, tablety). Interfejs użytkownika systemu Windows 8 nastawiono na pracę z ekranami

dotykowymi – posiada interfejs kafelkowy, w którym menu **Start** zostało zastąpione kafelkami na pulpicie (rys. 3a.). Kafelki umożliwiają dostęp do aplikacji systemu Windows 8 poprzez ich kliknięcie lub dotknięcie. Aby jednak korzystać z opcji dotykowych, potrzebny jest tablet lub monitor z funkcją **wielodotyku**.

Użytkownik może łączyć kafelki w dowolne grupy. Dla użytkowników, którzy przyzwyczaili się do tradycyjnego pulpitu, istnieje możliwość korzystania także z niego.

Nową funkcją wprowadzoną w Windows 8 jest pasek **Charms** (**Zaklęcia**) – na rysunku 3a widoczny w postaci kilku białych ikon po prawej stronie ekranu. Jest to menu, do którego dostęp jest możliwy w każdym momencie i z poziomu każdej aplikacji. Aby wyświetlić pasek **Zaklęcia**, wystarczy przesunąć kursor myszy lub palec (w przypadku ekranu dotykowego) od prawego górnego rogu ekranu w dół.

W Windows 8 usprawniono znajdowanie i otwieranie plików oraz wyszukiwanie aplikacji. Można wyszukiwać wszystkie elementy z poziomu ekranu **Start** (rys. 3a), wpisując je z klawiatury. Po wpisaniu pierwszego znaku, np. „p", pokaże się ekran z wyświetlonymi aplikacjami, których nazwy zaczynają się na literę „p" lub ją zawierają (rys. 3b).

W Windows 10 zmodyfikowano menu **Start**, łącząc listę z poprzednich wersji systemu z ekranem kafelkowym (rys. 4.).

Niezależnie od zmieniającego się wyglądu menu **Start**, w kolejnych wersjach systemu Windows nie zmienia się jego funkcja. Korzystając z tego menu, możemy przejrzeć zainstalowane w komputerze programy i uruchomić wybrany z nich.



Rys. 3a. *Interfejs systemu Microsoft Windows 8 (ekran **Start**)*



Rys. 3b. *Wyszukiwanie aplikacji w Windows 8*



Rys. 4. *Interfejs systemu Microsoft Windows 10*

W **Wielozadaniowość**

Możliwość równoczesnego uruchamiania wielu aplikacji.

3.3. Mac OS/macOS

Czy wiesz że...

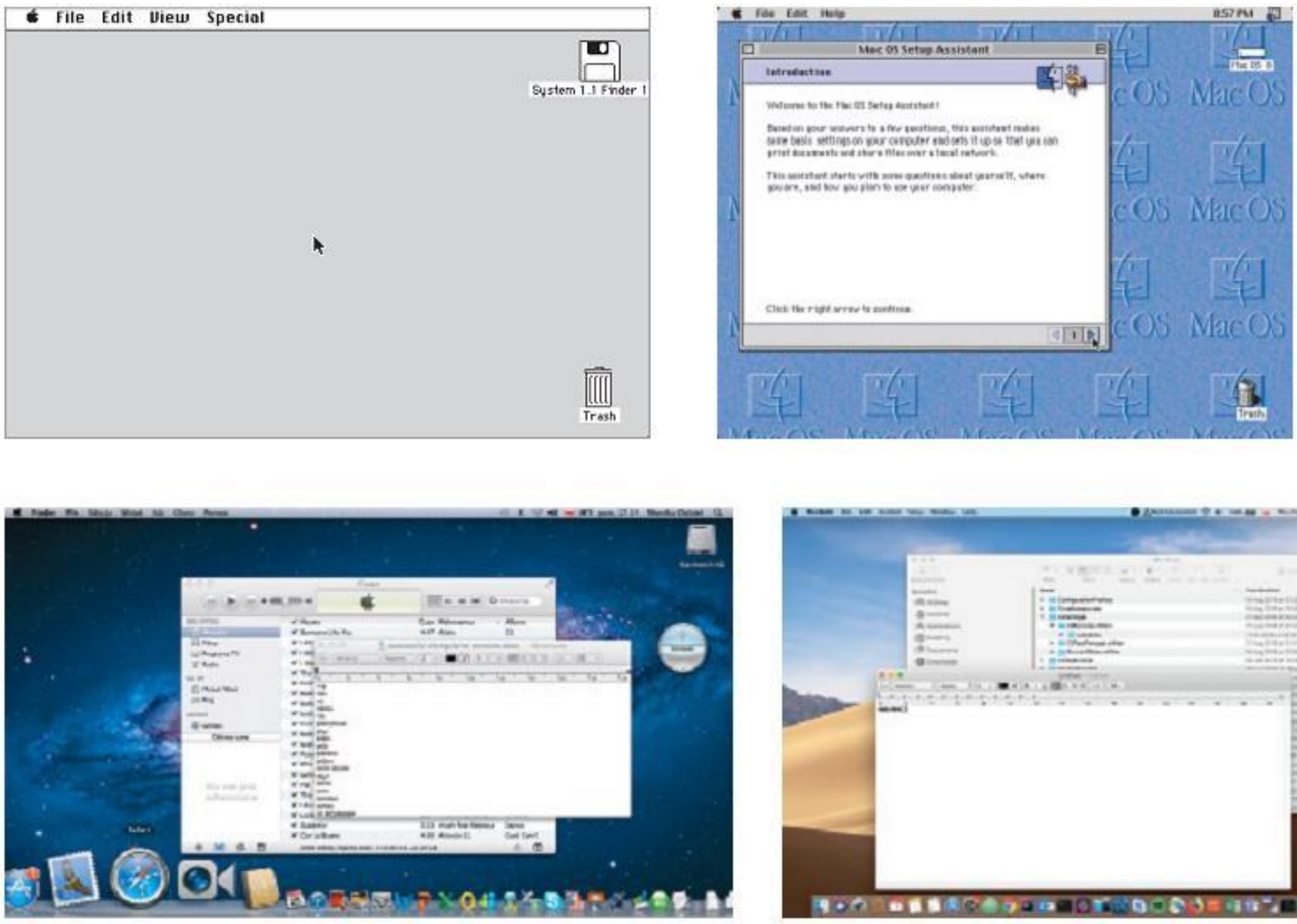
system Mac OS był podobno testowany przez babcię Steve'a Woźniaka (konstruktora komputera Apple)?

Zanim powstał Microsoft Windows, komputery Macintosh (obecnie Mac) firmy Apple dysponowały już graficznym interfejsem użytkownika – **GUI** (ang. ***Graphical User Interface***). Opracowany dla nich system operacyjny Mac OS cechowała łatwość obsługi i instalacji. GUI był tak długo zmieniany, aż stał się przyjazny i oczywisty nawet dla zupełnie początkujących użytkowników. W tym systemie ekran monitora również przedstawia pulpit z ikonami.

Produkowanie komputerów tylko przez jedną firmę ograniczyło kłopoty związane z kompatybilnością. Instalacja i konfiguracja systemu nie sprawiają większych problemów, dlatego komputery Mac są uważane za bardziej przyjazne dla mniej doświadczonych użytkowników.

Komputery Mac są często wykorzystywane przez grafików komputerowych, ponieważ opracowano dosyć duży wybór programów graficznych pracujących w systemie Mac OS.

Następcą systemu Mac OS jest system o nieznacznie zmienionej nazwie – **macOS** (również z rodziny systemów operacyjnych Mac OS).



Rys. 5. *Zmiany graficznego interfejsu systemu Mac OS (kolejno wersje: 1.0, 8, X 10.7) i macOS w wersji 10.14.3*

3.4. Unix

Opracowany na początku lat 70. XX w. Unix był jednym z najbardziej popularnych systemów operacyjnych do zastosowań profesjonalnych.

Unix jest systemem **wielodostępowym** i wielozadaniowym. Posiada bardzo rozbudowane metody zabezpieczeń na każdym poziomie hierarchii katalogów i plików, potrafi obsługiwać wiele procesorów, jak również organizować systemy wielomaszynowe (**klastry**). Unix był pierwowzorem coraz bardziej popularnego dziś Linuksa.

System Unix miał znaczny wpływ na dzisiejszą informatykę, m.in. temu systemowi zawdzięcza swoją popularność język programowania C. Jeden z wariantów systemu operacyjnego Unix stał się podstawą do stworzenia systemu macOS.

W **Wielodostępowość**

Cecha systemu operacyjnego, pozwalająca na korzystanie z tego samego komputera różnym użytkownikom, mającym swoje konta i własne ustawienia.

3.5. Linux

Linux to popularny, darmowy system operacyjny, działający na komputerach różnego typu. Oparty jest na filozofii *open source*, tzn. system jest darmowy, a kod źródłowy systemu jest dostępny dla każdego. Linux dał się poznać jako system stabilny, dopracowany i o dużych możliwościach. Szczególnie chętnie wykorzystuje się go jako system dla serwerów internetowych. Użytkownicy komputerów domowych również instalują Linuksa. Sprzyja temu coraz większa liczba aplikacji użytkowych, skutecznie zastępujących drogie programy platformy Windows. Specjalizowane wersje Linuksa wykorzystywane są w urządzeniach elektronicznych (np. routery, inteligentne telewizory itp.).

Linux jest dostępny w postaci tzw. **dystrybucji**. Stanowią one zbiór oprogramowania zawierający system operacyjny (ściślej: jądro systemu) wraz z niezbędnymi sterownikami urządzeń oraz zestaw podstawowych programów użytkowych, takich jak programy biurowe, internetowe, proste gry i narzędzia systemowe. Najbardziej znane dystrybucje to: Ubuntu, Red Hat, SUSE, Debian. Są one dostępne bezpłatnie, można je np. pobrać z Internetu.



Rys. 6. *Środowisko graficzne Gnome 2.11 dla systemu Linux*



Rys. 7. *Środowisko graficzne Gnome 3.30 dla systemu Linux*

Linux może być sterowany za pomocą poleceń wprowadzanych przez **powłokę** (ang. *shell*) lub poprzez GUI. Najbardziej znane warianty GUI dostępne dla Linuksa to: KDE, Gnome, Xfce, Unity.

Największą popularność Linux zdobył jako system do obsługi serwerów. Jest powszechnie używany w serwisach internetowych do obsługi WWW i poczty elektronicznej. Często spotyka się go w sieciach lokalnych w funkcji serwera plików i aplikacji.

3.6. Systemy operacyjne na urządzenia mobilne

Systemy operacyjne przeznaczone na urządzenia mobilne są dostosowane do ich specyfiki. W szczególności dotyczy to sposobu wyświetlania obrazu (mały ekran o bardzo dużej rozdzielczości) oraz sposobu obsługi za pomocą dotyku i gestów (bez pośrednictwa dodatkowych urządzeń jak klawiatura lub mysz). Systemy te obsługują również rozmaite czujniki, w które wyposażone są urządzenia mobilne (np. system określania pozycji GPS, czujnik przyspieszenia, temperatury, pola magnetycznego) oraz wbudowane urządzenia wejścia, w szczególności kamerę (kamery) cyfrową, aparat fotograficzny. Ważnym elementem takiego systemu jest efektywne zarządzanie korzystaniem z energii, ponieważ urządzenia mobilne zasilane są akumulatorami.



Rys. 8. *Interfejs systemu operacyjnego Android*



Rys. 9. *Interfejs systemu operacyjnego iOS*

Najbardziej popularne systemy na urządzenia mobilne to Android (rys. 8.) i iOS (rys. 9.).

W systemach operacyjnych urządzeń mobilnych bardzo ważne jest zapewnienie bezpieczeństwa danych. Odbywa się to na kilku poziomach:

- zabezpieczenie sprzętu przed nieuprawnionym dostępem – np. za pomocą kodu PIN, hasła, kształtu rysowanego przez użytkownika, obrazu twarzy, odcisku palca;
- zabezpieczenie przed złośliwym oprogramowaniem oraz kontroli aplikacji przed ich zainstalowaniem;
- zabezpieczenie danych zapisanych na urządzeniu – za pomocą przyznawania aplikacjom odpowiednich uprawnień; uprawnienia te mogą dotyczyć zarówno dostępu do elementów sprzętowych (np. kamery, GPS-u) jak i danych zapisanych na urządzeniu (np. zdjęć, kontaktów, SMS-ów).

Programy instalowane na urządzeniach mobilnych nazywane są **aplikacjami**. Cechą charakterystyczną zarówno systemu Android, jak i iOS jest możliwość instalacji na urządzeniach z tymi systemami wyłącznie aplikacji dystrybuowanych przez cyfrowe sklepy z aplikacjami. W przypadku systemu Android sklepem tym jest Google Play, w przypadku systemu iOS – App Store. W sklepach tych znajdują się zarówno aplikacje płatne,

jak i darmowe oraz zarabiające na wyświetlaniu reklam. Aplikacje posegregowane są na różne kategorie. Przed zakupem i instalacją aplikacji można zapoznać się jej ocenami i w ten sposób zdecydować, czy dana aplikacja będzie odpowiednia.

Standardowo na urządzeniach mobilnych instalowane są aplikacje umożliwiające telefonowanie, odbieranie i wysyłanie SMS-ów i MMS-ów, tworzenie i przeglądanie kontaktów, zarządzanie terminarzem, tworzenie notatek, przeglądanie Internetu, dostęp do poczty elektronicznej. Popularnością cieszą się aplikacje do komunikacji w różnorakich formach: Snapchat, Instagram, WhatsApp, TikTok.

Dostęp do większości funkcji urządzenia mobilnego wymaga posiadania specjalnego konta u operatora systemu – firmy Google w przypadku systemu Android i Apple w przypadku systemu iOS.

4. Oprogramowanie użytkowe, narzędziowe i języki programowania

Według kryterium przeznaczenia programy komputerowe dzielimy na: systemy operacyjne, programy użytkowe, języki programowania, programy narzędziowe.

Nadrzędnym programem jest system operacyjny. Pozostałe grupy oprogramowania podano w tabeli 1.

Programy narzędziowe	Języki programowania	Programy użytkowe
narzędzia systemowe	uniwersalne	edytory tekstu
programy antywirusowe	tworzenie aplikacji internetowych	edytory grafiki, dźwięku, wideo
menedżery plików	dostęp do baz danych	przeglądarki internetowe
archiwizatory	programowanie wizualne	programy finansowo-księgowe
inne	assemblery	gry komputerowe
		inne

Tabela 1. *Schemat podziału narzędzi IT i przykłady podstawowych grup programów*



Programy i aplikacje do prawidłowego działania wymagają określonego środowiska.

Na każdy program, z którego korzystamy, należy mieć licencję.

Programy użytkowe są najczęściej tworzone dla konkretnego systemu operacyjnego, choć coraz częściej powstają wersje danego programu dla kilku systemów operacyjnych (na przykład pakiet LibreOffice, programy graficzne GIMP i Inkscape, przeglądarki internetowe Mozilla Firefox i Google Chrome czy program pocztowy Mozilla Thunderbird mają swoje wersje dla systemów Windows, macOS i Linux).

Możemy również korzystać z programów umieszczonych w chmurze obliczeniowej, zainstalowanych na **serwerach internetowych**.

Programy narzędziowe wykonują zadania związane z analizą i optymalizacją działania systemu operacyjnego lub danego urządzenia. Programy te są zwykle wbudowane w system operacyjny. Na przykład w skład systemu Windows wchodzą Defragmentator dysku, Edytor rejestru czy Menedżer urządzeń.

Do programów narzędziowych zaliczamy również programy antywirusowe, programy do nagrywania płyt CD/DVD, aplikacje zabezpieczające dane, archiwizatory.

Języki programowania służą do tworzenia programów. Każdy język programowania to zbiór określonych instrukcji i zasad składni, za pomocą których programista tworzy tzw. **kod źródłowy**. Kod źródłowy jest następnie zamieniany na **kod wynikowy** (maszynowy, czyli ciąg instrukcji procesora) przez **translator** (**kompilator** lub **interpreter**), dołączony do danego języka (więcej o tych zagadnieniach przeczytasz w temacie C1).

Przykładowe zastosowania języków programowania:

- uniwersalne – np. Python, C, C++;
- programowanie wizualne – np. Visual C, Visual Basic, Delphi;
- tworzenie aplikacji internetowych – np. Java, JavaScript, PHP;
- dostęp do baz danych – np. SQL;
- tworzenie sterowników urządzeń i oprogramowania układowego, sterującego pracą urządzeń elektronicznych.

Programy użytkowe służą do rozwiązywania problemów z różnych dziedzin i wykonywania konkretnych zadań, takich jak:

- edycja tekstu – edytory tekstu, np. Microsoft Word, LibreOffice Writer;
- wykonywanie analiz, obliczeń, wykresów – arkusze kalkulacyjne, np. Microsoft Excel, LibreOffice Calc;
- zarządzanie zbiorami danych – bazy danych, np. Microsoft Access, LibreOffice Base;
- projektowanie różnych obiektów (elewacji budynków, mostów, tuneli) – programy projektowe, np. AutoCAD;
- tworzenie rysunków, edycja zdjęć – edytory grafiki, np. Paint, GIMP, Adobe Photoshop, Corel PhotoPaint, CorelDRAW, 3D Studio Max, Inkscape;
- opracowywanie muzyki i filmów – edytory dźwięków i filmów, np. Audacity, Adobe Premiere;
- tworzenie prezentacji komputerowych – programy do tworzenia prezentacji, np. Microsoft PowerPoint, LibreOffice Impress;
- przeglądanie zasobów Internetu – przeglądarka internetowa, np. Microsoft Internet Explorer, Mozilla Firefox, Microsoft Edge, Google Chrome, Opera;
- wystawianie dokumentów sprzedaży i zakupu, dokumentów magazynowych, naliczanie podatków, wykonywanie przelewów – programy finansowo-księgowe;
- rozrywka i zabawa – gry komputerowe, w tym edukacyjne;
- nauka – programy edukacyjne.

Większość tradycyjnych programów (np. pakiety biurowe, przeglądarki internetowe, programy graficzne) ma swoje wersje na urządzenia przenośne – aplikacje mobilne. Aplikacje te są często darmowe lub można je pobrać za niewielką opłatą ze specjalnego sklepu internetowego (np. Google Play, App Store).

Obecnie coraz większą popularnością cieszy się przetwarzanie danych w chmurze obliczeniowej. Użytkownik nie nabywa programu ani nie musi go instalować na swoim komputerze – korzysta z oprogramowania zainstalowanego na serwerze internetowym, płacąc (ewentualnie) za możliwość korzystania z usługi. Przykładem tego rodzaju oprogramowania jest usługa Dokumenty Google, pozwalająca na bezpłatne korzystanie z edytora tekstu, arkusza kalkulacyjnego, bazy danych i innych programów wyłącznie za pomocą przeglądarki internetowej.

5. Przykładowe typy plików

Plik może zawierać: program lub jego część, dokument tekstowy, tabelę arkusza kalkulacyjnego, bazę danych, rysunek, zdjęcie, utwór muzyczny, film, animację.

Przykładowe **rozszerzenia** i **typy plików**:

- *exe*, *com* – pliki wykonywalne systemu Windows (uruchomienie pliku spowoduje wykonanie zawartego w nim programu);
- *bat* – tzw. plik wsadowy (wykonywalny plik tekstowy; po uruchomieniu następuje wykonanie zawartych w nim instrukcji);
- *sys*, *dll* – pliki składowe systemu operacyjnego i oprogramowania;
- *txt* – pliki tekstowe;
- *rtf*, *doc*, *docx* – pliki dokumentów tekstowych (np. programu Microsoft Word);
- *pdf* – pliki dokumentów elektronicznych;
- *zip*, *rar* – pliki zarchiwizowane (skompresowane);
- *jpg*, *gif*, *bmp* – pliki graficzne;
- *wav*, *mid*, *mp3* – pliki muzyczne;
- *dwg* – pliki programu AutoCAD;
- *htm*, *html*, *htmlx* – pliki hipertekstowe;
- *ppt*, *pptx*, *pps*, *ppsx* – pliki programu Microsoft PowerPoint;
- *xls*, *xlsx* – pliki programu Microsoft Excel.

N **Nazwa pliku**

Składa się z dwóch członów: nazwy właściwej i rozszerzenia.

R **Rozszerzenie pliku**

Określa typ pliku i zwykle wskazuje program, w którym utworzono plik.

Często tworzymy pliki zawierające teksty. Należy odróżniać **pliki tekstowe** (posiadające zwykle rozszerzenie *txt*) od **dokumentów tekstowych** (np. plików tworzonych w edytorze tekstu Microsoft Word, zapisanych z rozszerzeniem *doc* lub *docx*). Plik tekstowy zawiera wyłącznie tekst (bez formatowania, tabel, obrazów itp.), natomiast dokument tekstowy może łączyć tekst z dodatkowymi elementami.

Warto zapamiętać

- Aby korzystać ze środków IT, niezbędne są narzędzia IT, czyli oprogramowanie.
- Nadrzędnym programem, kontrolującym pracę komputera, jest system operacyjny.
- Systemy operacyjne powszechnie stosowane w obecnie używanych komputerach typu PC to: Windows, Unix, Linux. W komputerach typu Mac jest to system Mac OS/macOS.
- Zadania związane z analizą i optymalizacją działania systemu operacyjnego lub danego urządzenia wykonują programy narzędziowe.
- Do tworzenia programów komputerowych służą języki programowania.
- Do rozwiązania problemów z różnych dziedzin i wykonywania konkretnych zadań możemy dobrać odpowiednie programy użytkowe, np. do utworzenia dokumentu tekstowego – edytor tekstu, do zarządzania zbiorami danych – bazy danych, do wykonania obliczeń – arkusz kalkulacyjny itd.
- Program lub jego część oraz dokumenty tworzone w programach użytkowych (dokument tekstowy, tabelę arkusza kalkulacyjnego, bazę danych, rysunek, zdjęcie, utwór muzyczny, film, animację) zapisujemy w plikach pod określoną nazwą.
- Rozszerzenie nazwy pliku wskazuje na typ pliku i zwykle na program, w którym utworzono plik.

Pytania i polecenia

1. Jaką funkcję pełni system operacyjny?
2. Wymień powszechnie używane systemy operacyjne.
3. Opisz w punktach historię rozwoju systemu Windows.
4. Jaki system operacyjny został utworzony dla komputerów typu Mac? Omów krótko historię jego powstania.
5. Jaką wspólną cechę mają systemy Windows i Mac OS/macOS?
6. Jaki system operacyjny (która wersja) jest używany w pracowni szkolnej? Omów jego cechy.
7. Wyjaśnij znaczenie określeń: *wielozadaniowy* i *wielodostępowy*.
8. Przedstaw podział narzędzi IT.
9. Do czego służą programy narzędziowe?
10. Jaka jest funkcja języków programowania?
11. Omów przeznaczenie programów użytkowych, podając przykłady konkretnych programów.

Dla zainteresowanych

12. Oceń przyszłość systemów operacyjnych.
13. Utwórz tabelę zawierającą porównanie wszystkich systemów operacyjnych omówionych w temacie. Dobierz odpowiednie kryteria porównawcze.
14. Znajdź w Internecie informacje na temat nowości wprowadzonych w ostatniej aktualizacji systemu Windows.

Wokół dokumentów komputerowych

B

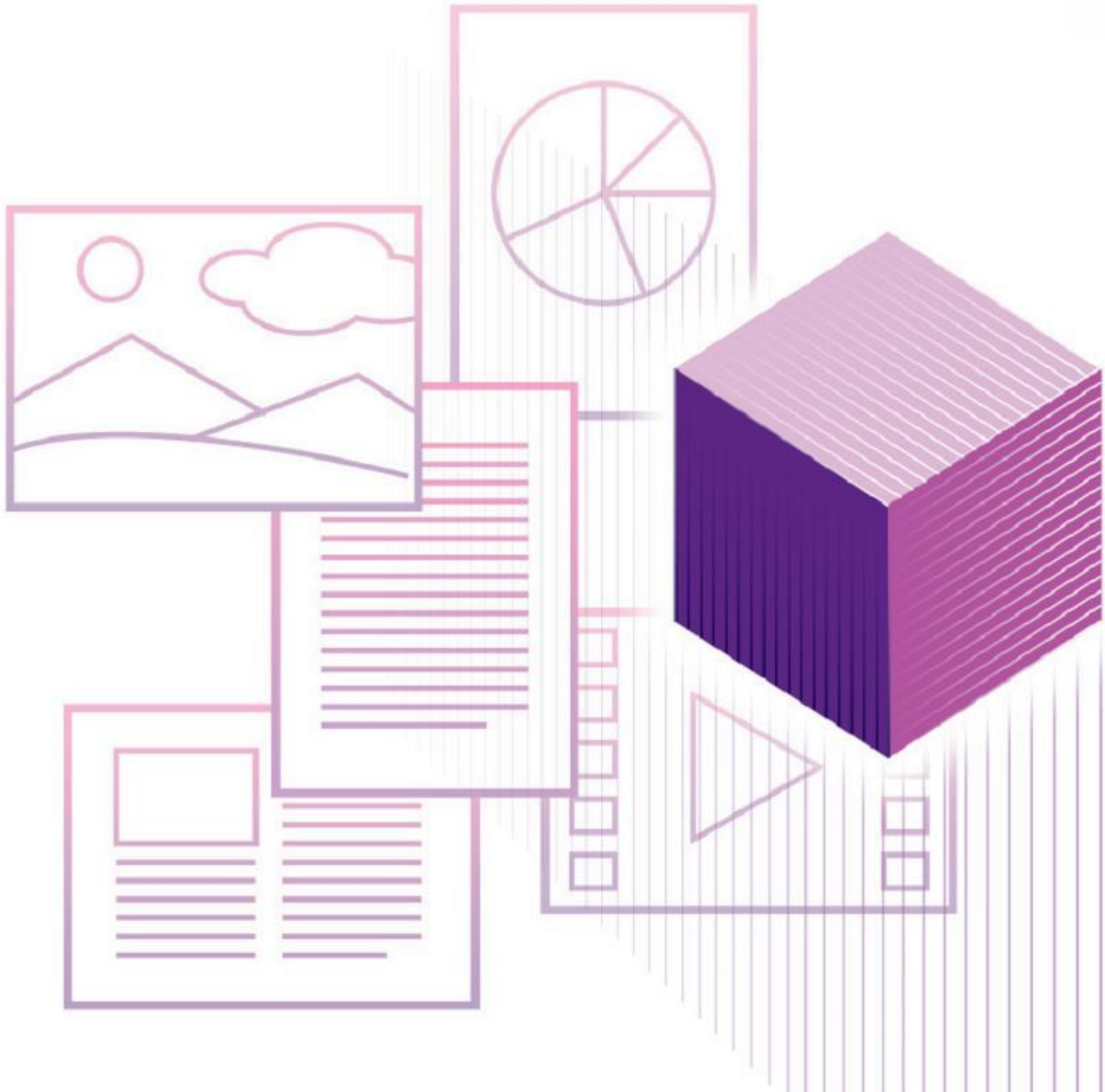
Moduł

Temat B1. Opracowywanie dokumentów tekstowych o rozbudowanej strukturze

Temat B2. Dzielenie dokumentu tekstowego i praca w trybie recenzji

Temat B3. Opracowywanie grafiki rastrowej

Temat B4. Zasady tworzenia prezentacji multimedialnej



Temat B1. Opracowywanie dokumentów tekstowych o rozbudowanej strukturze

1. Nagłówek i stopka dokumentu
2. Style tekstu
   2.1. Style standardowe
   2.2. Style niestandardowe
3. Tworzenie konspektu dokumentu
4. Odwołania w dokumencie tekstowym
   4.1. Spis treści
   4.2. Spis ilustracji, tabel i wykresów
   4.3. Przypisy

Warto powtórzyć

1. Na czym polega redagowanie i formatowanie tekstu?
2. Jakie są zasady redagowania tekstu?
3. Jakie są podstawowe sposoby formatowania tekstu?
4. Do czego służą przypisy?

1. Nagłówek i stopka dokumentu

Nagłówek i stopka N

Odpowiednio górna i dolna część strony dokumentu, w której można umieścić tekst lub inne obiekty (np. grafikę).

Informacje zawarte w nagłówku lub w stopce pojawiają się automatycznie odpowiednio w górnej i dolnej części każdej strony dokumentu. Użytkownik może m.in. zróżnicować informacje na stronach parzystych i nieparzystych lub usunąć nagłówek (stopkę) z pierwszej strony dokumentu.

W nagłówku i stopce umieszcza się m.in.: numer strony, datę utworzenia dokumentu, nazwę pliku, tytuł dokumentu lub jego części, nazwę instytucji, adres, nazwisko autora. Opcje wstawiania nagłówka i stopki możemy znaleźć na karcie **Wstawianie** lub w menu **Widok** (zależnie od wersji programu).

Ćwiczenie 1. Wpisujemy informacje do nagłówka i stopki

1. Odszukaj w Internecie treść powieści „Przedwiośnie" Stefana Żeromskiego. Skopiuj treść rozdziału pt. „Rodowód" do dokumentu tekstowego. Umieść w nagłówku dokumentu tytuł *„Przedwiośnie" – fragment* oraz nazwisko i imię autora, a w stopce – numery stron.
2. Zapisz plik pod nazwą *Przedwiośnie*.

Aby zróżnicować treść nagłówka i stopki dla stron parzystych i nieparzystych oraz dla pierwszej strony, należy (zależne od wersji programu):

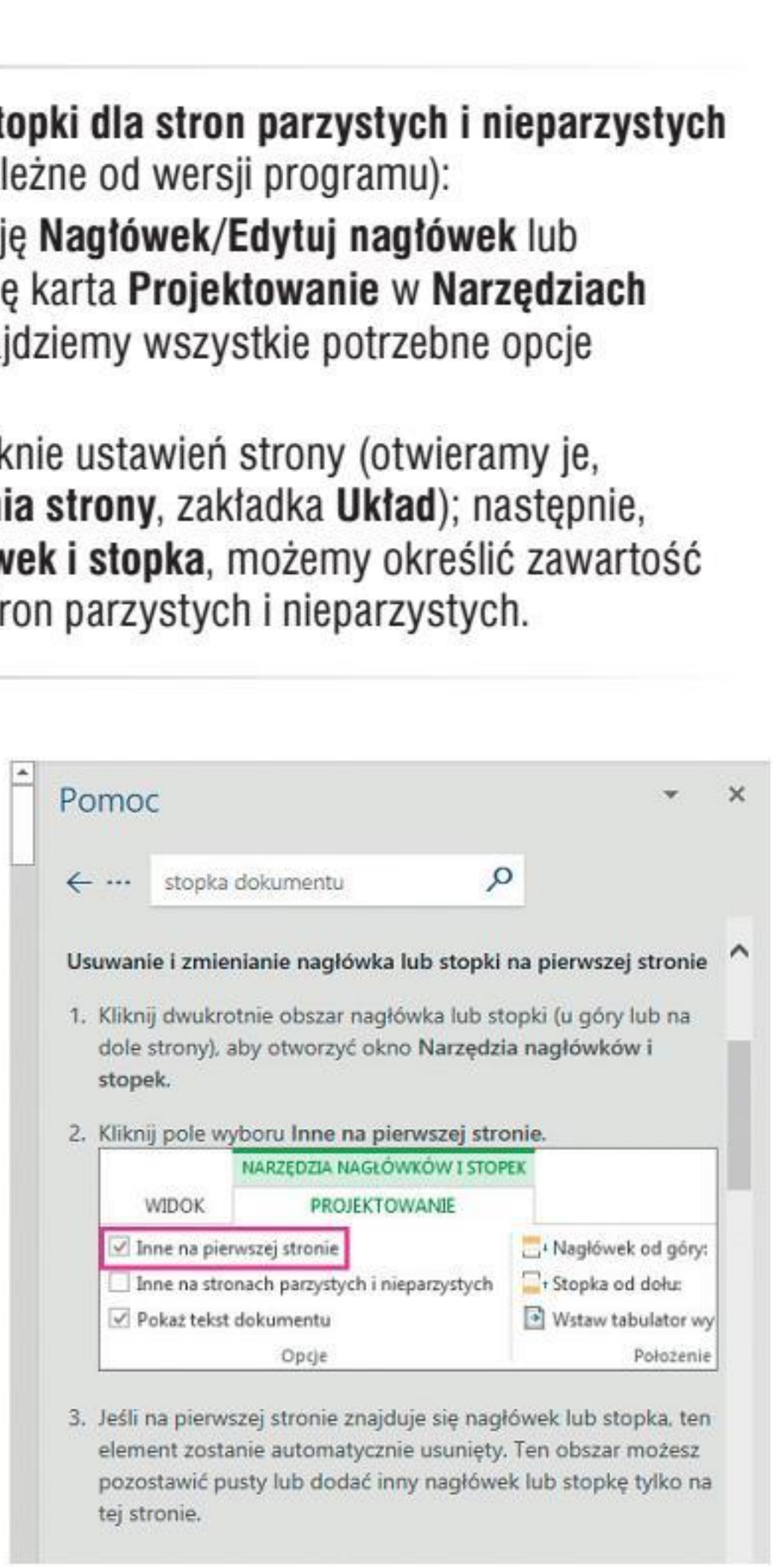
- na karcie **Wstawianie** wybrać opcję **Nagłówek/Edytuj nagłówek** lub **Stopka/Edytuj stopkę** – odsłoni się karta **Projektowanie** w **Narzędziach nagłówków i stopek**, na której znajdziemy wszystkie potrzebne opcje

lub

- zaznaczyć odpowiednią opcję w oknie ustawień strony (otwieramy je, korzystając z menu **Plik/Ustawienia strony**, zakładka **Układ**); następnie, korzystając z menu **Widok/Nagłówek i stopka**, możemy określić zawartość nagłówka i stopki oddzielnie dla stron parzystych i nieparzystych.

Ćwiczenie 2. Różnicujmy stopkę dla stron parzystych i nieparzystych oraz dla pierwszej strony

1. Otwórz plik *Przedwiośnie*, zapisany w ćwiczeniu 1.
2. Na stronach parzystych umieść numer strony w stopce po lewej stronie, a na nieparzystych – po prawej. Pomiń numer na pierwszej stronie. Informacje na temat pomijania numeru na pierwszej stronie znajdź w **Pomocy**.
3. Na stronach parzystych pozostaw ten dotychczasowy nagłówek, a na nieparzystych umieść tytuł „Rodowód".
4. Zapisz plik pod tą samą nazwą.



***Rys. 1.** Okno **Pomocy** (Microsoft Word)*

Wskazówka: Aby otworzyć **Pomoc**, można wybrać odpowiednią opcję menu lub nacisnąć klawisz **F1**. W polu tekstowym okna **Pomocy** należy wpisać odpowiednie hasło, np. „stopka dokumentu".

2. Style tekstu

2.1. Style standardowe

{ Napisaliśmy w edytorze tekstu obszerną pracę z języka polskiego. Jak najszybciej sformatować w jednakowy sposób tytuły rozdziałów i podrozdziałów? }

Do formatowania tytułów rozdziałów i podrozdziałów możemy zastosować tzw. **style nagłówkowe** – standardowe style tekstu proponowane przez program.

Styl tekstu S

Zbiór parametrów formatowania tekstu, takich jak: krój i rozmiar czcionki, odstęp między wierszami (interlinia), położenie tabulatorów i wielkość wcięć, sposób obramowania, sposób numerowania itp.

Akapit A

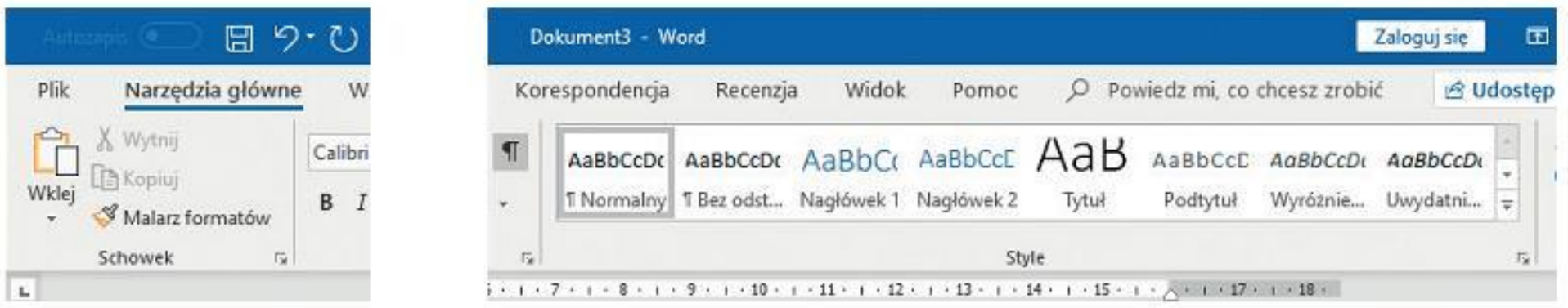
W informatyce to fragment tekstu pisany bez naciśnięcia klawisza **Enter**. Naciskając klawisz **Enter**, kończymy jeden akapit i zaczynamy następny.

Styl tekstu łączy pod jedną nazwą kilka różnych parametrów formatowania. Dzięki temu, formatując poszczególne fragmenty tekstu, nie musimy za każdym razem ustawiać wszystkich parametrów formatowania (np. kroju, rozmiaru, koloru, pogrubienia czcionki) – wystarczy zastosować zdefiniowany styl.

W nowym dokumencie domyślnym stylem jest styl o nazwie *Normalny*. Inne style można przejrzeć i wybrać, korzystając z opcji menu programu. Przykłady różnych stylów pokazano na rysunku 2. Są wśród nich style nagłówkowe (*Nagłówek 1*, *Nagłówek 2*).

Aby zastosować wybrany styl, należy:

- zaznaczyć tekst, który chcemy sformatować przy użyciu danego stylu (w przypadku niektórych stylów, np. nagłówkowych, wystarczy kliknąć dowolne miejsce **akapitu**),
- kliknąć wybrany styl w galerii stylów (rys. 2.).

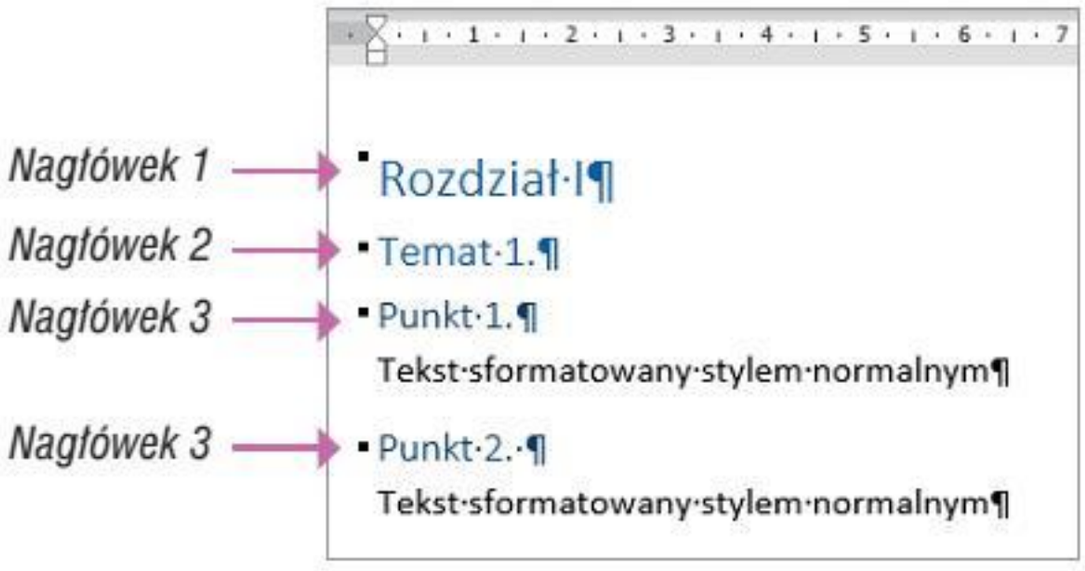


Rys. 2. *Galeria stylów w grupie* ***Style*** *na karcie* ***Narzędzia główne*** *(Microsoft Word)*

Zależnie od wersji edytora tekstu mogą być dostępne różne style, służące do formatowania m.in.: tytułu i podtytułu, nagłówków, akapitów, tabel, list numerowanych czy znaków.

Przykład 1. Stosowanie stylów nagłówkowych

Chcemy zmienić styl normalny, zastosowany do tytułów rozdziałów, tematów i punktów, na styl nagłówkowy (rys. 3.).



Rys. 3. *Zastosowanie stylów nagłówkowych*

1. Kliknij dowolne miejsce tytułu rozdziału (nie trzeba zaznaczać całego tytułu).
2. Odszukaj w zestawie stylów styl o nazwie *Nagłówek 1* i kliknij go.
3. Powtórz kroki 1. i 2., wybierając odpowiednio style *Nagłówek 2*, *Nagłówek 3*, aby sformatować nazwy tematów i punktów.

Uwaga: Styl możemy również wybrać przed rozpoczęciem pisania tekstu.

Ćwiczenie 3. Stosujemy style nagłówkowe

1. Utwórz nowy dokument tekstowy. Przepisz tytuły tematów B1 i B2 z modułu B tego podręcznika oraz punktów i podpunktów z tych tematów. Korzystając z przykładu 1., zastosuj odpowiednie style nagłówkowe do tytułów tematów, punktów i podpunktów.
2. Zapisz plik pod nazwą *Moduł B*.

Ćwiczenie 4. Stosujemy style nagłówkowe i inne dostępne w programie

1. Sprawdź, jakie style są dostępne w używanym przez ciebie programie.
2. Otwórz plik *Przedwiośnie* zapisany w ćwiczeniu 2. Na pierwszej stronie, na początku tekstu umieść tytuł: *Stefan Żeromski „Przedwiośnie"* oraz podtytuł *„Rodowód"*. Sformatuj tekst, stosując dla tytułu i podtytułu oraz dla pozostałego tekstu style wybrane spośród dostępnych w edytorze tekstu, z którego korzystasz.
3. Zapisz plik pod nazwą *Przedwiośnie ze stylami*.

2.2. Style niestandardowe

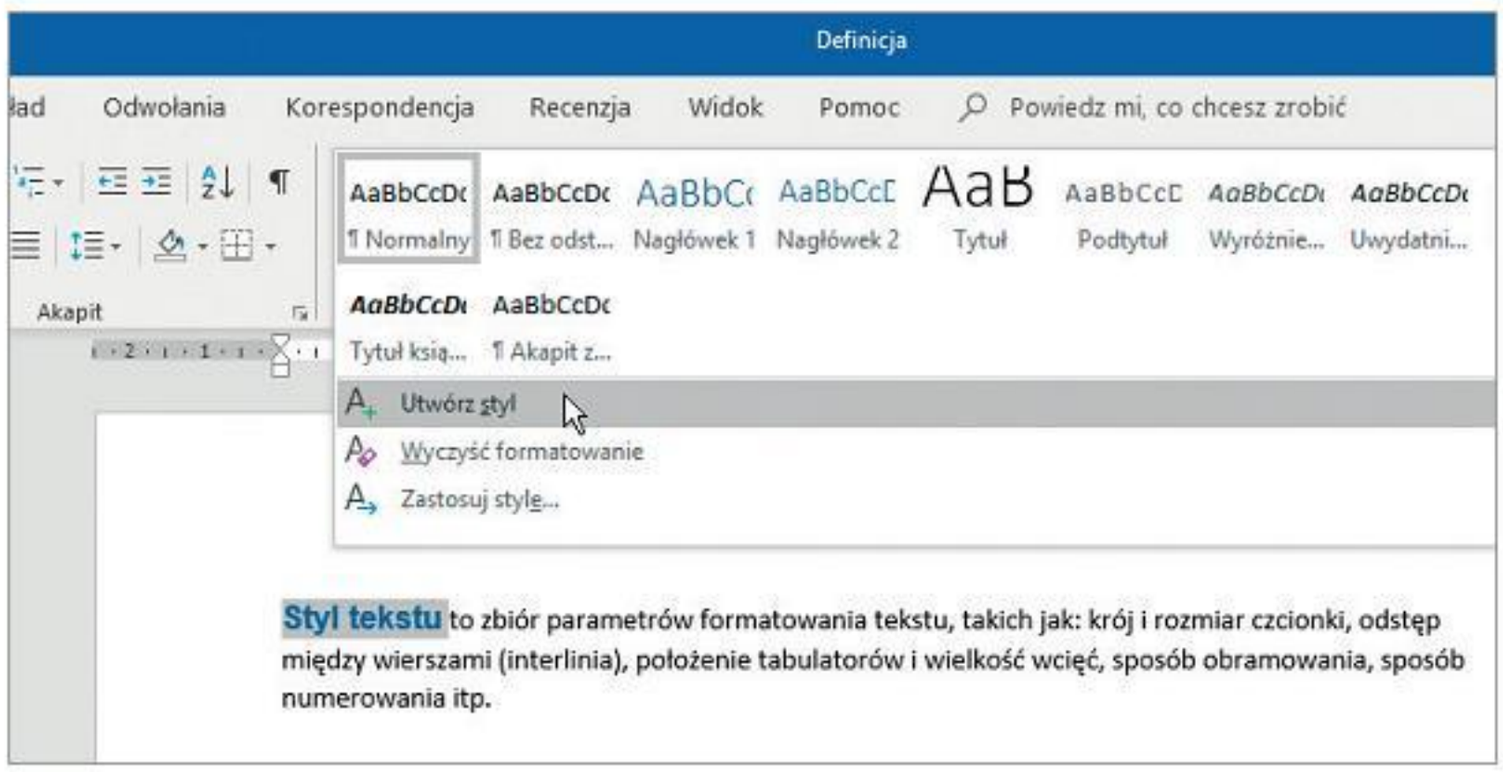
Jeśli wśród stylów proponowanych przez edytor nie znajdziemy odpowiedniego stylu, możemy utworzyć własny, niestandardowy, dostosowany do naszych potrzeb. Utworzony styl możemy zastosować w bieżącym dokumencie lub udostępnić dla nowych dokumentów opartych na tym samym szablonie.

Styl niestandardowy można przygotować przynajmniej na dwa sposoby:

1. Sformatować tekst, stosując znane parametry formatowania tekstu, określić zastosowany zbiór parametrów jako styl, nazwać go i dodać do galerii stylów.
2. Zmodyfikować istniejący styl, zmieniając parametry formatowania tekstu i nazwę stylu w oknie modyfikowania stylu.

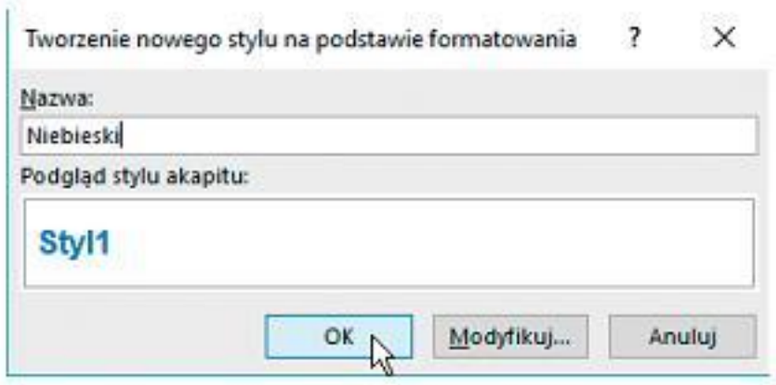
Przykład 2. Tworzenie własnego stylu na podstawie istniejącego formatowania

1. Zaznacz tekst i sformatuj go, stosując wybrane przez siebie parametry formatowania, np. pogrubioną czcionkę Arial o rozmiarze 12 punktów, w kolorze niebieskim.
2. Rozwiń listę wyboru galerii stylów i kliknij polecenie **Utwórz styl** (rys. 4a).

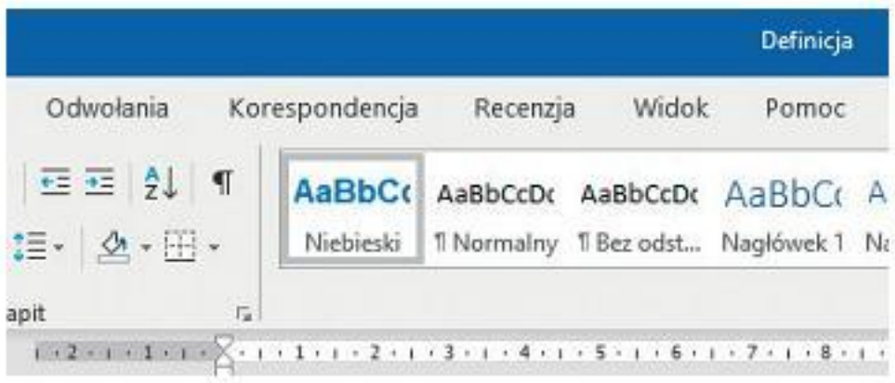


Rys. 4a. *Tworzenie własnego stylu na podstawie istniejącego formatowania (Microsoft Word)*

3. W otwartym oknie **Tworzenie nowego stylu na podstawie formatowania** (rys. 4b) nazwij styl, np. *Niebieski*, i kliknij przycisk **OK**.
4. Utworzony styl (tu: *Niebieski*) pojawi się w galerii stylów (rys. 4c) – można go użyć do formatowania innych fragmentów tekstu w tworzonym dokumencie.



Rys. 4b. *Okno* ***Tworzenie nowego stylu na podstawie formatowania***



Rys. 4c. *Nowy styl o nazwie* Niebieski *widoczny w galerii stylów*

Ćwiczenie 5. Tworzymy nowy styl na podstawie istniejącego formatowania

1. Utwórz nowy dokument tekstowy. Napisz w nim, czym jest styl tekstu (przepisz definicję z tego tematu). Zapisz plik pod nazwą *Definicja*.
2. Korzystając z przykładu 2., utwórz nowy styl o nazwie *Niebieski*.
3. Zastosuj styl *Niebieski* do wyróżnienia słów „styl tekstu", „interlinia", „tabulatorów" i „wcięć".
4. Zapisz plik pod tą samą nazwą.

Uwaga

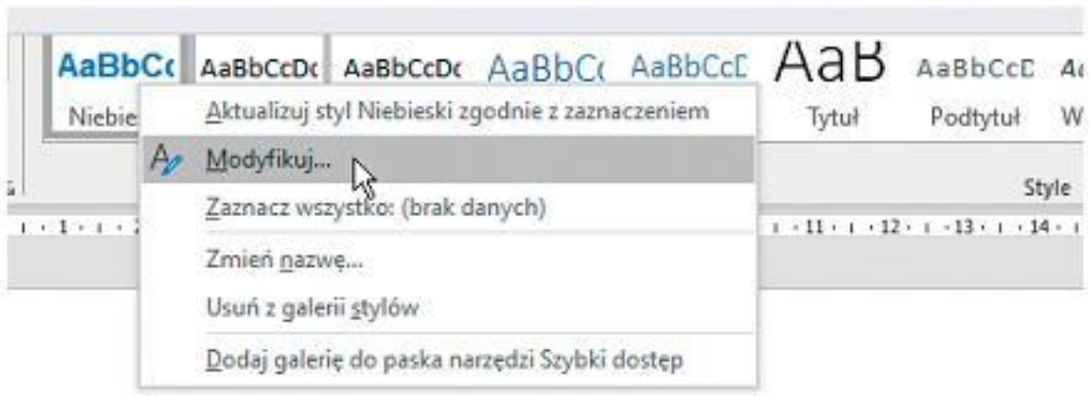
Należy pamiętać, że zmodyfikowany styl będzie miał inny, zmieniony przez nas zbiór parametrów formatowania.

Style własne (dostępne w danym dokumencie) i standardowe możemy modyfikować, np. zmienić rozmiar czcionki w utworzonym w ćwiczeniu 5. stylu *Niebieski* lub zmienić kolor czcionki w standardowym stylu *Nagłówek 1*. Zmodyfikowany styl możemy zapisać pod tą samą lub nową nazwą.

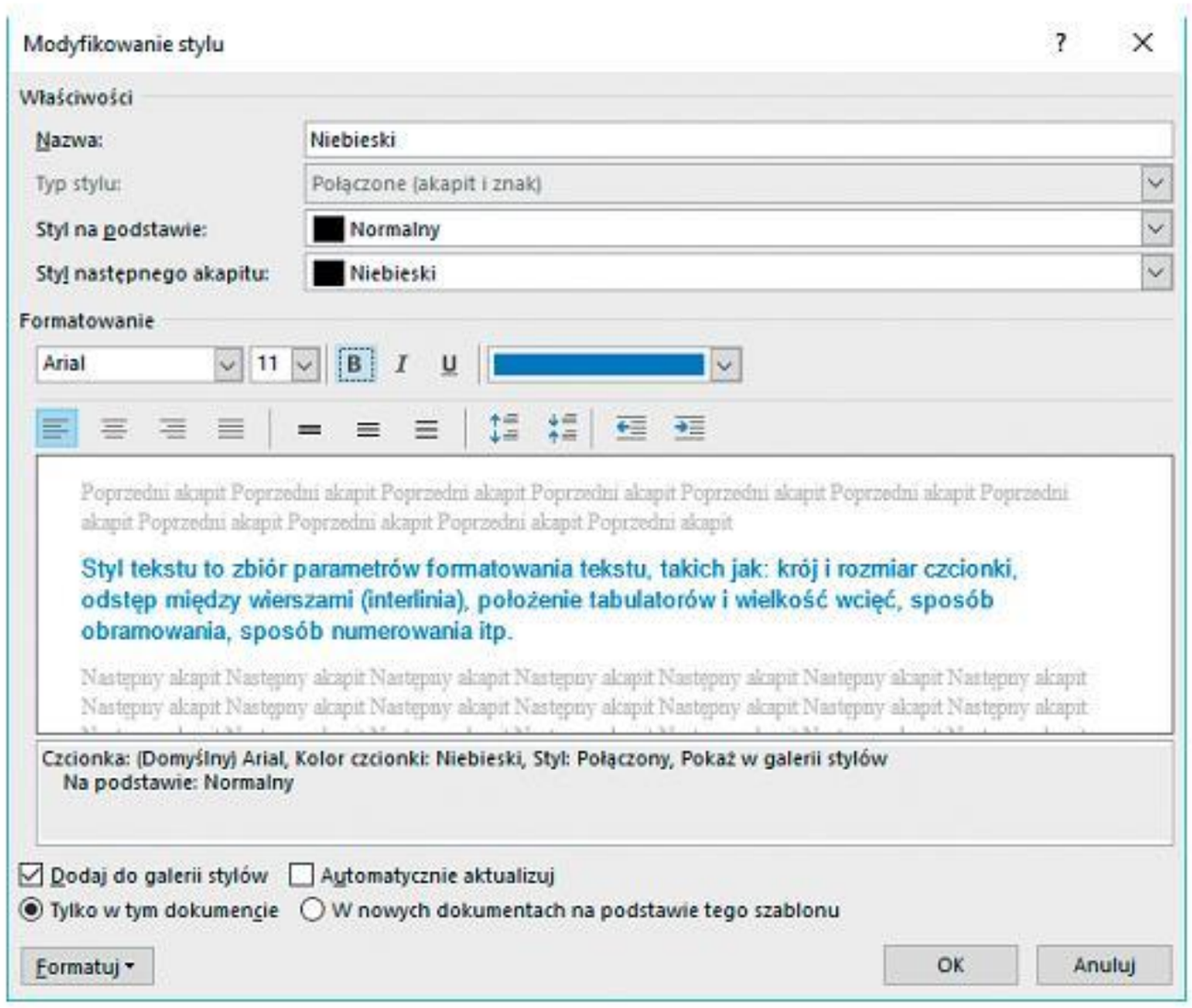
Przykład 3. Modyfikowanie stylu

1. Kliknij prawym przyciskiem myszy wybrany styl w galerii stylów i wybierz polecenie **Modyfikuj** (rys. 5a).
2. W otwartym oknie **Modyfikowanie stylu** zmień parametry formatowania (rys. 5b), a następnie:
   - kliknij przycisk **OK** – nazwa stylu pozostanie niezmieniona

   lub
   - zmień nazwę w polu **Nazwa** i kliknij przycisk **OK** – styl zostanie zapisany pod nową nazwą.



Rys. 5a. *Modyfikowanie stylu (Microsoft Word)*



Rys. 5b. *Okno* ***Modyfikowanie stylu***

Ćwiczenie 6. Modyfikujemy styl

1. Otwórz plik *Definicja*, zapisany w ćwiczeniu 5. Zmodyfikuj styl *Niebieski* – zrezygnuj z pogrubienia i zmień rozmiar czcionki na 11 punktów. Zapisz styl pod tą samą nazwą. Co zauważasz?
2. Zapisz plik pod tą samą nazwą.

Wszystkie fragmenty, w których zastosowano utworzony styl, po zmodyfikowaniu stylu są automatycznie formatowane z wykorzystaniem nowych parametrów.

Styl·tekstu·to·zbiór·parametrów·formatowania·tekstu,·takich·jak:·krój·i·rozmiar·czcionki,·odstęp·między·wierszami·(**interlinia**),·położenie·**tabulatorów**·i·wielkość·**wcięć**,·sposób·obramowania,·sposób·numerowania·itp.¶

Rys. 6a. *Tekst po zastosowaniu stylu* Niebieski *– ćwiczenie 5.*

Styl·tekstu·to·zbiór·parametrów·formatowania·tekstu,·takich·jak:·krój·i·rozmiar·czcionki,·odstęp·między·wierszami·(interlinia),·położenie·tabulatorów·i·wielkość·wcięć,·sposób·obramowania,·sposób·numerowania·itp.¶

Rys. 6b. *Tekst po zastosowaniu zmodyfikowanego stylu* Niebieski *– ćwiczenie 6.*

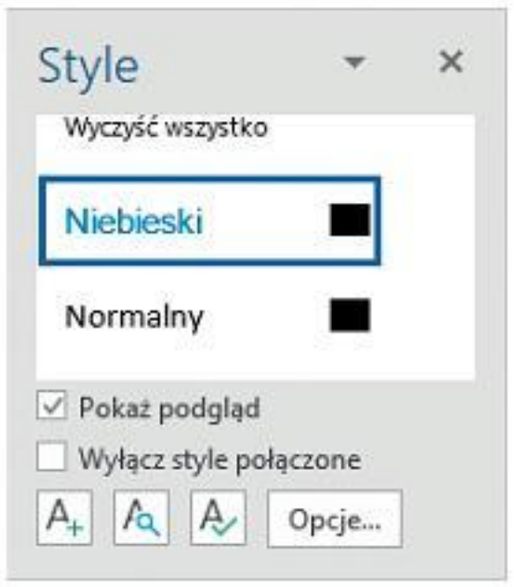
Ćwiczenie 7. Udostępniamy styl w nowych dokumentach

Dobra rada

Przed udostępnieniem stylu w nowych dokumentach warto się zastanowić, czy zmodyfikowany styl (zwłaszcza standardowy) będzie przydatny tylko w redagowanym dokumencie, czy również w nowych dokumentach.

1. Otwórz plik *Definicja*, zapisany w ćwiczeniu 6. Udostępnij styl *Niebieski* w nowych dokumentach (poszukaj tej możliwości w oknie **Modyfikowane stylu**).
2. Utwórz nowy dokument tekstowy. Przepisz treść przykładu 3. z tego tematu. Zastosuj styl *Niebieski* do wyrazów wyróżnionych pogrubieniem.
3. Zapisz dokument tekstowy w pliku pod nazwą *Przykład 3*.

W galerii stylów są umieszczone najczęściej wykorzystywane style. Aby zobaczyć wszystkie style, należy otworzyć okienko **Style**, klikając przycisk . W razie potrzeby wybrany styl można przenieść do galerii stylów lub go z niej usunąć.



Rys. 7. *Okienko* ***Style***

Aby usunąć styl z galerii stylów, należy z menu kontekstowego stylu wybrać polecenie **Usuń z galerii stylów**. Usunięcie stylu z galerii nie usuwa go z wykazu stylów w okienku **Style**.

Ćwiczenie 8. Przenosimy wybrany styl do galerii stylów i usuwamy styl z galerii

1. Korzystając z **Pomocy**, odszukaj możliwość przenoszenia wybranego stylu do galerii stylów.
2. Przenieś wybrany styl do galerii stylów, a następnie usuń go z galerii. Sprawdź, czy dany styl jest widoczny w okienku **Style**.

3. Tworzenie konspektu dokumentu

W edytorze tekstu możemy stosować różne sposoby wyświetlania dokumentu (dostępne w menu **Widok**). Na przykład, jeśli chcemy wiedzieć, w jaki sposób tekst, grafika i inne elementy zostaną rozmieszczone na wydrukowanej stronie, wybieramy **Układ wydruku**. Jeśli oprócz treści dokumentu chcemy również zobaczyć jego strukturę – wybieramy **Konspekt**.

Każdy dokument programu Microsoft Word może być wyświetlony w widoku **Konspektu**. Jednak korzystanie z udogodnień związanych z pracą w tym widoku jest możliwe dopiero po odpowiednim sformatowaniu tekstu (tytuły rozdziałów i podrozdziałów sformatowane przy użyciu stylów nagłówkowych).

Możliwość tworzenia konspektu przydaje się podczas redagowania dłuższych prac, które często mają budowę hierarchiczną, składającą się m.in. z: rozdziałów, tematów, punktów, podpunktów.

W widoku **Konspektu** można zwinąć treść akapitów, aby zobaczyć tylko nagłówki na wybranym poziomie, a następnie rozwinąć, aby zobaczyć wszystkie akapity. Można również zmienić kolejność nagłówków i fragmentów tekstu czy zmienić poziom nagłówka bądź tekstu. Dzięki tym możliwościom znacznie łatwiej jest planować i modyfikować strukturę dokumentu.

Gdy w nowym, pustym dokumencie przełączymy się na widok **Konspektu**, program Microsoft Word automatycznie nada pierwszemu akapitowi styl nagłówka – *Nagłówek 1*.

Ćwiczenie 9. Oglądamy dokument w widoku **Konspektu**

1. Otwórz plik *Moduł B,* zapisany w ćwiczeniu 3. i wyświetl dokument w widoku **Konspektu**. Zwróć uwagę na symbole konspektu przedstawiające sposób organizacji dokumentu.
2. Zwiń strukturę dokumentu, aby zobaczyć tylko nagłówki na dwóch poziomach. Dodaj tytuł tematu B3, korzystając ze spisu treści tego podręcznika. Następnie rozwiń całą strukturę i dodaj punkty oraz podpunkty do dodanego tematu, odpowiednio wszystko formatując. Przejdź do widoku **Układ wydruku**.
3. Zapisz plik pod tą samą nazwą.

Ćwiczenie 10. Modyfikujemy dokument w widoku **Konspektu**

Otwórz plik *Moduł B*, zapisany w ćwiczeniu 9. Wyświetl dokument w widoku **Konspektu**.

1. W temacie B1 usuń punkt „Style tekstu”, a punkty „Style standardowe” i „Style niestandardowe” przenieś na wyższy poziom.
2. Zmień kolejność punktów w temacie B2 – punkt „Szablony dokumentów” zamień z punktem „Praca w trybie recenzji”.
3. Rozwiń dokument i sprawdź, czy wszystkie polecenia zostały poprawnie wykonane.
4. Zapisz plik pod tą samą nazwą.

! Uwaga

Tekst podrzędny, zamieszczony pod przenoszonym nagłówkiem, przenosi się razem z nim.

4. Odwołania w dokumencie tekstowym

W rozbudowanej pracy z języka polskiego, składającej się z kilku rozdziałów, zawierającej rysunki i tabele chcielibyśmy umieścić spisy treści rozdziałów, rysunków i tabel. Jak przygotować dokument tekstowy, aby te spisy utworzyć automatycznie?

Edytor tekstu umożliwia m.in.: wstawianie do dokumentu tekstowego spisu treści, umieszczanie podpisów pod ilustracjami, tabelami i wykresami oraz tworzenie ich spisu. Można też wstawiać indeksy, przypisy, cytaty, bibliografię.

Niektóre edytory tekstu, zwłaszcza starsze wersje, mogą nie posiadać wszystkich wymienionych możliwości. W programie Microsoft Word należy ich szukać na karcie **Odwołania** lub w opcji **Wstaw/Odwołanie** (zależnie od wersji programu).

O Odwołania

Umożliwiają przechodzenie do konkretnych fragmentów tekstu i obiektów w ramach jednego dokumentu.

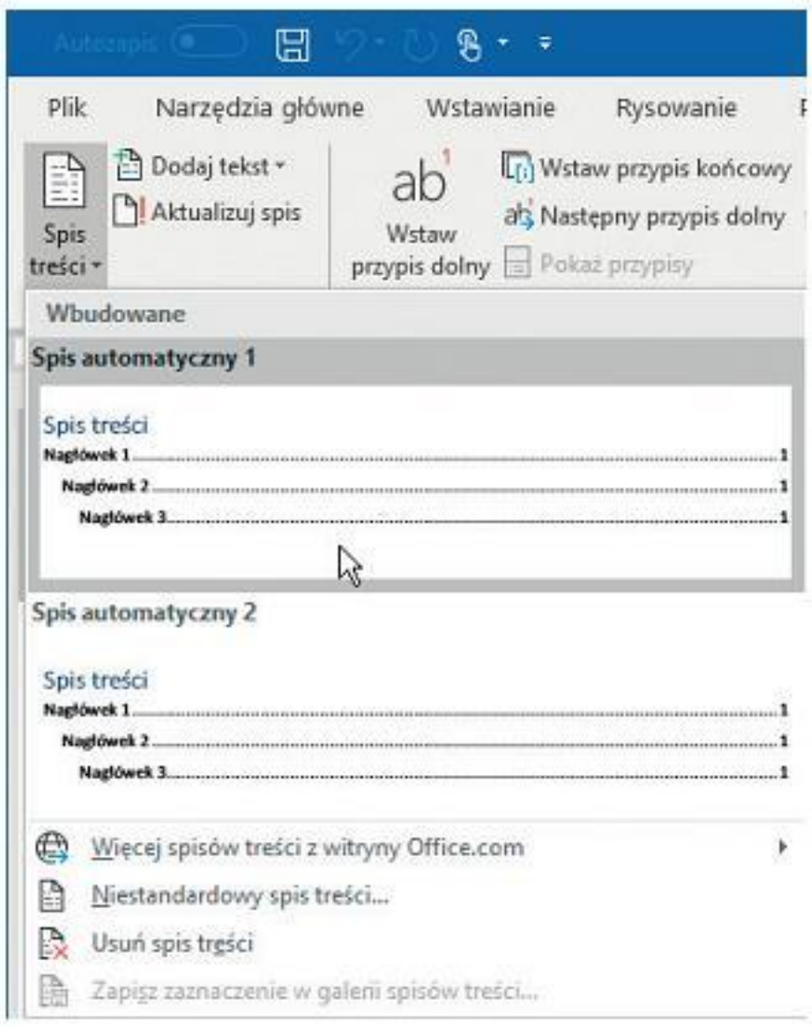
4.1. Spis treści

Do dokumentu tekstowego możemy automatycznie wstawić spis treści, pod warunkiem, że wcześniej odpowiednio przygotujemy tekst – do formatowania tytułów rozdziałów, punktów itd. użyjemy stylów nagłówkowych.

W czasie tworzenia spisu treści tytuły o odpowiednio ustalonym stylu są porządkowane ze względu na przypisany im poziom, po czym zostaje im przypisany numer strony, na której występują w tekście. Nawet jeśli w dokumencie nie zostały wstawione numery stron, pojawią się one w spisie treści.

Aby wstawić spis treści, należy:

- sformatować stylami nagłówkowymi punkty, które chcemy umieścić w spisie treści,
- ustawić kursor w miejscu, w którym chcemy umieścić spis treści,
- na karcie **Odwołania** lub w opcji **Wstaw/Odwołanie** (zależnie od wersji programu) wybrać funkcję wstawiania spisu treści,
- kliknąć styl (format) spisu treści wybrany z proponowanych przez program (rys. 8.).



***Rys. 8.** Wstawianie spisu treści (Microsoft Word) – rozwinięta lista wyboru z różnymi stylami spisów treści*

Ćwiczenie 11. Tworzymy spis treści

1. Odszukaj opcję wstawiania spisu treści w edytorze tekstu, którego używasz. W razie problemów skorzystaj z **Pomocy**.
2. Otwórz plik *Moduł B*, zapisany w ćwiczeniu 10. i wstaw na początku dokumentu spis treści.
3. Zapisz plik pod tą samą nazwą.

W spisie treści pozycje spisu lub numery stron są traktowane jak odwołania. Dzięki temu, wybierając pozycję ze spisu, można szybko przemieścić się do odpowiedniego fragmentu dokumentu.

Aby przenieść się do wybranego miejsca w dokumencie, należy:

- trzymając wciśnięty klawisz **Ctrl**, kliknąć daną pozycję w spisie treści

lub

- kliknąć daną pozycję w spisie treści i nacisnąć klawisz **Enter**.

Aby zaktualizować spis treści po zmianie lub usunięciu wybranego tytułu w treści dokumentu, należy kliknąć w obszarze spisu i z menu kontekstowego wybrać opcję **Aktualizuj pole** lub nacisnąć klawisz **F9**. W zależności od potrzeb można aktualizować cały spis treści lub tylko numery stron.

Ćwiczenie 12. Aktualizujemy spis treści

1. Otwórz plik *Moduł B*, zapisany w ćwiczeniu 11.
2. Zmień w temacie B1 tytuł punktu „Odwołania w dokumencie tekstowym" na „Stosowanie odwołań w dokumencie tekstowym", a tytuł podpunktu „Spis ilustracji, tabel i wykresów" na „Spis rysunków, tabel i wykresów". Zaktualizuj spis treści.
3. Dodaj w treści dokumentu numerowanie punktów i podpunktów (1., 2., 3., 4., 4.1., 4.2., 4.3.). Ponownie zaktualizuj spis treści.
4. Zapisz plik pod tą samą nazwą.

Uwaga

Jeśli na karcie **Widok** w grupie **Pokazywanie** klikniemy pole wyboru przy poleceniu **Okienko nawigacji**, struktura dokumentu będzie widoczna w oddzielnym panelu.

4.2. Spis ilustracji, tabel i wykresów

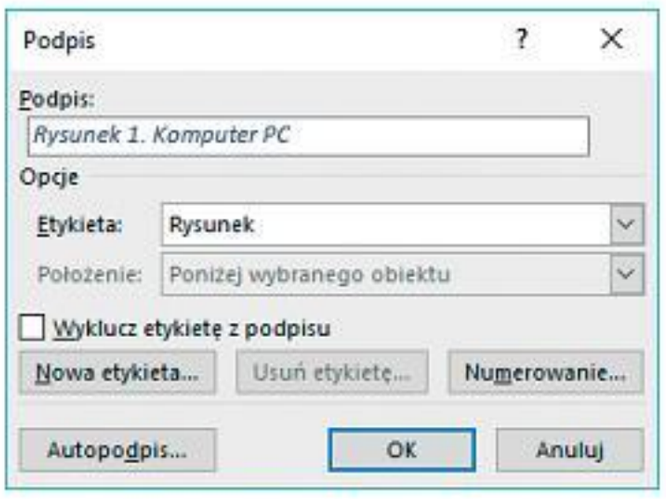
Do dokumentu tekstowego możemy automatycznie wstawić spis ilustracji, pod warunkiem, że w odpowiedni sposób wstawimy podpisy pod ilustracjami.

Polecenia wstawiania podpisu oraz spisów ilustracji, tabel i wykresów znajdziemy w tej samej opcji, co polecenie wstawiania spisu treści – na karcie **Odwołania** lub w opcji **Wstaw/Odwołanie** (zależnie od wersji programu).

Przykład 4. Wstawianie podpisu pod ilustracją

1. Uaktywnij rysunek lub ustaw kursor tekstowy w miejscu, w którym chcesz umieścić podpis.
2. Wybierz opcję wstawiania podpisu, np. **Wstaw podpis**.
3. W oknie **Podpis** wybierz m.in. rodzaj etykiety (np. *Rysunek*) i sposób numerowania (np. 1, 2, 3, …). Następnie wpisz treść podpisu w polu **Podpis**, np. *Komputer PC*. Etykietę z numerem i tekst podpisu oddziel kropką (rys. 9.).

Po wstawieniu podpisów pod każdą ilustracją (tabelą, wykresem) będzie umieszczona treść podpisu oraz etykieta wraz z numeracją, np. *Rysunek 1., Rysunek 2., ..., Tabela 1., Tabela 2., ..., Wykres 1., Wykres 2., ….*



Rys. 9. *Okno* ***Podpis*** *(Microsoft Word)*

Ćwiczenie 13. Umieszczamy podpisy pod ilustracjami

1. W nowym dokumencie tekstowym umieść krótkie opisy komputerów przenośnych, tj. notebook, tablet, smartfon. Przed każdym opisem dodaj tytuł (nazwę komputera).
2. Obok każdego opisu wstaw odpowiednie zdjęcie. Opisy i zdjęcia komputerów znajdź w Internecie. Źródła informacji, czyli adresy stron internetowych, z których zaczerpniesz informacje o każdym z komputerów, wklej na końcu dokumentu i opisz, jakiego komputera dotyczą.
3. Pod każdą ilustracją wstaw podpis według wzoru pokazanego na rys. 10.
4. Zapisz opis komputerów w pliku pod nazwą *Komputery*.

Uwaga

Spis ilustracji można aktualizować w taki sam sposób, jak spis treści.



***Rys. 10.** Przykład podpisanej ilustracji*

Aby wstawić do dokumentu tekstowego spis ilustracji (tabel, wykresów itp.), należy:

- pod ilustracjami (tabelami, wykresami itp.) umieścić podpisy,
- ustawić kursor w miejscu, w którym powinien być umieszczony spis ilustracji,
- wybrać opcję wstawiania spisu ilustracji.

W spisie ilustracji, podobnie jak w spisie treści, każda pozycja jest traktowana jak odwołanie, dzięki czemu po jej kliknięciu możemy przemieścić się do odpowiedniej ilustracji.

Ćwiczenie 14. Tworzymy spis ilustracji

1. Otwórz plik *Komputery*, zapisany w ćwiczeniu 13. Sformatuj odpowiednio tytuły. Utwórz spis treści i wstaw na początku tekstu. Utwórz spis ilustracji i wstaw go pod spisem treści.
2. Zapisz plik pod tą samą nazwą.

Ćwiczenie 15. Tworzymy spis tabel

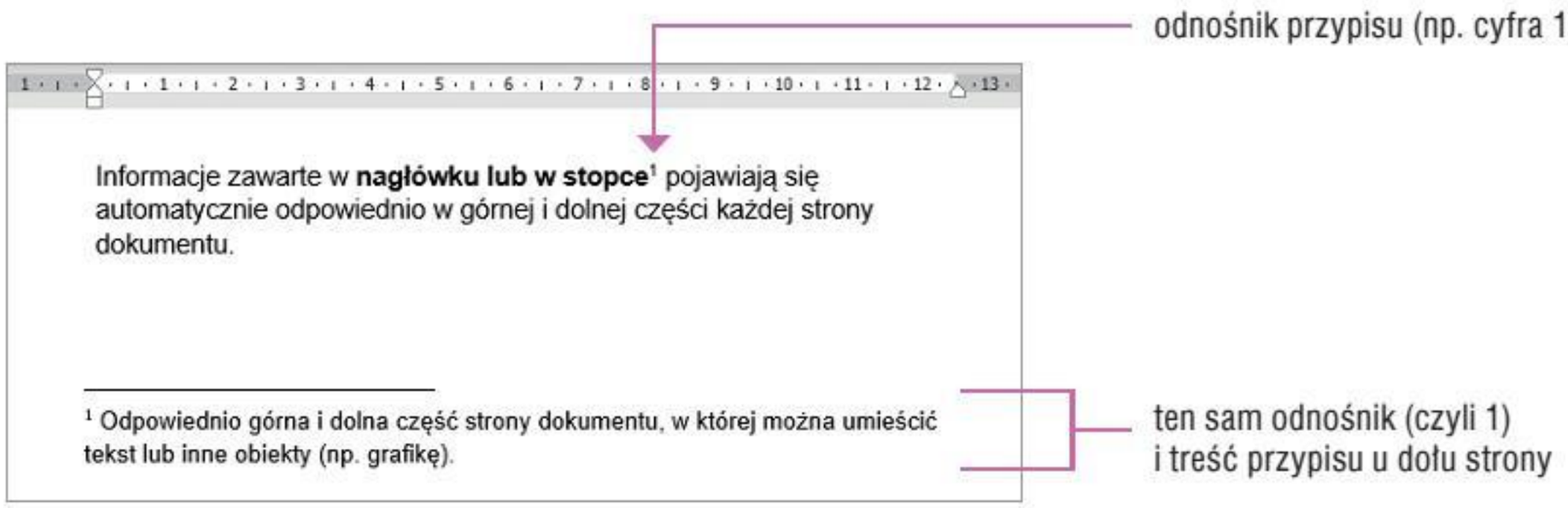
1. Otwórz plik *Komputery*, zapisany w ćwiczeniu 14. Przygotuj trzy tabele, umieszczając je po informacjach o każdym z rodzajów komputerów. W każdej tabeli umieść przykładowe dane każdego z komputerów (rys. 11.). Przykładowe informacje znajdź w Internecie.
2. Pod każdą tabelą umieść podpis, np. „Tabela 1. Przykładowe dane laptopa".
3. Utwórz spis tabel i umieść go pod spisem ilustracji.
4. Zapisz plik pod tą samą nazwą.

Laptop	
Przekątna ekranu	15,6 cali
System operacyjny	Windows 10
Wielkość pamięci RAM	8 GB
Typ procesora	

Rys. 11. *Przykładowa tabela – ćwiczenie 15.*

4.3. Przypisy

Przypisy stosuje się w celu podania źródła informacji lub zapisania komentarza czy wyjaśnienia. Istnieją dwa rodzaje przypisów: dolne (pojawiają się u dołu danej strony) i końcowe (pojawiają się na końcu całego tekstu dokumentu). Przypis wstawia się, wybierając odpowiednią opcję na karcie **Odwołania** lub w opcji **Wstaw/Odwołanie** (zależnie od wersji programu). Po jej wybraniu w miejscu położenia kursora tekstowego pojawi się odnośnik do wstawionego przypisu. Ten sam odnośnik pojawia się u dołu strony lub na końcu dokumentu (zależnie od rodzaju przypisu) i w tym miejscu wprowadzamy treść przypisu (rys. 12.).



Rys. 12. *Zastosowanie przypisu dolnego*

Treść przypisu jest pokazywana przy odnośniku po wskazaniu odnośnika kursorem myszy. Dwukrotne kliknięcie odnośnika przypisu przenosi nas do treści przypisu (odnośnik odwołuje się do tego przypisu).

Ćwiczenie 16. Stosujemy przypisy

1. Otwórz plik *Komputery*, zapisany w ćwiczeniu 15. W programie, którego używasz, odszukaj opcję wstawiania przypisu.
2. Utwórz przypisy dolne do podpisów pod zdjęciami lub do wybranych zdań opisu lub tytułów, wskazując źródło. Skorzystaj z adresów stron internetowych umieszczonych na końcu dokumentu (w ćwiczeniu 13.). Po wstawieniu źródeł informacji (adresów stron) w przypisach, usuń je z dokumentu.
3. Zapisz plik pod tą samą nazwą.

Wskazówka: Odnośnik przypisu należy umieścić na końcu fragmentu tekstu, którego przypis ma dotyczyć.

Warto zapamiętać

- Jeśli tworzymy dokumenty tekstowe o rozbudowanej strukturze, warto korzystać z różnych sposobów wyświetlania dokumentu, w szczególności z widoku **Konspektu**.
- W stopce lub w nagłówku należy umieszczać informacje, które powinny znaleźć się na wszystkich stronach dokumentu (np. numery stron).
- Dla zapewnienia jednolitego wyglądu odpowiednich fragmentów dokumentu warto korzystać ze standardowych stylów tekstu i tworzyć własne.
- W dokumencie można tworzyć spisy treści, ilustracji i tabel oraz wstawiać przypisy.

Pytania i polecenia

1. Jakie informacje umieszcza się zwykle w nagłówku, a jakie w stopce dokumentu tekstowego?
2. Czym jest styl tekstu?
3. W jaki sposób stosuje się style standardowe, dostępne w edytorze tekstu?
4. Do czego można zastosować style nagłówkowe?
5. W jaki sposób można utworzyć styl niestandardowy?
6. Korzystając z **Pomocy** programu Microsoft Word, omów różne sposoby wyświetlania dokumentu tekstowego. Wyjaśnij na przykładach zastosowanie poszczególnych widoków.
7. W jaki sposób należy przygotować konspekt dokumentu? Jakie są zalety konspektu?
8. W jaki sposób należy przygotować dokument, aby można było utworzyć spis treści?
9. Co oznacza fakt, że pozycja spisu treści jest jednocześnie odwołaniem?
10. W jaki sposób należy przygotować spis ilustracji i tabel umieszczonych w dokumencie?
11. W jaki sposób należy wstawić przypis dolny?

Zadania

1. Otwórz plik *Komputery* zapisany w ćwiczeniu 16. Umieść w nagłówku dokumentu swoje imię i nazwisko wyrównane do prawej strony na stronach nieparzystych, a do lewej – na parzystych. W stopce dokumentu umieść numery stron – wyrównaj je do środka. Pomiń numer na pierwszej stronie. Zapisz plik pod tą samą nazwą.
2. Otwórz plik *Definicja* zapisany w ćwiczeniu 6. Dodaj pod tekstem definicję akapitu. Zastosuj styl *Niebieski* do wyrazu „akapit". Zapisz plik pod nazwą *Definicje*.
3. Otwórz plik *Definicje* zapisany w zadaniu 2. Zmodyfikuj styl *Niebieski*, zmieniając kolor czcionki na czerwony. Nazwij zmodyfikowany styl *Czerwony*. Co zauważasz? Zapisz plik pod tą samą nazwą.
4. Utwórz nowy dokument tekstowy. Zmodyfikuj styl *Normalny*: zastosuj czcionkę Arial o rozmiarze 9 punktów i pojedynczą interlinię. Zapisz styl pod nazwą *Rozmiar9*. Zastosuj utworzony styl do zapisania odpowiedzi na pytanie 9. Zapisz plik pod nazwą *Odwołania*.
5. Otwórz plik *Odwołania* zapisany w zadaniu 4. Dodaj w kolejnym akapicie przykłady stosowania odwołań w dokumencie poznane w tym temacie. Odpowiedz na pytanie: *Dlaczego w nowym akapicie został zastosowany styl* Rozmiar9? Uzasadnij odpowiedź. Zapisz plik pod tą samą nazwą.

6. Utwórz nowy dokument tekstowy. Przygotuj plan opracowania na temat wybranych e-usług (możesz skorzystać z tematu D2). Zastosuj odpowiednie style nagłówkowe do punktów i podpunktów. Wyświetl dokument w widoku konspektu. Przygotuj i wstaw do dokumentu spis treści. Zapisz plik pod nazwą *E-usługi*.
7. Przygotuj w edytorze tekstu w postaci konspektu plan opracowania na temat budowy komputera (możesz skorzystać z tematu A1). Zapisz plik pod nazwą *Budowa komputera*.
8. Otwórz plik *Budowa komputera* zapisany w zadaniu 7. Przygotuj spis treści, stosując style nagłówkowe. Umieść spis treści na początku dokumentu. Dodaj numerowanie: dla pierwszego poziomu – *Zagadnienie 1*, *Zagadnienie 2*, …; dla drugiego – 1, 2, 3, …, a dla trzeciego – 1.1, 1.2, … . Zaktualizuj spis treści. Zapisz plik pod tą samą nazwą.
9. Utwórz nowy dokument tekstowy. Opisz w kilku zdaniach swoją wymarzoną podróż – wymień ciekawe miejsca, które planujesz zwiedzić. Znajdź w Internecie przykładowe zdjęcia przedstawiające te miejsca i wklej do dokumentu. Źródła informacji, czyli adresy stron internetowych, na których znajdziesz informacje o każdym z miejsc, wklej na końcu dokumentu i opisz, jakiego miejsca dotyczą. Pod każdym zdjęciem umieść podpis. Wstaw na końcu opisu podróży spis zdjęć. Zapisz opis podróży w pliku pod nazwą *Podróż*.
10. Otwórz plik *Podróż* zapisany w zadaniu 9. Do każdego podpisu dodaj przypis dolny z informacją o źródle zdjęcia. Skorzystaj z adresów stron internetowych umieszczonych na końcu dokumentu. Po wstawieniu źródeł informacji (adresów stron) w przypisach, usuń je końca z dokumentu. Zapisz plik pod tą samą nazwą.

Dla zainteresowanych

11. Odszukaj w menu edytora tekstu lub w **Pomocy** możliwość zablokowania funkcji autokorekty wielkiej litery na początku zdania. Zastanów się, czy warto tę funkcję zablokować.
12. Utwórz nowy dokument tekstowy. Opisz w kilku zdaniach pięć wybranych filmów z bieżącego roku. Zapisz plik pod nazwą *Filmy*. Znajdź w Internecie przykładowe zdjęcia z tych filmów, pobierz je i wklej do tekstu. Umieść pod zdjęciami podpisy. Źródła informacji, czyli adresy stron internetowych, z których zostały pobrane zdjęcia, wklej na końcu dokumentu i opisz, jakiego filmu dotyczą. Do każdego podpisu dodaj przypis dolny z informacją o źródle zdjęcia. Po umieszczeniu źródeł informacji (adresów stron) w przypisach, usuń je z końca dokumentu. Wstaw na końcu opisu filmów spis zdjęć. Zapisz opis filmów w pliku pod nazwą *Filmy*.
13. Sprawdź w **Pomocy**, w jaki sposób można wykonać indeks słów dla dokumentu tekstowego. Otwórz plik *Filmy*, zapisany w zadaniu 12. Wybierz kilka słów i przygotuj indeks. Zapisz plik pod tą samą nazwą.
14. Sprawdź, czy w używanym przez ciebie edytorze tekstu jest możliwe wstawienie do dokumentu tekstowego bibliografii, w której są umieszczone m.in. odwołania do wszystkich cytatów w tekście. Jeśli tak, w nowym dokumencie tekstowym opisz w punktach sposób postępowania.

Temat B2. Dzielenie dokumentu tekstowego i praca w trybie recenzji

1. Podział dokumentu tekstowego na strony, sekcje i kolumny
   1.1. Podział dokumentu na strony
   1.2. Podział dokumentu na sekcje
   1.3. Rozmieszczanie tekstu w kolumnach
2. Szablony dokumentów
3. Praca w trybie recenzji
   3.1. Śledzenie zmian
   3.2. Wstawianie komentarzy
   3.3. Porównywanie dokumentów
4. Praktyczny poradnik

Warto powtórzyć

1. Z jakich opcji wstawiania nagłówka i stopki dokumentu należy skorzystać, aby różniły się one treścią na stronach parzystych i nieparzystych?
2. Jakie jest zastosowanie stylów tekstu?
3. W jaki sposób umieszcza się spis treści, ilustracji i tabel w dokumencie tekstowym?

1. Podział dokumentu tekstowego na strony, sekcje i kolumny

W pracy o komputerach chcemy umieścić wybrane fragmenty tekstu na oddzielnych stronach, część tekstu ułożyć w dwóch kolumnach, a stronom zawierającym tabele nadać orientację poziomą. W jaki sposób można wymusić podział tekstu na strony? W jaki sposób umieścić na jednej stronie część tekstu w jednej kolumnie, a część – w dwóch kolumnach? Czy można w jednym dokumencie niektóre strony ułożyć poziomo, a inne pionowo?

1.1. Podział dokumentu na strony

Edytor tekstu automatycznie dzieli tekst na strony, uwzględniając aktualne ustawienia (m.in. rozmiar strony, marginesów). Czasem jednak trzeba wymusić podział na strony w konkretnym miejscu, np. na początku kolejnego rozdziału. Stosuje się wtedy **ręczny podział strony**.

Podział strony tworzymy, korzystając z odpowiedniej opcji na karcie **Wstawianie** lub w menu **Wstaw** (zależnie od wersji programu). Znak podziału zostanie wstawiony w miejscu aktualnego położenia kursora i będzie widoczny po włączeniu wyświetlania znaków niedrukowanych (czyli po kliknięciu przycisku ¶ powodującego wyświetlanie wszystkich znaków).

Ćwiczenie 1. Dzielimy ręcznie dokument na strony

1. Otwórz plik *Komputery* zapisany w zadaniu 1. z tematu B1. Włącz opcję umożliwiającą pokazywanie wszystkich znaków. Utwórz przed każdym opisem podział strony. Obejrzyj dokument w **Podglądzie wydruku**.
2. Zapisz plik pod tą samą nazwą.

1.2. Podział dokumentu na sekcje

Dzięki podziałowi na sekcje w jednym dokumencie można stosować różne parametry formatowania, np. inną szerokość marginesów, różną liczbę kolumn tekstu (nawet na tej samej stronie) czy niejednolitą orientację stron (pozioma, pionowa). Jeśli nie utworzymy podziału, edytor potraktuje cały dokument jako jedną sekcję.

Podział sekcji nie jest równoznaczny z podziałem strony (strona może być podzielona na kilka sekcji albo sekcja może składać się z wielu stron).

S **Sekcja**

Część dokumentu, wyodrębniona np. w celu zastosowania innych parametrów formatowania od użytych w pozostałych częściach tekstu.

Aby utworzyć sekcję (wstawić znak podziału sekcji), należy skorzystać z opcji **Znaki podziału** na karcie **Układ** w grupie **Ustawienia strony** lub w menu **Wstaw** (zależnie od wersji programu).

Aby usunąć podział strony lub sekcji, wystarczy usunąć znak podziału (tak jak każdy inny znak). Aby usunąć tego rodzaju oznaczenia z całego tekstu, najlepiej skorzystać z automatycznego wyszukiwania i zamiany znaków.

Ćwiczenie 2. Dzielimy dokument na sekcje

1. Otwórz plik *Komputery* zapisany w ćwiczeniu 1.
2. Dodaj do każdej tabeli po trzy kolumny z prawej strony i umieść w nich inne przykładowe parametry danego urządzenia. Każdą tabelę umieść na oddzielnej stronie, zmieniając orientację strony na poziomą. Tekst pozostaw na stronach w układzie pionowym.
3. Zapisz plik pod tą samą nazwą.

Wskazówki:

- Zamień wstawione w ćwiczeniu 1. podziały strony na podziały sekcji. Zwróć uwagę na różne sposoby podziału tekstu na sekcje. Wybierz **Podział sekcji/Następna strona**.
- Polecenia pozwalające zmienić układ strony, m.in. orientację, znajdziesz na karcie **Układ** lub w menu **Plik/Ustawienia strony** (zależnie od wersji programu).

1.3. Rozmieszczanie tekstu w kolumnach

Aby umieścić tekst w kolumnach, należy go zaznaczyć i wybrać odpowiednią opcję na karcie **Układ** lub z menu **Format.**

Aby umieścić tekst w kolumnach tej samej długości, należy wstawić na jego końcu **ciągły znak podziału sekcji**.

Ćwiczenie 3. Rozmieszczamy tekst w dwóch kolumnach

1. Odszukaj w edytorze tekstu opcję umieszczania tekstu w kolumnach.
2. Otwórz plik *Przedwiośnie* zapisany w ćwiczeniu 2. z tematu B1. Umieść tekst w dwóch kolumnach z odstępem 1 cm. Zadbaj, aby na końcu żadnego wiersza nie pozostały jednoliterowe spójniki i przyimki. Wyjustuj tekst w kolumnach.
3. Zapisz plik pod nazwą *Przedwiośnie kolumny*.

Wskazówka: Aby ustalić odstęp między kolumnami, należy skorzystać z opcji **Kolumny** (**Więcej kolumn**) na karcie **Układ** w grupie **Ustawienia strony**.

Dzięki sekcjom można dzielić tekst na fragmenty składające się z różnej liczby kolumn. W takim przypadku tworzymy sekcje w miejscach, w których zmienia się liczba kolumn.

Ćwiczenie 4. Rozmieszczamy tekst w kilku kolumnach

1. Otwórz ponownie plik *Przedwiośnie* zapisany w ćwiczeniu 2. z tematu B1. Wyjustuj tekst. Zmień rozmiar czcionki na 10 punktów. Włącz automatyczne dzielenie wyrazów. Na kolejnych stronach treść umieść odpowiednio w jednej, dwóch i trzech kolumnach z odstępem 1 cm. Zadbaj, aby w każdym przypadku kolumny były takiej samej długości. Czy po umieszczeniu danej strony tekstu w kolumnach treść nadal mieści się na jednej stronie?
2. Włącz podgląd wszystkich znaków. Sprawdź, jakie znaki podziału pojawiły się w miejscach, w których zmienia się liczba kolumn.
3. Usuń wszystkie jednoliterowe spójniki i przyimki, które pozostają na końcach wierszy.
4. Zapisz plik pod tą samą nazwą.

Wskazówka: Aby włączyć automatyczne dzielenie wyrazów, należy skorzystać z opcji **Dzielenie wyrazów** na karcie **Układ** w grupie **Ustawienia strony**.

2. Szablony dokumentów

Szablon S

Wzorzec podstawowej struktury dokumentu. Może określać m.in.: style, klawisze skrótów przypisane poleceniom, makra, czcionki, ustawienia marginesów, układ strony.

Dokumenty programu Microsoft Word są oparte na **szablonach**. Szablon możemy wykorzystać do tworzenia dokumentów o podobnej strukturze.

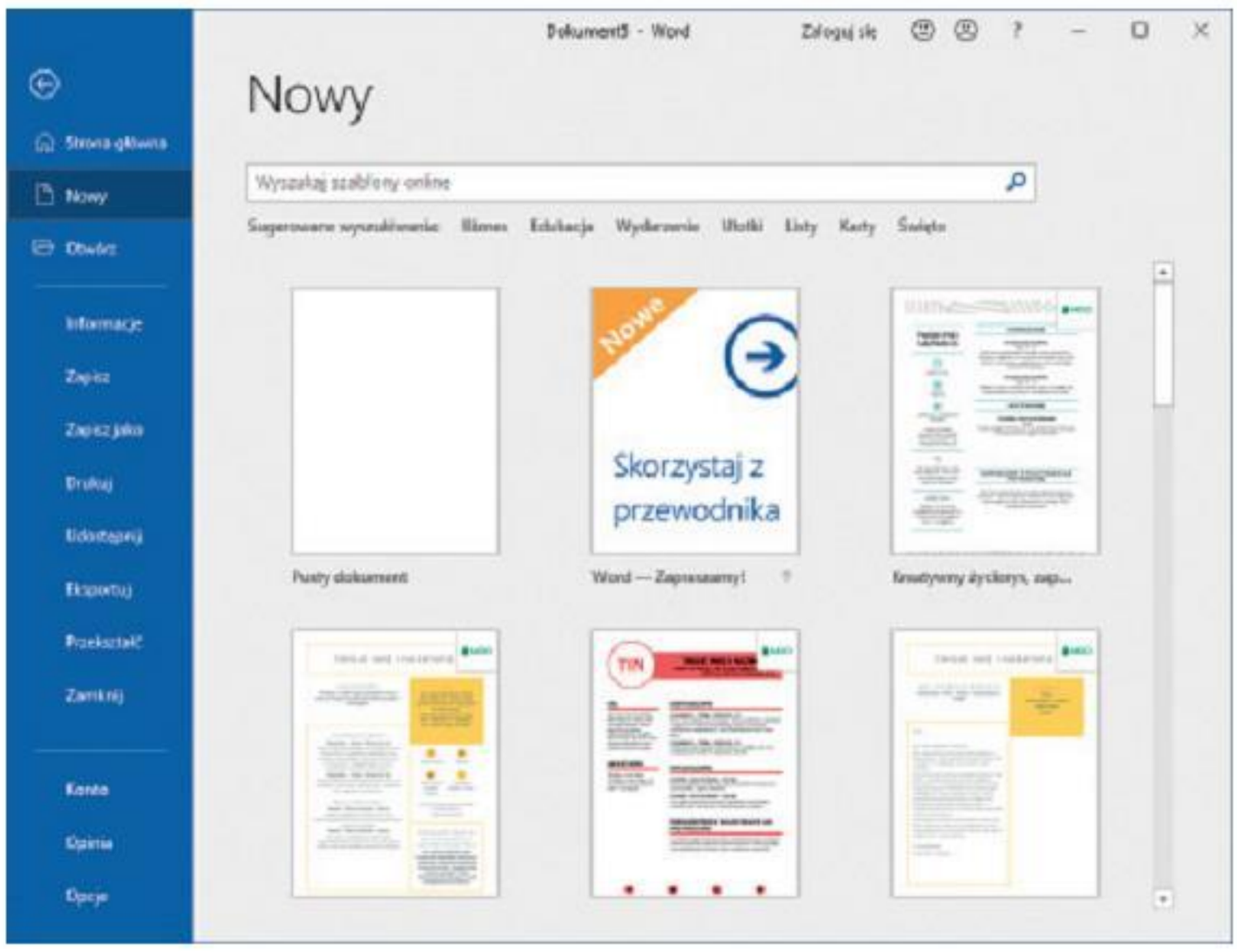
Program Microsoft Word zawiera wiele gotowych szablonów, służących m.in. do przygotowywania życiorysów, oficjalnych listów, faksów, not czy kalendarzy. Zależnie od wersji programu mogą być udostępnione różne zestawy szablonów. Szablon jest zazwyczaj zapisany jako plik z rozszerzeniem *dotx* lub *dot*.

Domyślnie każdy nowy dokument oparty jest na szablonie standardowym **Pusty dokument**.

Aby zastosować inny szablon, należy w menu **Plik** wybrać opcję tworzenia nowego dokumentu i kliknąć wybrany szablon z zestawu proponowanego przez program.

Ćwiczenie 5. Korzystamy z szablonu

1. Przejrzyj szablony proponowane przez edytor tekstu, z którego korzystasz.
2. Na podstawie wybranego szablonu przygotuj własny życiorys.
3. Zapisz plik pod nazwą *Zyciorys*.



Rys. 1. *Przykładowe szablony (Microsoft Word)*

3. Praca w trybie recenzji

W dwuosobowej grupie pracujesz nad projektem z algorytmiki wykonywanym w edytorze tekstu. Chcesz wprowadzić kilka zmian w tekście nadesłanym przez kolegę (koleżankę). Jak można sprawnie i czytelnie nanieść korektę w dokumencie tekstowym, aby zmiany były widoczne dla drugiej osoby?

Edytory tekstu umożliwiają analizowanie zmian dokonanych w dokumencie tekstowym, co znacznie ułatwia prace redakcyjne w sytuacji, kiedy tekst opracowuje kilka osób (np. podczas projektu grupowego).

W edytorze tekstu można śledzić wybrane operacje, m.in. zmiany formatowania, wstawianie, usuwanie i przenoszenie fragmentów tekstu. W dowolnym momencie pracy nad tekstem można się z tymi zmianami zapoznać oraz je zaakceptować lub odrzucić. Jeśli dokument otworzy inna osoba, również będzie mogła zobaczyć zaakceptowane i odrzucone zmiany.

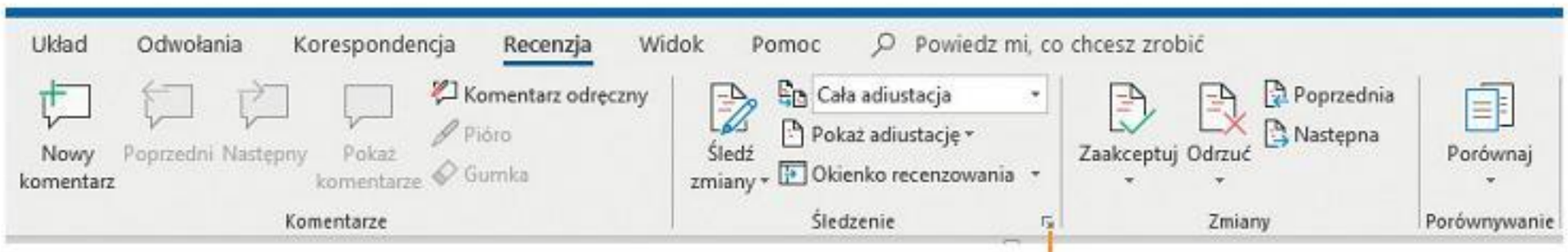
3.1. Śledzenie zmian



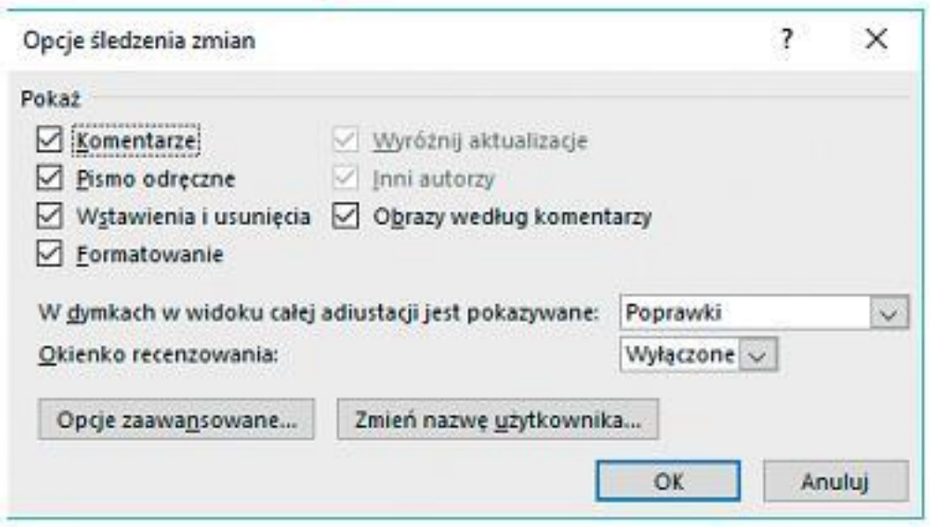
Jeśli włączymy śledzenie zmian, korzystając z opcji **Śledź zmiany** na karcie **Recenzja** lub w menu **Narzędzia**, wprowadzone zmiany będą widoczne w dokumencie.

W starszych wersjach programu Microsoft Word można włączyć pasek narzędzi **Recenzja**, na którym są umieszczone opcje dotyczące recenzji.

Opcje śledzenia zmian można zmienić w **Opcjach śledzenia zmian** (rys. 2b), klikając przycisk na karcie **Recenzja** w grupie **Śledzenie** (rys. 2a).



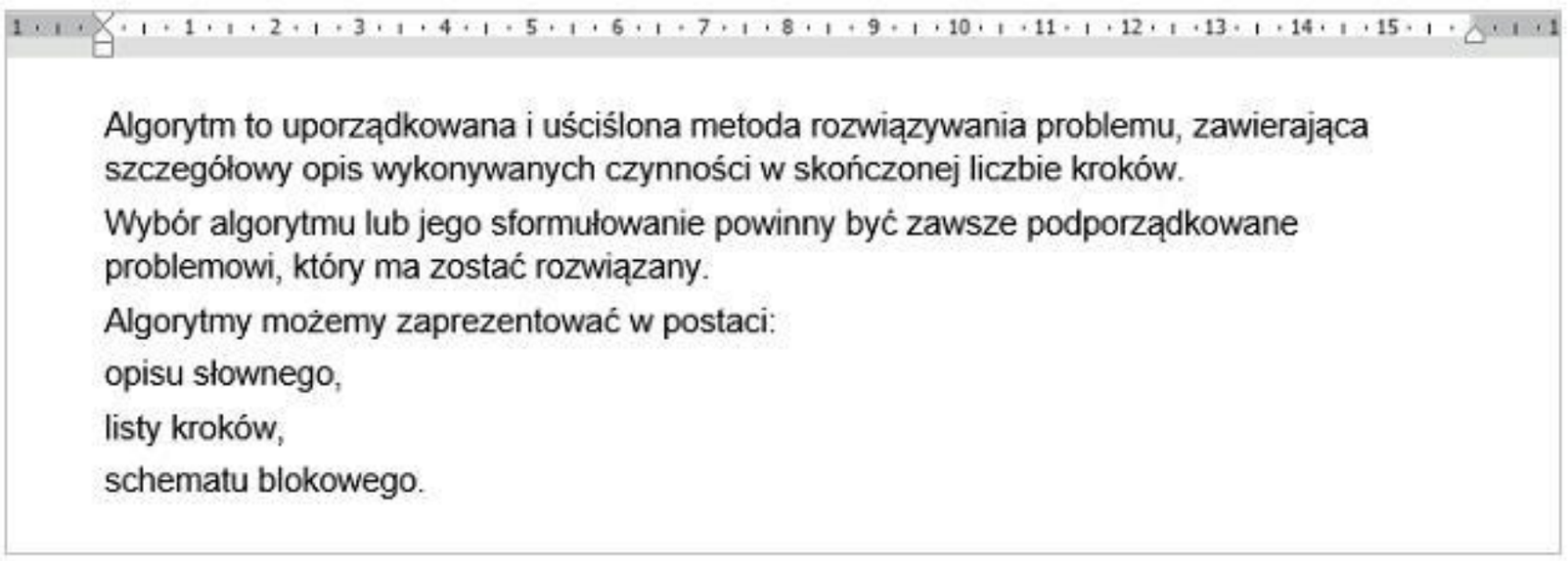
Rys. 2a. *Odkryta karta* ***Recenzja*** *z opcjami recenzji*



Rys. 2b. *Okno* ***Opcje śledzenia zmian***

Na rysunku 3b pokazano poprawiony tekst z rysunku 3a. Po prawej stronie dokumentu na rysunku 3b widzimy informacje o zmianach w dokumencie. Aby wyświetlić wszystkie zmiany przy użyciu różnych kolorów tekstu i linii, musimy wybrać opcję **Cała adiustacja** w grupie **Śledzenie**. Opcja **Prosta adiustacja** wskazuje, gdzie wprowadzono zmiany za pomocą pionowej linii na marginesie. Opcja **Bez adiustacji** ukrywa adiustację, pokazując wygląd wprowadzonych zmian.

Wprowadzone zmiany możemy również zobaczyć w okienku recenzowania w dolnej części okna dokumentu (rys. 3b).


Algorytm to uporządkowana i uściślona metoda rozwiązywania problemu, zawierająca szczegółowy opis wykonywanych czynności w skończonej liczbie kroków.

Wybór algorytmu lub jego sformułowanie powinny być zawsze podporządkowane problemowi, który ma zostać rozwiązany.

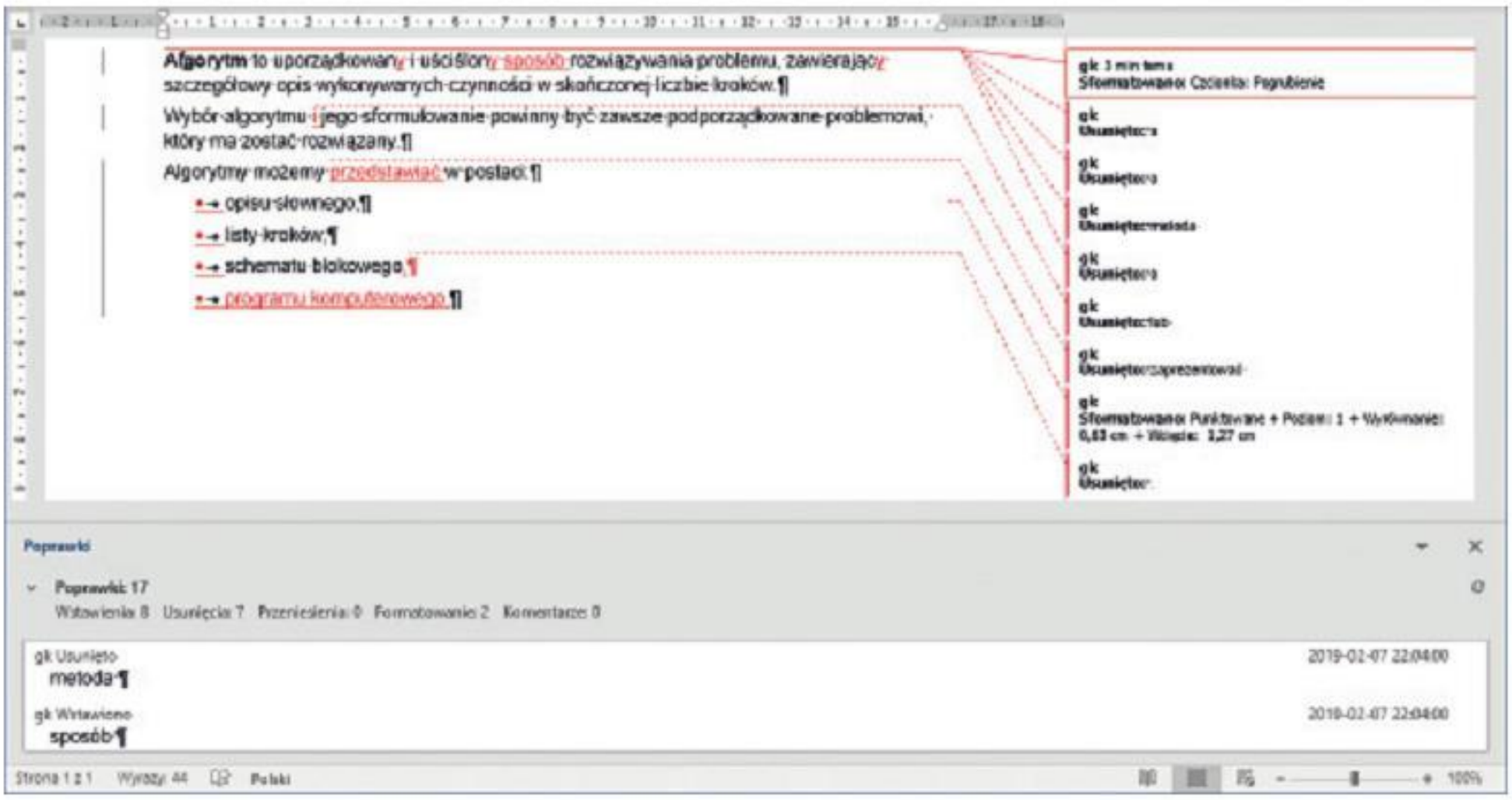
Algorytmy możemy zaprezentować w postaci:

opisu słownego,

listy kroków,

schematu blokowego.

Rys. 3a. *Tekst źródłowy do ćwiczenia 6.*

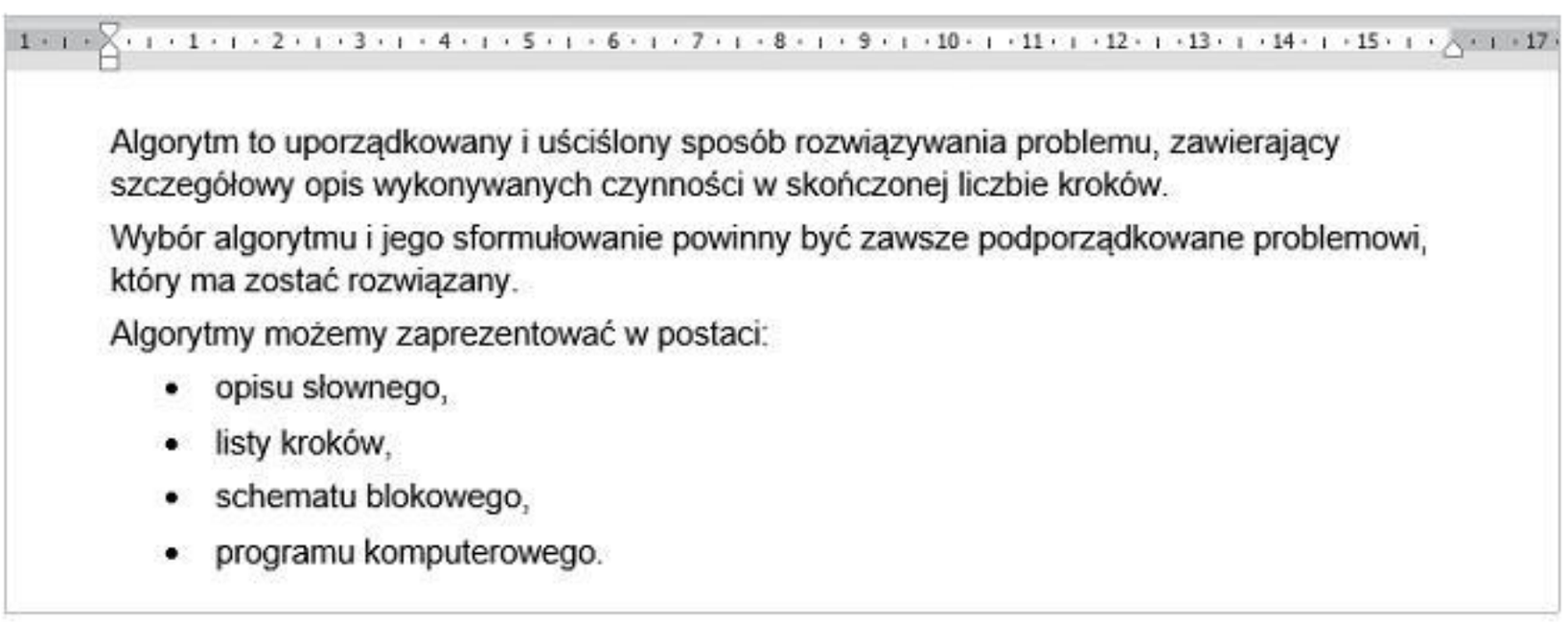


Rys. 3b. *Tekst z poprawkami (w dokumencie włączono śledzenie zmian) – ćwiczenie 6.*

Ćwiczenie 6. Nanosimy poprawki w trybie recenzji

1. Utwórz nowy dokument tekstowy i przepisz tekst pokazany na rysunku 3a. Zapisz dokument w pliku pod nazwą *Algorytm*.
2. Włącz śledzenie zmian i wykonaj następujące poprawki:
   a. pogrub słowo „algorytm",
   b. w pierwszym akapicie tekst „uporządkowana i uściślona metoda" zamień na „uporządkowany i uściślony sposób" oraz wyraz „zawierająca" – na „zawierający",
   c. w drugim akapicie zamień: spójnik „lub" na „i",
   d. w trzecim akapicie zamień wyraz „zaprezentować" – na „przedstawić",
   e. w szóstym akapicie zamień kropkę po słowie „blokowego" na przecinek,
   f. utwórz siódmy akapit i dodaj podpunkt „programu komputerowego" zakończony kropką,
   g. dodaj wypunktowanie sposobów prezentacji algorytmów.
3. Zapisz plik pod nazwą *Korekta 1*.

Wskazówka: Aby zamienić tekst „uporządkowana i uściślona metoda" na tekst „uporządkowany i uściślony sposób", wystarczy w wyrazie „uporządkowana" usunąć ostatnią literę „a" i wstawić w to miejsce literę „y", w wyrazie „uściślona" usunąć ostatnią literę „a" i wstawić w to miejsce literę „y" oraz usunąć wyraz „metoda" i w to miejsce wstawić wyraz „sposób".



Rys. 3c. *Tekst z rysunku 3b po zaakceptowaniu wybranych zmian – ćwiczenie 7.*

Przeglądając zmiany i komentarze, można zaakceptować albo odrzucić wszystkie lub wybrane poprawki (opcje **Zaakceptuj** lub **Odrzuć** w grupie **Zmiany**).

Ćwiczenie 7. Akceptujemy i odrzucamy wprowadzone zmiany

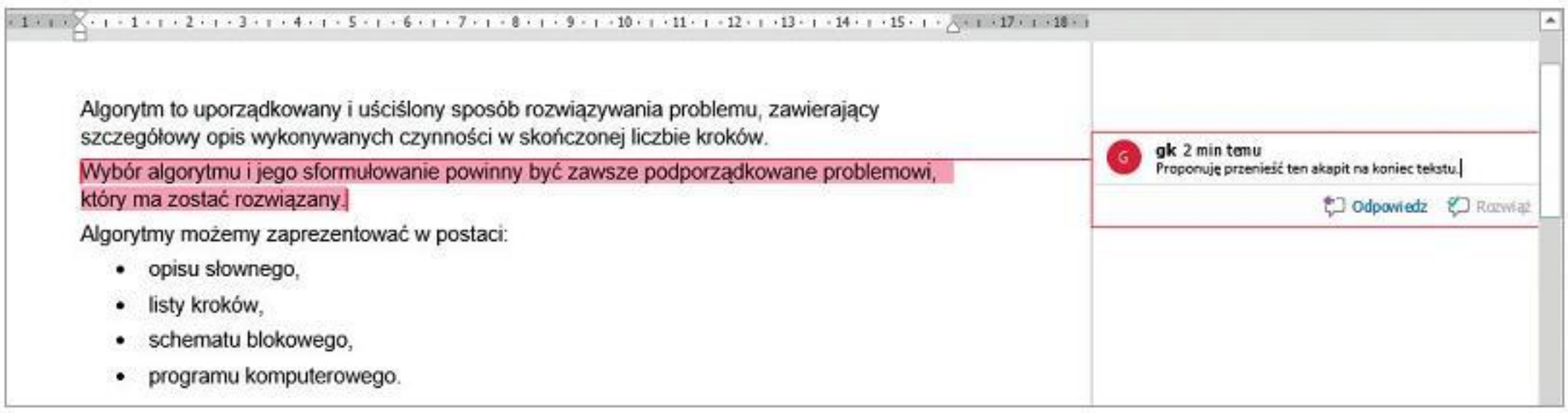
1. Otwórz plik *Korekta 1* zapisany w ćwiczeniu 6. Zaakceptuj następujące zmiany:
   a. zmianę tekstu „uporządkowana i uściślona metoda" na tekst „uporządkowany i uściślony sposób",
   b. zmianę wyrazu „zawierająca" na „zawierający",
   c. zamianę kropki po słowie „blokowego" na przecinek,
   d. dodany podpunkt „programu komputerowego.",
   e. dodane wypunktowanie ze sposobami prezentacji algorytmów.
2. Odrzuć dodanie pogrubienia na słowo „algorytm" oraz zmianę wyrazu „zaprezentować" na „przedstawić". Tekst po zaakceptowaniu i odrzuceniu wybranych zmian powinien wyglądać tak jak pokazano na rysunku 3c.
3. Zapisz plik pod nazwą *Korekta 2*.

3.2. Wstawianie komentarzy

Aby dodać własny komentarz do wybranego fragmentu tekstu, należy go zaznaczyć, a następnie wybrać opcję **Nowy komentarz** na karcie **Recenzja** w grupie **Komentarze,** opcję **Komentarz** w menu **Wstaw** lub przycisk **Wstawianie nowego komentarza** na pasku narzędzi **Recenzja** (zależnie od wersji programu).

Jeśli chcemy dodać komentarz do jednego wyrazu, wystarczy umieścić kursor tekstowy w dowolnym miejscu wyrazu.

Na rysunku 3d po prawej stronie dokumentu widzimy przykładowy komentarz do drugiego akapitu tekstu. Fragment tekstu, do którego dodano komentarz, został wyróżniony zacieniowaniem.



Rys. 3d. *Tekst z rysunku 3c z wstawionym komentarzem – ćwiczenie 8.*

Ćwiczenie 8. Dodajemy komentarze do dokumentu

1. Otwórz plik *Korekta 2* zapisany w ćwiczeniu 7.
2. Wstaw komentarz „Proponuję przenieść ten akapit na koniec tekstu" do akapitu zaczynającego się od słów „Wybór algorytmu".
3. Zapisz plik pod nazwą *Korekta 3*.

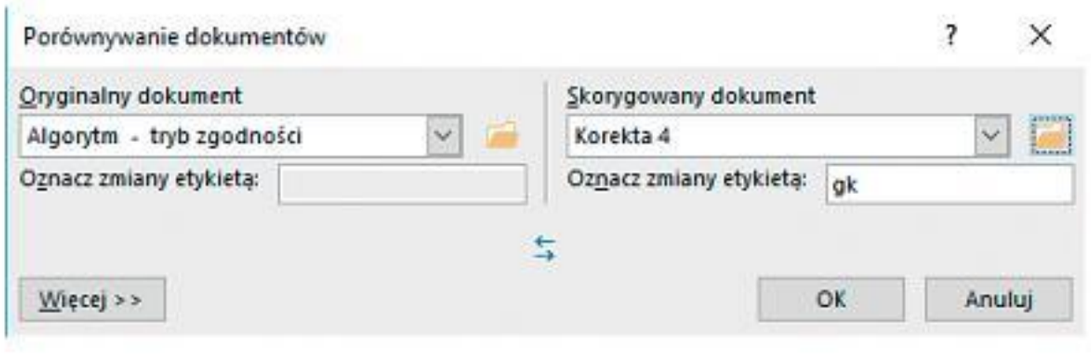
Ćwiczenie 9. Akceptujemy komentarz

1. Otwórz plik *Korekta 3* zapisany w ćwiczeniu 8.
2. Wykonaj polecenie z wstawionego komentarza i usuń komentarz.
3. Zapisz plik pod nazwą *Korekta 4*.

3.3. Porównywanie dokumentów

W edytorze tekstu Microsoft Word możemy porównać dwie wersje dokumentu, np. dokument oryginalny i skorygowany. Efekty porównania mogą być pokazane w pliku oryginalnym, zmienionym lub w nowym dokumencie (ustawienie domyślne).

Aby porównać dwa dokumenty, należy wybrać opcję **Porównaj** na karcie **Recenzja**, a następnie w otwartym oknie **Porównywanie dokumentów** (rys. 4.) wybrać pliki do porównania.



Rys. 4. *Okno* ***Porównywanie dokumentów***

Ćwiczenie 10. Porównujemy dokumenty

Porównaj dokumenty zapisane w plikach *Algorytm* i *Korekta 4*. Wyniki porównania zobacz w nowym dokumencie.

4. Praktyczny poradnik

Porada 1. Kopiowanie formatu

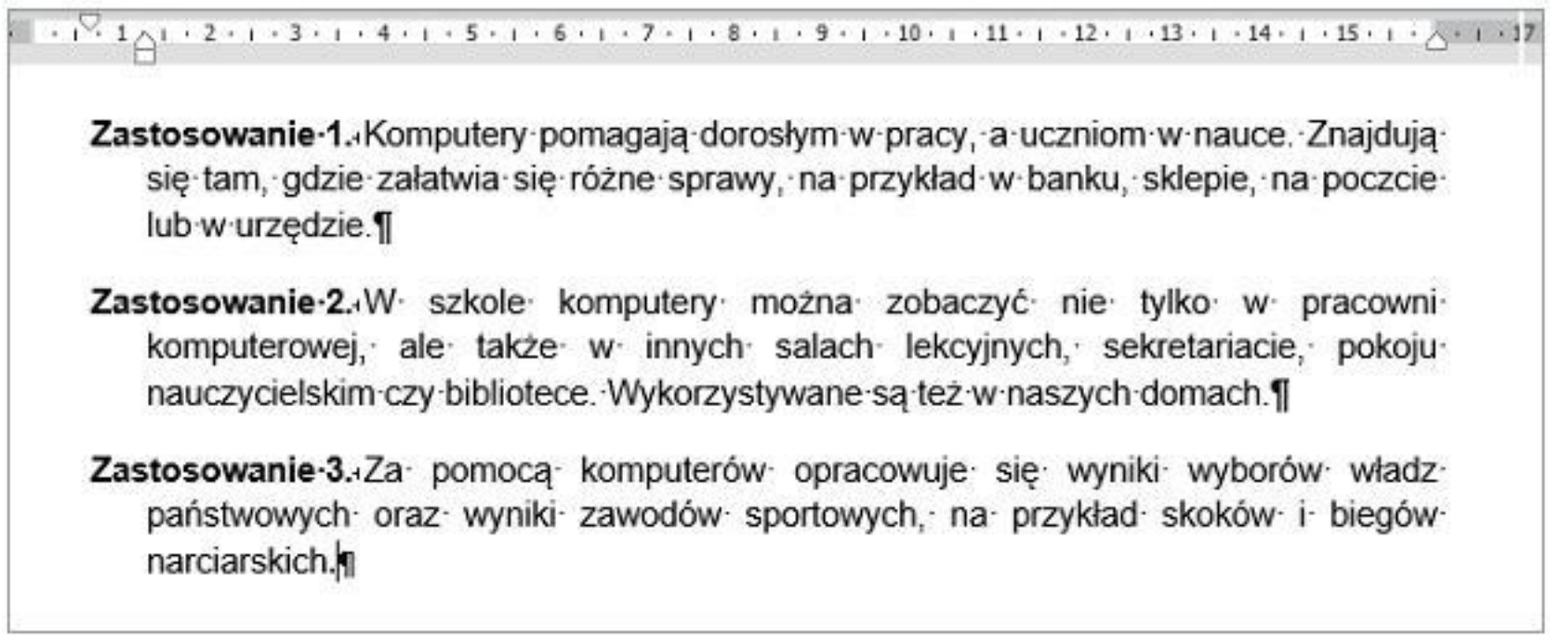
Jeśli chcemy, aby inne fragmenty tekstu były sformatowane tak samo jak wybrany fragment tekstu, można skopiować format fragmentu tekstu, korzystając z przycisku **Malarz formatów** (**Kopiowanie formatu**) .

W tym celu należy zaznaczyć fragment tekstu, którego format chcemy skopiować, kliknąć przycisk (kursor przybierze kształt pędzla) i przeciągnąć pędzlem po fragmencie tekstu, w którym chcemy zmienić formatowanie.

Jeśli chcemy skopiować formatowanie do kilku fragmentów tekstu, należy dwukrotnie kliknąć przycisk **Malarz formatów** (**Kopiowanie formatu**). Aby zakończyć kopiowanie, należy ponownie kliknąć przycisk **Malarz formatów** (**Kopiowanie formatu**).

Porada 2. Dostosowywanie numerowania

Jeśli nie odpowiada nam żaden z proponowanych przez program sposobów numerowania, możemy zdefiniować nowy format numeracji poprzez dobranie własnych parametrów czcionki, stylu numeracji, sposobów wyrównania itp. Możemy również dostosować do własnych potrzeb sposób wypunktowania, zmieniając rodzaj punktora. Możliwości te znajdziemy w opcjach numerowania i wypunktowania.

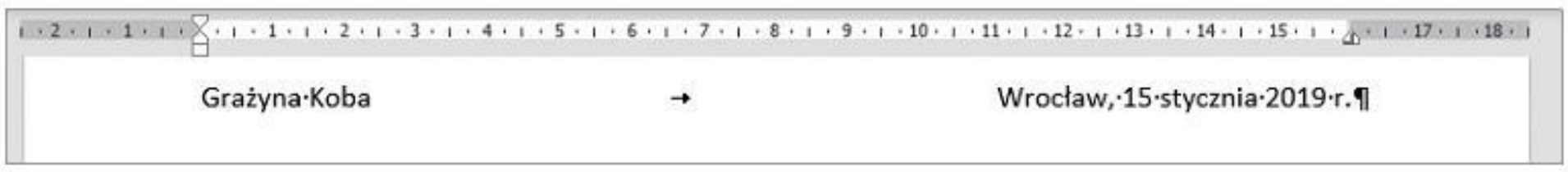


Rys. 5. *Przykład dostosowania rodzaju numeracji*

Porada 3. Zastosowanie tabulatora prawego

Jeśli w górnej części dokumentu (np. pisma urzędowego lub podania) chcemy zapisać w tym samym wierszu imię i nazwisko (wyrównane do lewego marginesu) oraz miejscowość i datę (wyrównane do prawego marginesu), należy:

- ustawić wyrównywanie tekstu do lewego marginesu,
- na linijce ustawić tabulator prawy na prawym marginesie,
- wpisać imię i nazwisko, nacisnąć klawisz **Tab** i wpisać nazwę miejscowości oraz datę.



Rys. 6. *Przykład zastosowania tabulatora prawego*

Porada 4. Kiedy nie należy używać tabulatora?

Nie zaleca się wykonywania wcięcia pierwszego wiersza akapitu przy użyciu tabulatora. Znak tabulacji to dodatkowy znak zapisany w tekście. Połączenie dwóch akapitów może spowodować, że znak ten znajdzie się w środku wiersza i spowoduje niepotrzebne rozsunięcie tekstu. Taki tabulator należy zastąpić ustawieniem wcięcia pierwszego wiersza akapitu.

Porada 5. Wyszukiwanie i zamiana tekstu, słów i znaków

Edytor tekstu umożliwia szybkie wyszukanie fragmentu tekstu, określonych słów czy pojedynczych znaków. Istnieje również możliwość wyszukania i zamienienia znaków niewidocznych na wydruku, np. znaków końca akapitu, spacji, spacji nierozdzielającej, ręcznego podziału wiersza. Są to tzw. **znaki specjalne**. Opcje znajdowania i zamiany tekstu i znaków znajdziemy na karcie **Narzędzia główne** w grupie **Edytowanie** lub w menu **Edycja** (zależnie od wersji programu).

Porada 6. Konwersja tekstu na tabelę

Tekst zapisany w akapitach można przekonwertować na tabelę. Kolejne akapity tekstu zostaną umieszczone w wierszach tabeli.

Przykład 1. Konwertowanie tekstu na tabelę

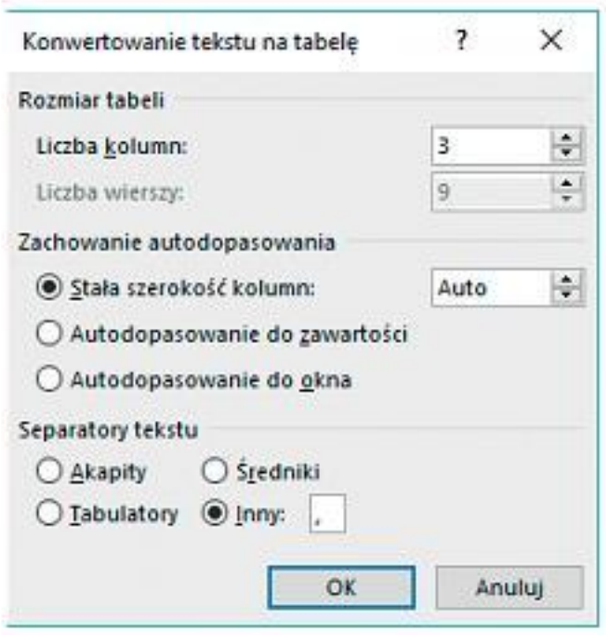
Chcemy zamienić tekst z informacjami o przyjazdach i odjazdach pociągów (rys. 7a) na tabelę składającą się z trzech kolumn i dziewięciu wierszy.

1. Każdy fragment tekstu, który ma być umieszczony w jednym wierszu tabeli, zakończ znakiem końca akapitu.
2. Każdy fragment tekstu, który ma być umieszczony w innej kolumnie tabeli, rozdziel wybranym separatorem, np. znakiem tabulacji, spacją, przecinkiem, średnikiem (na rys. 7a zastosowano przecinek).
3. Zaznacz tekst, który ma być umieszczony w tabeli.
4. Wybierz opcję konwertowania tekstu na tabelę (zwykle w opcjach tworzenia tabeli, np. **Tabele** na karcie **Wstawianie**) – otworzy się okno podobne do pokazanego na rys. 7b. Liczba kolumn i wierszy zostanie wypełniona automatycznie, gdy wybierzesz zastosowany w tekście separator.

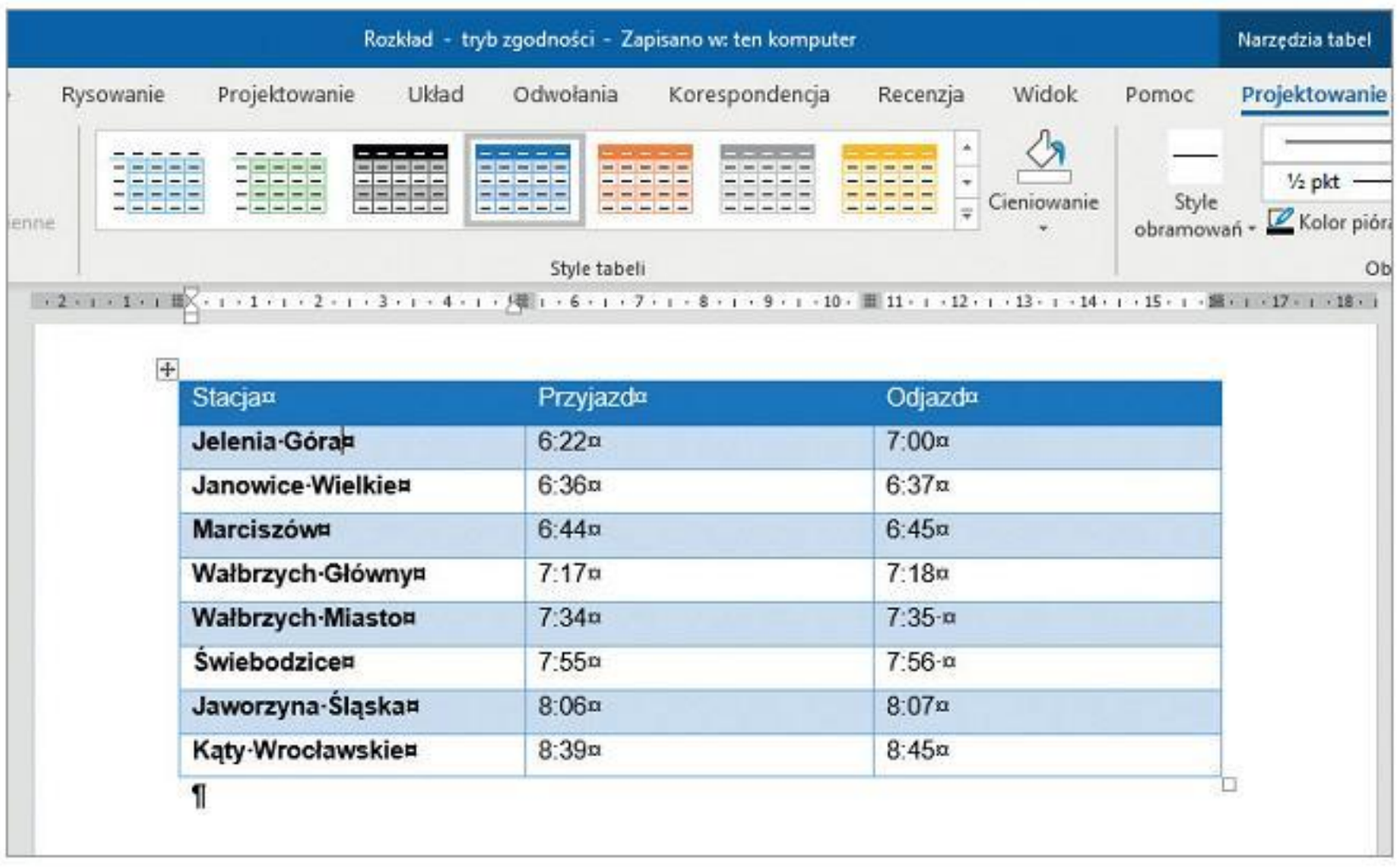
Uwaga: W ten sam sposób możemy konwertować fragment dokumentu tekstowego, który zawiera nie tylko tekst, ale i inne obiekty, np. rysunki.

Stacja,Przyjazd,Odjazd¶
Jelenia·Góra,6:22,7:00¶
Janowice·Wielkie,6:36,6:37¶
Marciszów,6:44,6:45¶
Wałbrzych·Główny,7:17,7:18¶
Wałbrzych·Miasto,7:34,7:35·¶
Świebodzice,·7:55,7:56·¶
Jaworzyna·Śląska,8:06,8:07¶
Kąty·Wrocławskie,8:39,8:45¶

Rys. 7a. *Tekst przygotowany do konwersji na tabelę*



Rys. 7b. *Okno, w którym ustalamy parametry konwersji tekstu na tabelę*



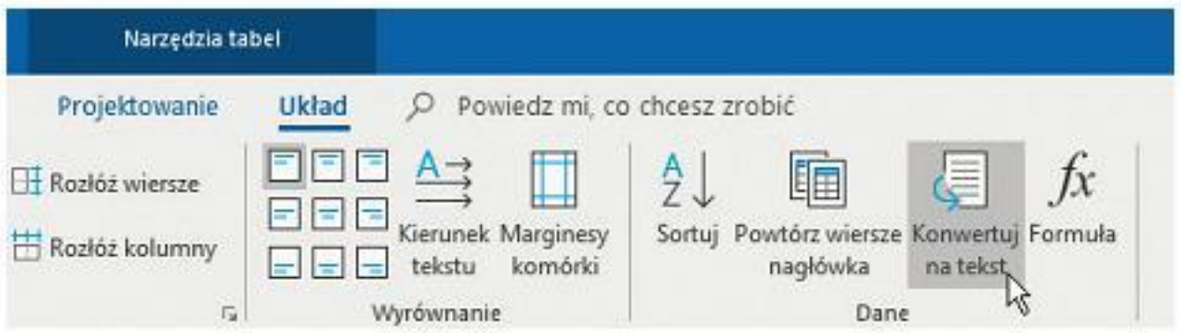
Rys. 7c. *Tekst z rysunku 7a po konwersji na tabelę; dodatkowo tabelę sformatowano*

Ćwiczenie 11. Konwertujemy tekst na tabelę

1. Utwórz nowy dokument tekstowy i przepisz tekst pokazany na rysunku 7a. Zapisz plik pod nazwą *Rozkład*.
2. Wykonaj konwersję tekstu na tabelę. Sformatuj tabelę, korzystając z wybranego stylu w **Narzędziach tabel** na karcie **Projektowanie** (rys. 7c).
3. Zapisz plik pod nazwą *Rozkład 1*.

Porada 7. Konwersja tabeli na tekst

Tabelę można przekonwertować na tekst. W tym celu należy zaznaczyć całą tabelę lub wybrane wiersze, a następnie na karcie **Układ** w **Narzędziach tabel** wybrać opcję **Konwertuj na tekst** (rys. 8.). Następnie w oknie **Konwertowanie tabeli na tekst** należy wybrać znak separatora, który rozdzieli informacje umieszczone w kolumnach tabeli. Wiersze tabeli zostaną rozdzielone znakami końca akapitu.



Rys. 8. *Wybieranie opcji konwertowania tabeli na tekst*

Ćwiczenie 12. Konwertujemy tabelę na tekst

1. Otwórz plik *Rozkład 1* zapisany w ćwiczeniu 11. Wykonaj konwersję tabeli na tekst. W miejsce granic kolumn tabeli jako separatory tekstu wstaw znaki tabulacji.
2. Zapisz plik pod nazwą *Rozkład 2*.

Wskazówka: W przypadku problemów skorzystaj z **Pomocy**.

Porada 8. Pisanie prostych wzorów bez korzystania z edytora równań

Jeśli we wzorze matematycznym, fizycznym czy chemicznym występują tylko indeksy górne i dolne, zapisując go, nie musimy korzystać z edytora równań. Wzory można zapisać, korzystając z funkcji formatowania znaków **Indeks górny** lub **Indeks dolny**. Na przykład: $a^2 + b^2 = c^2$ lub $Fe_2(SO_4)_3$.

Jeśli we wzorze potrzebny jest tylko wybrany symbol, np. Ω, Δ, π, można skorzystać z opcji wstawiania symboli **Symbol** na karcie **Wstawianie** w grupie **Symbole** lub w menu **Wstaw** (zależnie od wersji programu).

Ćwiczenie 13. Piszemy proste wzory w edytorze tekstu

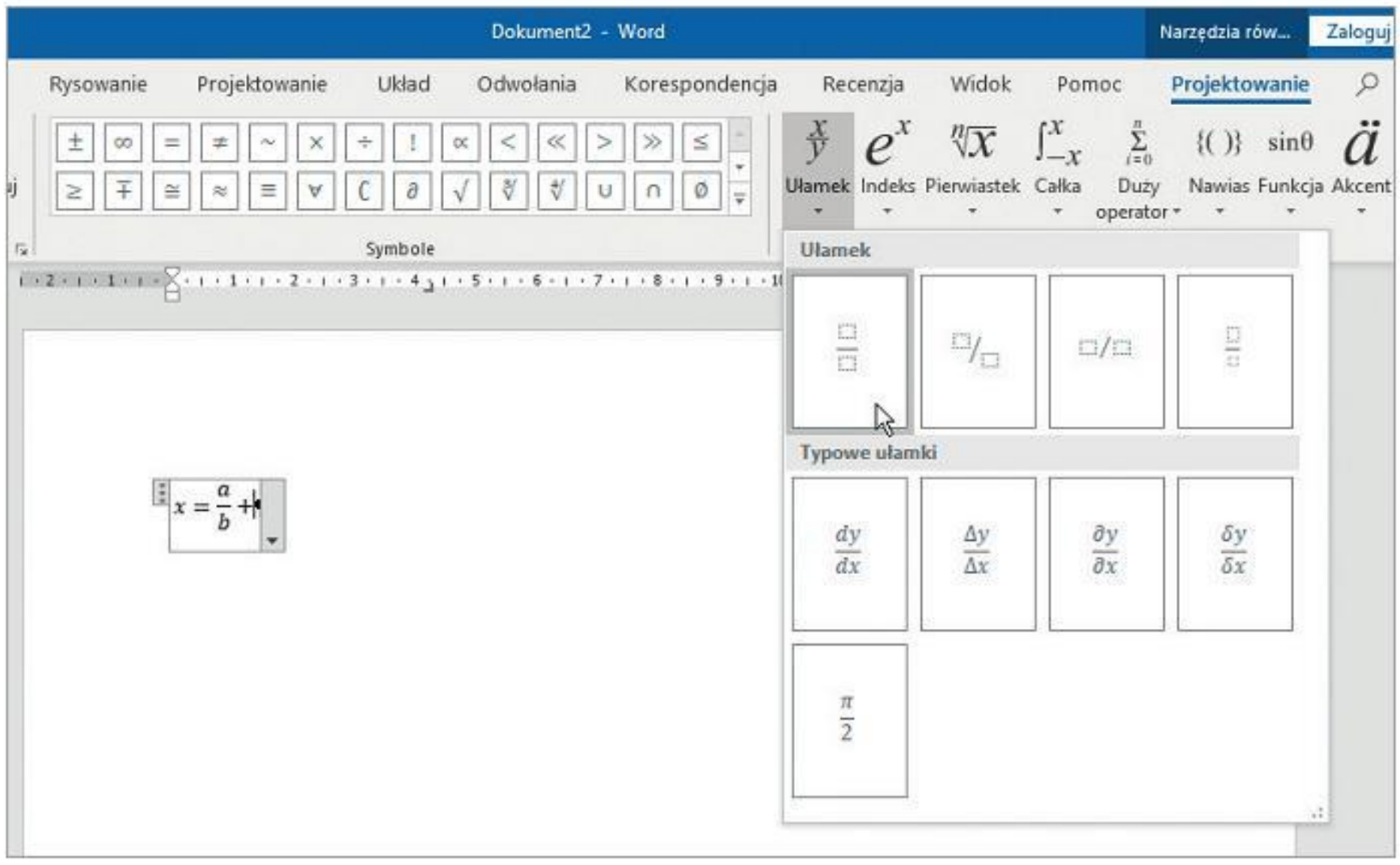
Zapisz w dokumencie tekstowym:
Pole koła: $P = \pi r^2$; *Obwód koła:* $O = 2\pi r$.
Sinus sumy kątów o miarach α *i* β: $\sin(\alpha + \beta) = \sin\alpha \cdot \cos\beta + \cos\alpha \cdot \sin\beta$.
Prędkość chwilowa w ruchu prostoliniowym: $V = \Delta s / \Delta t$; *gdzie* $\Delta s = s - s_1$; $\Delta t = t - t_1$.
Jednostki oporu elektrycznego: $1\ \Omega = 1\ V / 1\ A$; $1\ M\Omega = 10^6\ \Omega$; $1\ k\Omega = 10^3\ \Omega$; $1\ m\Omega = 10^{-3}\ \Omega$.

Porada 9. Korzystanie z edytora równań

Aby zapisać złożone wzory matematyczne lub fizyczne (zawierające np. znak pierwiastka, ułamki zwykłe i piętrowe), możemy skorzystać z programu **edytor równań** dołączonego do edytora tekstu.

Zależnie od wersji edytora tekstu edytor równań umożliwia tworzenie mniej lub bardziej skomplikowanych wzorów. Edytor równań uruchomiamy, korzystając z opcji **Równanie** na karcie **Wstawianie** lub z opcji **Obiekt** w menu **Wstaw**.

Wzór edytujemy w polu, które pojawia się po uruchomieniu edytora równań, korzystając z wzorców proponowanych przez program (rys. 9.). Po zamknięciu okna edytora tekstu wzór zostaje zapisany w dokumencie i jest traktowany jak rysunek. Podwójne kliknięcie utworzonego wzoru powoduje powrót do jego edycji.



Rys. 9. *Przykład edytora równań (Microsoft Word)*

Porada 10. Pisanie w innych językach

Jeśli w edytorze Microsoft Word chcemy napisać tekst w języku obcym, najprościej skorzystać z opcji wstawiania symboli. Możemy w ten sposób zapisać znaki charakterystyczne dla danego języka, np. niemieckie (ä, ü, ö, ß), czeskie (á, ě, ř, ž) lub rosyjskie (б, д, ё, ж, й, щ). Do często używanych znaków przypisane są skróty klawiaturowe.

Ćwiczenie 14. Piszemy tekst w języku niemieckim

1. Przepisz do dokumentu tekstowego poniższy fragment słowniczka:
   dziewczyna – *das Mädchen*, bułka – *das Brötchen*, burak – *die Rübe,*
   chrabąszcz – *der Maikäfer,* klucz – *der Schlüssel,* ulica – *die Straße*.
2. Zapisz plik pod nazwą *Słowniczek 1*.

System operacyjny Windows umożliwia zmianę układu klawiatury w zależności od wybranego języka. Więcej informacji na ten temat można odszukać w **Pomocy** systemu Windows, wpisując odpowiednie hasło, np. „języki", „klawiatura", „zmienianie układu klawiatury", „pisanie w innych językach". Zmian języka/układu klawiatury dokonujemy w oknie opcji regionalnych i językowych (np. **Region i język** lub **Opcje regionalne i językowe**; zakładka **Klawiatura i języki** lub **Języki**).

Warto zapamiętać

- Jeśli tworzymy dokumenty tekstowe o rozbudowanej strukturze, warto stosować (w razie potrzeby) podział dokumentu na strony, sekcje i kolumny.
- Aby zastosować we fragmentach dokumentu różne parametry formatowania strony, np. inną orientację, należy wstawić w odpowiednich miejscach znaki podziału sekcji.
- Aby tworzyć dokumenty o typowej strukturze, możemy korzystać z szablonów. Program Microsoft Word zawiera m.in. szablony życiorysu, ulotki, listu motywacyjnego, menu (jadłospisu).
- Edytor tekstu wyposażony jest w funkcje ułatwiające pracę redakcyjną nad tekstem. Aby śledzić zmiany (np. operacje redagowania, formatowania) wprowadzane w dokumencie tekstowym, można włączyć opcję śledzenia zmian na karcie **Recenzja** lub w menu **Narzędzia** (zależnie od wersji programu). Praca w trybie recenzji umożliwia również wstawianie komentarzy i porównanie dwóch dokumentów.
- W edytorze tekstu możliwa jest konwersja tekstu na tabelę i odwrotnie – tabeli na tekst. Przed konwersją tekstu na tabelę trzeba tekst odpowiednio sformatować, m.in. każdy wiersz tekstu, który ma być umieszczony w jednym wierszu tabeli, należy zakończyć znakiem końca akapitu, a każdy fragment tekstu, który ma być umieszczony w innej kolumnie tabeli – należy rozdzielić wybranym separatorem, np. znakiem tabulacji, spacją, przecinkiem, średnikiem.
- Edytory tekstu umożliwiają redagowanie wzorów matematycznych, fizycznych, chemicznych i innych. W tym celu możemy skorzystać z edytora równań, opcji formatowania czcionki (indeksy górne i dolne) oraz opcji wstawiania symboli. Korzystając z symboli, można również wprowadzać znaki charakterystyczne dla innych języków, np. niemieckiego, rosyjskiego, greckiego.

Pytania i polecenia

1. W jakim celu stosuje się ręczny podział strony?
2. Czym jest sekcja? W jakim celu stosuje się podział sekcji?
3. Czym jest szablon dokumentu? Podaj kilka przykładów zastosowania szablonów.
4. W jaki sposób można śledzić zmiany wprowadzone w dokumencie tekstowym?
5. W jaki sposób można wstawić własne komentarze do dokumentu tekstowego?
6. Jak można porównać dwa dokumenty tekstowe? Wyjaśnij na przykładzie.
7. W jaki sposób wykonać konwersję tekstu na tabelę?
8. W jaki sposób wykonać konwersję tabeli na tekst?
9. W jaki sposób można zapisywać wzory matematyczne, fizyczne, chemiczne i inne?
10. W jaki sposób można napisać w edytorze tekstu tekst w języku niemieckim, stosując znaki charakterystyczne dla tego języka?

Zadania

1. Otwórz plik *Budowa komputera* zapisany w zadaniu 8. z tematu B1. Każde zagadnienie (punkt pierwszego poziomu opracowania) umieść na oddzielnej stronie. Zapisz plik pod tą samą nazwą.
2. Przygotuj jednostronicową ulotkę o swojej szkole, potrzebną na dni otwarte szkoły. Zastosuj podział tekstu na kolumny: w górnej części ulotki tekst powinien być umieszczony w jednej kolumnie, a w dolnej – w dwóch.
3. Korzystając z wybranego szablonu, przygotuj papier firmowy biura turystycznego.
4. Korzystając z wybranego szablonu, przygotuj kalendarz na następny rok.
5. Utwórz nowy dokument tekstowy. Opisz w nim możliwości pracy w trybie recenzji. Zapisz plik pod nazwą *Recenzja 1*. Prześlij plik koledze (koleżance), który (która) wprowadzi do dokumentu poprawki i komentarze w trybie recenzji, zapisze plik pod nazwą *Recenzja 2* i ci go odeśle. Zaakceptuj lub odrzuć wprowadzone poprawki. Porównaj pliki *Recenzja 1* i *Recenzja 2*.
6. Przepisz tekst z rysunku 5., pomijając nazwy punktów: *Zastosowanie 1.*, *Zastosowanie 2.*, … . Zdefiniuj format numerowania taki, jaki pokazano na rysunku 5. Ponumeruj akapity. Zapisz plik pod nazwą *Zastosowania*.
7. Korzystając z edytora równań, zapisz:
   a. definicję potęgi o wykładniku naturalnym,
   b. definicję potęgi o wykładniku zerowym,
   c. wyrażenie:

$$\left[\left(2^{-3}-\frac{3}{8}\right)^{-1}+\left(\frac{1}{2}\right)^{-4}\right]^{-2}=\left[\left(\left(\frac{1}{2}\right)^{3}-\frac{3}{8}\right)^{-1}+2^{4}\right]^{-2}=\left[\left(\frac{1}{8}-\frac{3}{8}\right)^{-1}+16\right]^{-2}=$$
$$=\left[\left(-\frac{1}{4}\right)^{-1}+16\right]^{-2}=(-4+16)^{-2}=\left(\frac{1}{12}\right)^{2}=\frac{1}{144}$$

8. Zapisz w edytorze tekstu litery rosyjskiego alfabetu: А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я.
9. Otwórz plik *Słowniczek 1* zapisany w ćwiczeniu 14. Przekonwertuj tekst na tabelę składającą się z dwóch kolumn. Wcześniej przygotuj odpowiednio tekst, umieszczając każdy wyraz i jego tłumaczenie w oddzielnym akapicie. Jako separatorów użyj średników. Dodaj w pierwszym wierszu odpowiednie nagłówki. Zapisz plik pod nazwą *Słowniczek 2*.

Dla zainteresowanych

10. Korzystając z **Pomocy** programu Microsoft Word, zapoznaj się z dodatkowymi możliwościami recenzji dokumentu.
11. Korzystając z edytora równań, zapisz wzory funkcji:
    a. $f(x)=\begin{cases}\frac{1}{x+3}+3\frac{1}{2}, & \text{jeśli } x\in(-\infty,-3)\\ -\sqrt{x^2}+5, & \text{jeśli } x\in\langle -3,4\rangle\end{cases}$
    b. $f(x)=\frac{\sqrt{|x+1|-5}}{\sqrt{(x-2)^2-3}}$
12. Przygotuj pracę na wybrany temat z dziedziny, która cię interesuje. Zastosuj poznane sposoby opracowywania tekstu, np.: zdefiniuj własne style, wstaw tekst do stopki i nagłówka, umieść przypisy, spis treści i spis ilustracji, zastosuj podziały stron i sekcji oraz podział na kolumny.

Temat B3. Opracowywanie grafiki rastrowej

1. Rodzaje grafiki komputerowej
2. Formaty plików graficznych
   2.1. Formaty plików grafiki rastrowej
   2.2. Formaty plików grafiki wektorowej
3. Tworzenie obrazu w programie GIMP
4. Praca z warstwami w programie GIMP
5. Stosowanie narzędzi selekcji w programie GIMP
6. Edycja zdjęć w programie GIMP
   6.1. Wybrane filtry i efekty
   6.2. Zmiana kontrastu i nasycenia kolorów
   6.3. Kadrowanie i skalowanie

Warto powtórzyć

1. Co to jest format pliku?
2. Wymień nazwy poznanych programów graficznych.
3. Jakie znasz możliwości programów graficznych?
4. Jakie przekształcenia obrazu można wykonać w programie graficznym?

1. Rodzaje grafiki komputerowej

Grafika komputerowa to dział technologii informacyjnej związany z wykorzystywaniem komputerów do generowania i przetwarzania obrazów. Grafika komputerowa jest przydatna w wielu dziedzinach – od przygotowywania publikacji (ang. DTP – ***Desk**top **P**ublishing*), przez projektowanie wspomagane komputerowo (ang. CAD – *Computer **A**ided **D**esign*), po gry komputerowe i efekty specjalne w filmach.

Podział grafiki komputerowej ze względu na sposób tworzenia

Grafika rastrowa, nazywana także **grafiką bitmapową** – jest związana ze sposobem wyświetlania obrazu na ekranie komputera. Obraz składa się z wielu pojedynczych punktów (pikseli). Każdy punkt ma własną, niezależną od innych barwę. Grafikę rastrową wykorzystujemy na przykład przy przetwarzaniu fotografii czy zeskanowanych obrazów. Wadami grafiki rastrowej są: utrata jakości obrazu przy zmianie wymiarów (zwłaszcza przy powiększaniu) oraz duże rozmiary plików.

Przykładowe programy do tworzenia grafiki rastrowej: Paint, Adobe Photoshop, GIMP, Paint.NET.

Grafika wektorowa – obraz jest zdefiniowany matematycznie jako zbiór obiektów geometrycznych: odcinków, okręgów, wielokątów, krzywych. Każdemu obiektowi są przypisane atrybuty (kolor linii, grubość linii, wypełnienie, przezroczystość), ma on również określone wymiary i położenie względem innych obiektów. Zaletami grafiki wektorowej są: możliwość zmieniania rozmiaru obrazów bez utraty jakości i niewielka objętość plików. Grafikę wektorową stosuje się najczęściej do tworzenia rysunków technicznych, planów czy szkiców – nie nadaje się ona natomiast do zapisu fotografii. Przykładowe programy do tworzenia grafiki wektorowej: CorelDRAW, Adobe Illustrator, Inkscape.

Podział grafiki komputerowej ze względu na sposób reprezentowania danych obrazu

Grafika dwuwymiarowa (2D) to płaskie obrazy, reprezentujące dwuwymiarowe obiekty.

Grafika trójwymiarowa (3D) służy do przedstawiania obiektów przestrzennych oraz ich prezentacji w postaci dwuwymiarowych obrazów. Przetwarzanie grafiki trójwymiarowej jest znacznie bardziej skomplikowane niż przetwarzanie grafiki dwuwymiarowej oraz wymaga dużej mocy obliczeniowej komputera. Grafikę trójwymiarową wykorzystuje się między innymi w grach komputerowych i filmach.

Przykładowe programy do tworzenia grafiki trójwymiarowej: Paint 3D, 3D Studio Max, Blender, POV-Ray.

2. Formaty plików graficznych

Pliki graficzne z obrazami rastrowymi mają zwykle duży rozmiar. Wynika to z konieczności zapamiętania koloru każdego z punktów obrazu. Przy założeniu, że każdy z pikseli obrazu jest opisany za pomocą trzech bajtów (może przyjąć jeden z $2^{24} \approx 16$ milionów kolorów), do zapisania obrazu o rozmiarach 1024 x 768 pikseli potrzeba ponad 2 MB pamięci. Stosując jednak odpowiednie algorytmy kompresji, można zmniejszyć objętość takiego obrazu do 100-200 KB, przy minimalnej utracie jakości.

F **Format pliku**

Reguły określające wewnętrzną budowę pliku.

2.1. Formaty plików grafiki rastrowej

Formaty najczęściej wykorzystywane do zapisu grafiki rastrowej to: GIF, PNG, JPEG, TIFF, BMP. Pierwsze trzy z wymienionych formatów stosuje się powszechnie w Internecie, głównie ze względu na niewielkie rozmiary plików. Tabela 1. zawiera porównanie najistotniejszych cech popularnych formatów graficznych.

Format	Rozszerzenie nazwy pliku	Zalety	Wady	Zastosowania
BMP	*bmp*	obsługiwany przez większość popularnych programów graficznych	duże rozmiary plików	prosta grafika, ikony

Format	Rozszerzenie nazwy pliku	Zalety	Wady	Zastosowania
GIF	*gif*	możliwość tworzenia prostych animacji; możliwość ustawiania przezroczystości obrazu	paleta barw ograniczona do 256 kolorów	elementy graficzne na potrzeby stron WWW
JPEG	*jpg, jpe, jpeg*	dobra kompresja zdjęć przy niewielkiej utracie jakości	widoczna utrata jakości w przypadku niektórych typów obrazów	zdjęcia i inne obrazy z płynnymi przejściami tonalnymi
PNG	*png*	lepsza kompresja w porównaniu z formatem GIF	nieobsługiwany przez bardzo stare programy	elementy graficzne na potrzeby stron WWW
TIFF	*tif, tiff*	wierne odwzorowanie barw i możliwość przechowywania dodatkowych informacji powiązanych z obrazem	duże rozmiary plików	grafika na potrzeby DTP; przechowywanie obrazów bez utraty jakości; przenoszenie obrazów między różnymi platformami systemowymi
PSD	*psd*	zapis dodatkowych informacji o obrazie (np. warstwy, ścieżki, maski)	bardzo duże rozmiary plików	format programu Adobe Photoshop
XCF	*xcf*	zapis dodatkowych informacji o obrazie (np. warstwy, ścieżki, maski)	bardzo duże rozmiary plików	format programu GIMP

Tabela 1. *Wybrane formaty plików grafiki rastrowej. Trzyliterowe rozszerzenia nazw plików pochodzą z czasów systemu MS-DOS, który nie pozwalał na używanie dłuższych rozszerzeń*

Ćwiczenie 1. Porównujemy rozmiary plików zapisanych w różnych formatach

Otwórz w edytorze graficznym wybrany plik graficzny (z własnych zbiorów obrazów), a następnie zapisz go w każdym z formatów przedstawionych w tabeli 1., obsługiwanych przez twój program graficzny. Jeżeli używany przez ciebie edytor graficzny pozwala wybrać jakość przy zapisywaniu obrazu w formacie JPEG, wypróbuj kilka ustawień jakości. Następnie porównaj wielkość każdego z uzyskanych plików oraz oceń jakość obrazu. Wyniki umieść w tabeli utworzonej w edytorze tekstu.

Format pliku	Wielkość pliku w bajtach/kilobajtach	Subiektywna ocena jakości

2.2. Formaty plików grafiki wektorowej

Pliki grafiki wektorowej charakteryzują się mniejszą objętością w stosunku do plików grafiki rastrowej, co wiąże się z innym sposobem reprezentacji informacji o elementach obrazu. Uniwersalnym mechanizmem reprezentacji grafiki wektorowej jest język PostScript, wykorzystywany między innymi przez profesjonalne urządzenia drukujące. Pliki z instrukcjami języka PostScript posiadają rozszerzenia *ps* i *eps*. Inne rozpowszechnione formaty grafiki wektorowej to: CDR (format plików programu CorelDRAW; rozszerzenie *cdr*) i SVG (format programu Inkscape, obsługiwany np. przez przeglądarki internetowe; rozszerzenie *svg*).

Warto wiedzieć, że dokumenty w formacie PDF mogą zawierać zarówno tekst, jak i grafikę rastrową oraz wektorową, dzięki czemu można je wykorzystywać w grafice komputerowej.

3. Tworzenie obrazu w programie GIMP

Opracowywanie grafiki rastrowej pokażemy z wykorzystaniem bezpłatnego programu GIMP (z ang. ***G****NU* ***I****mage* ***M****anipulation* ***P****rogram*). W większości zastosowań dorównuje on możliwościom płatnych pakietów, takich jak Adobe Photoshop czy Corel Paint Shop Pro. Jest on często wykorzystywany do obróbki zdjęć czy tworzenia grafiki na potrzeby Internetu.

Po uruchomieniu programu GIMP zobaczymy okno **Edytor obrazów** z głównym menu programu (rys. 1.).



Aby rozpocząć tworzenie obrazu, należy skorzystać z opcji **Plik/Nowy**. W otwartym oknie tworzenia nowego obrazu można ustalić m.in. inny niż domyślny rozmiar obrazu.

Narzędzia rysowania, zaznaczania i przekształcania znajdziemy w menu **Narzędzia**. Wygodniej jednak otworzyć okno **Przybornik** (rys. 2a) w opcji **Okna/Nowy przybornik** lub **Okna/Przybornik**.



W programie GIMP możemy otworzyć również dodatkowe okna z innymi narzędziami. Możemy pracować w trybie wielu okien lub w trybie jednego okna, który włączamy, wybierając opcję **Tryb jednego okna** w menu **Okna**.

Przybornik zawiera przyciski narzędzi i wyboru kolorów. Większość narzędzi znamy z programu Paint. Niektóre z nich mają takie same nazwy (np. **Ołówek**, **Pędzel**, **Gumka**, **Zaznaczenie prostokątne**, **Tekst**), inne – zbliżone (np. **Wypełnienie kubełkiem**, **Obrót**, **Odbicie**). Jeśli włączymy tryb jednego okna, to **Przybornik** będzie umieszczony w oknie programu (podobnie jak w programie Paint).

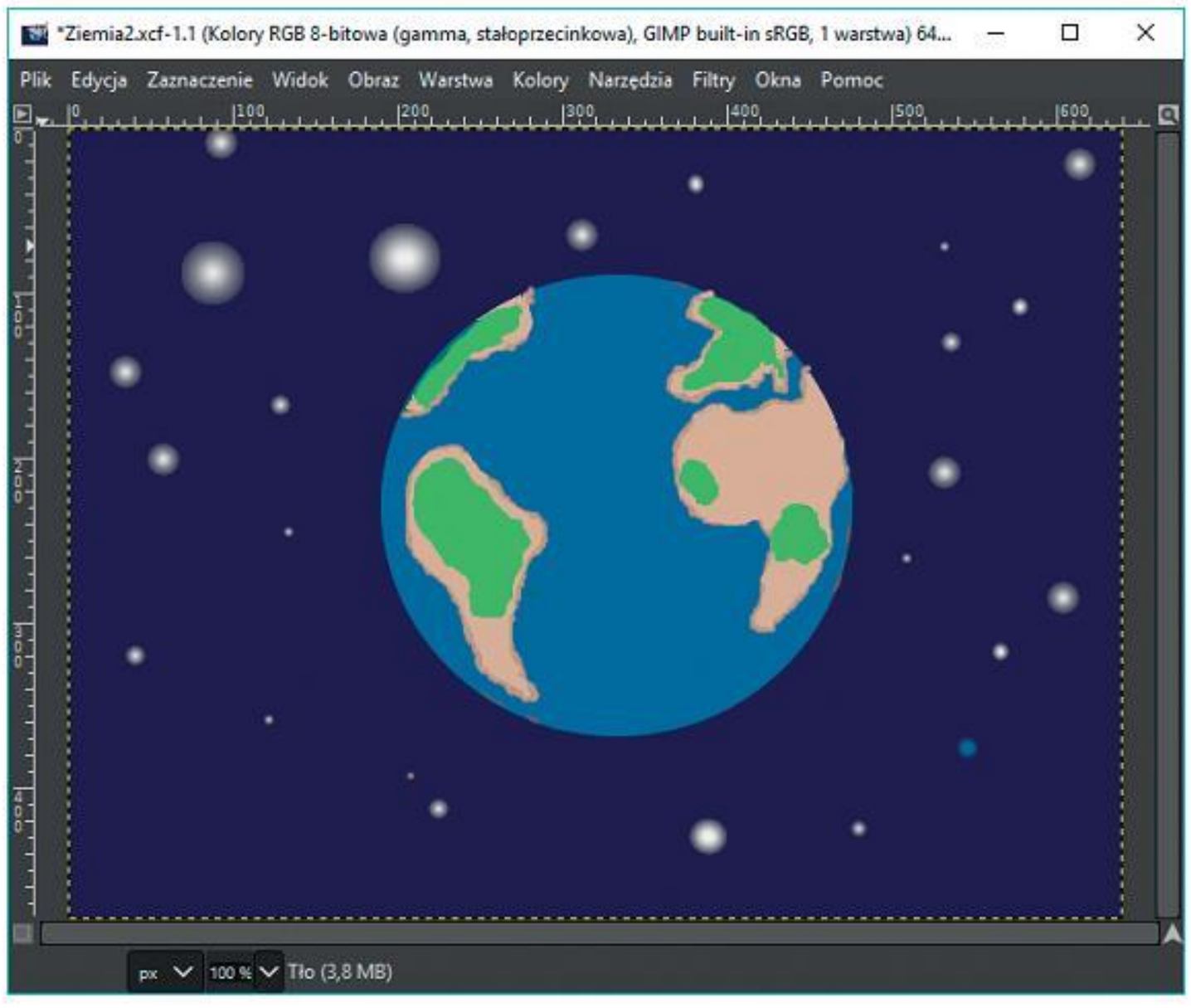
W programie GIMP do rysowania prostych figur można użyć narzędzi zaznaczania i za ich pomocą wyodrębnić prostokąt lub elipsę. Do rysowania linii można wykorzystać **Ołówek**.

Aby narysować linię prostą lub łamaną, można wybrać narzędzie **Ołówek** i, przytrzymując klawisz **Shift**, kliknąć w miejscu, gdzie ma się zaczynać prosta lub jej odcinek, a następnie kliknąć w miejscu, do którego ma sięgać prosta lub jej odcinek.

Aby narysować prostokąt lub elipsę, można:

- wybrać odpowiednie narzędzie zaznaczania i zaznaczyć kształt figury, np. prostokąta (kwadratu), elipsy (koła),
- pokolorować kontur figury lub wypełnić ją kolorem, wybierając np. z menu kontekstowego obrazu opcje **Edycja/Rysuj wzdłuż zaznaczenia** lub **Edycja/Wypełnij kolorem pierwszoplanowym**),
- usunąć zaznaczenie, klikając poza jego obszarem.

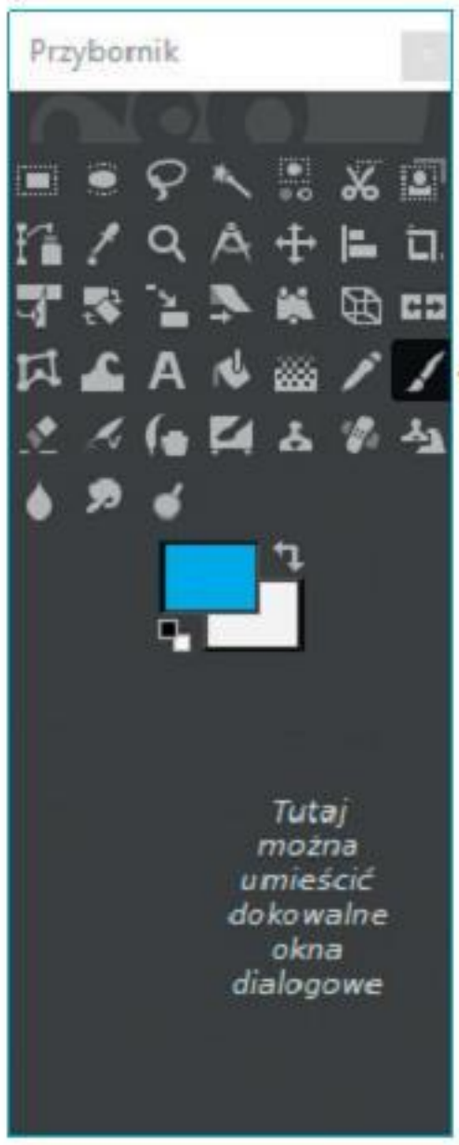
Wnętrze figury lub jej kontur możemy również wypełnić, stosując narzędzie **Wypełnienie kubełkiem** i klikając w dowolnym punkcie wnętrza figury lub jej konturu.



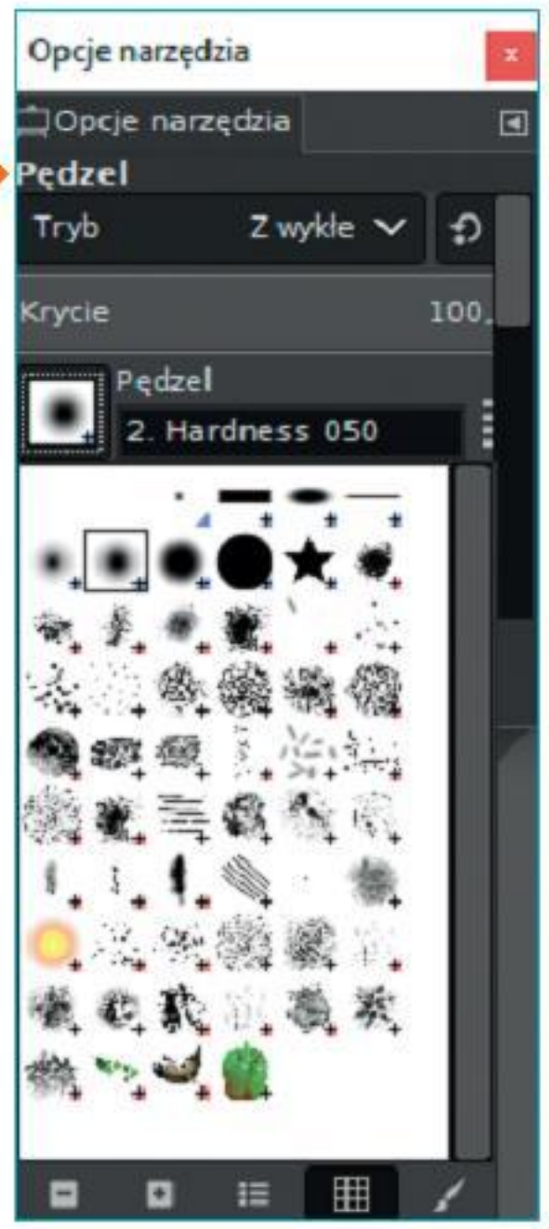
Rys. 1. *Okno* ***Edytor obrazów*** *programu GIMP (wersja 2.10.8) – praca w trybie wielu okien*

Każde narzędzie posiada dodatkowe opcje. Na rysunku 2b pokazano opcje narzędzia **Pędzel**. W oknie **Opcje narzędzia** można zmieniać rodzaj i rozmiar narzędzia oraz inne parametry.

Aby wyświetlić opcje wybranego narzędzia, należy dwukrotnie kliknąć przycisk narzędzia w **Przyborniku** (rys. 2a) – otworzy się okno **Opcje narzędzia** (rys. 2b). Jeśli włączymy opcję **Tryb jednego okna**, to każda z wybranych opcji otworzy się w oknie programu.



Rys. 2a. *Okno* ***Przybornik***



Rys. 2b. *Opcje narzędzia* ***Pędzel***



Ćwiczenie 2. Tworzymy obraz w programie GIMP

1. W programie GIMP utwórz nowy obraz o rozmiarach 640 x 480 pikseli.
2. Narysuj obraz podobny do pokazanego na rysunku 1., przedstawiający Ziemię widzianą z kosmosu. Użyj m.in. narzędzi: **Pędzel**, **Zaznaczenie eliptyczne**, **Wypełnienie kubełkiem**. Postaraj się uzyskać efekt rozświetlonych gwiazd różnej wielkości.
3. Zapisz plik pod nazwą *Ziemia*.

Wskazówka: Na rysunku 1. efekt rozświetlonych gwiazd uzyskano, klikając w obszarze rysunku **Pędzlem**  (rys. 2b). Aby uzyskać różną wielkość gwiazd, wystarczy zmieniać rozmiar **Pędzla**.



Aby w programie GIMP zapisać obraz w pliku, stosujemy opcje **Plik/Zapisz jako** lub **Plik/Zapisz** – plik zostanie zapisany z domyślnym rozszerzeniem *xcf*.

XCF to wewnętrzny format używany przez program GIMP. Wraz z obrazem zapisywane są dodatkowe informacje, między innymi o układzie warstw i zaznaczeniach. Dzięki temu można z nich ponownie korzystać po otwarciu zapisanego pliku.

Jeżeli chcemy zapisać jedynie ostateczną postać obrazu (bez dodatkowych informacji o warstwach i zaznaczeniach), należy wybrać opcję **Plik/Wyeksportuj jako** i użyć jednego z popularnych formatów graficznych, o których mowa w punkcie 2.1., np. PNG, JPEG.

4. Praca z warstwami w programie GIMP

Program GIMP, podobnie jak większość nowoczesnych programów graficznych, umożliwia pracę z **warstwami** obrazu. Warstwy można porównać do nałożonych na siebie przezroczystych folii. Każda z warstw zawiera fragmenty obrazu, które po nałożeniu na siebie składają się w jedną całość.

Wyróżnioną warstwą jest tak zwane **Tło**. W ćwiczeniu 2. rysunek Ziemi widzianej z kosmosu tworzyliśmy na pierwszej warstwie, czyli na **Tle**.

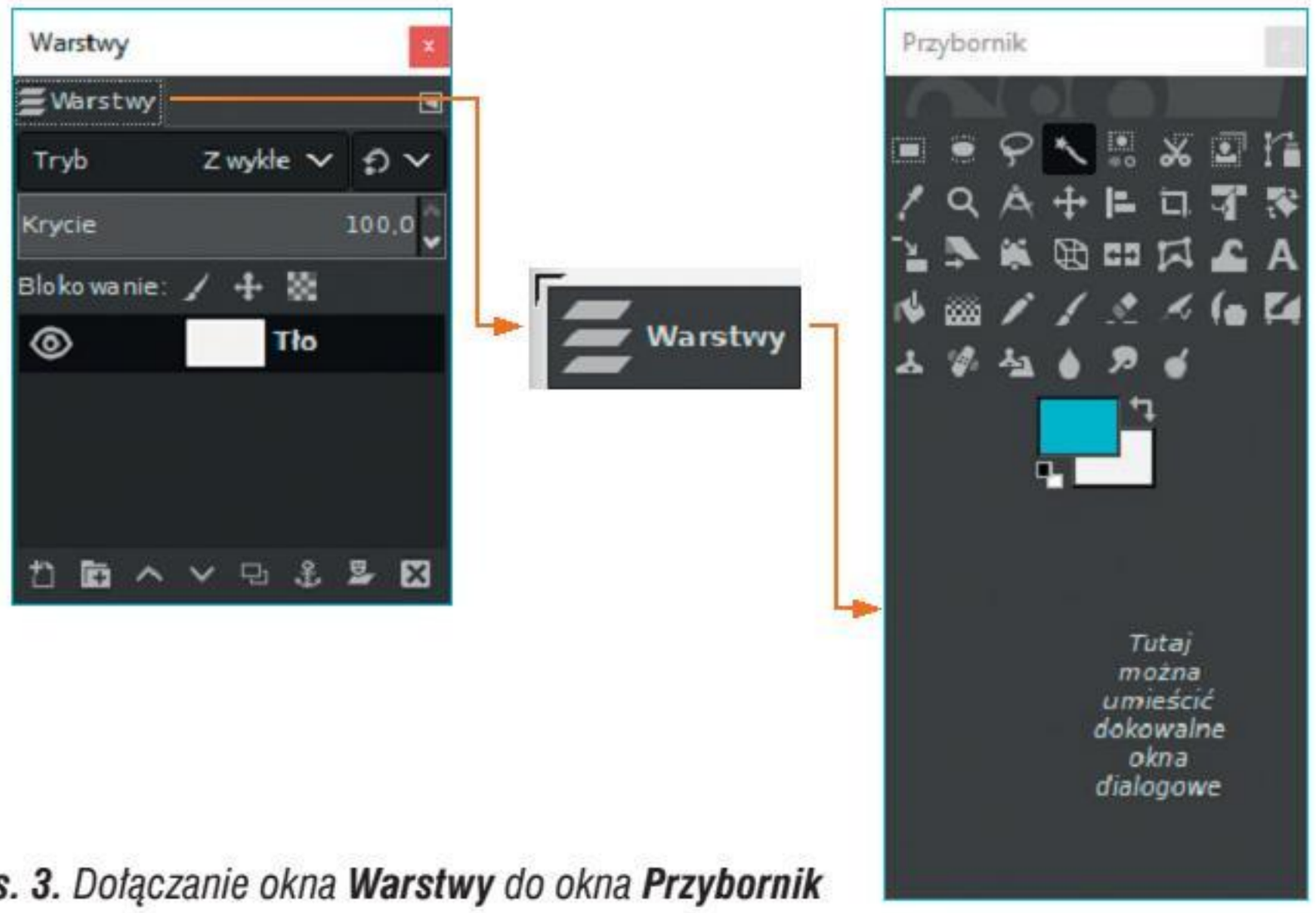
Zanim zaczniemy pracować na większej liczbie warstw, dołączmy okno **Warstwy** (z narzędziami do wykonywania operacji na warstwach) do **Przybornika**.

Aby dołączyć okno Warstwy do **Przybornika**, należy:

- w menu **Okna** wybrać opcję **Dokowalne okna dialogowe/Warstwy** – otworzy się okno **Warstwy**,
- przeciągnąć tytuł „Warstwy” na dolną część **Przybornika** (rys. 3.).

Uwaga

W przypadku pracy w trybie jednego okna po wybraniu w menu **Okna** opcji **Dokowalne okna dialogowe/Warstwy** okno **Warstwy** zostanie umieszczone od razu w oknie programu.



Rys. 3. *Dołączanie okna* ***Warstwy*** *do okna* ***Przybornik***

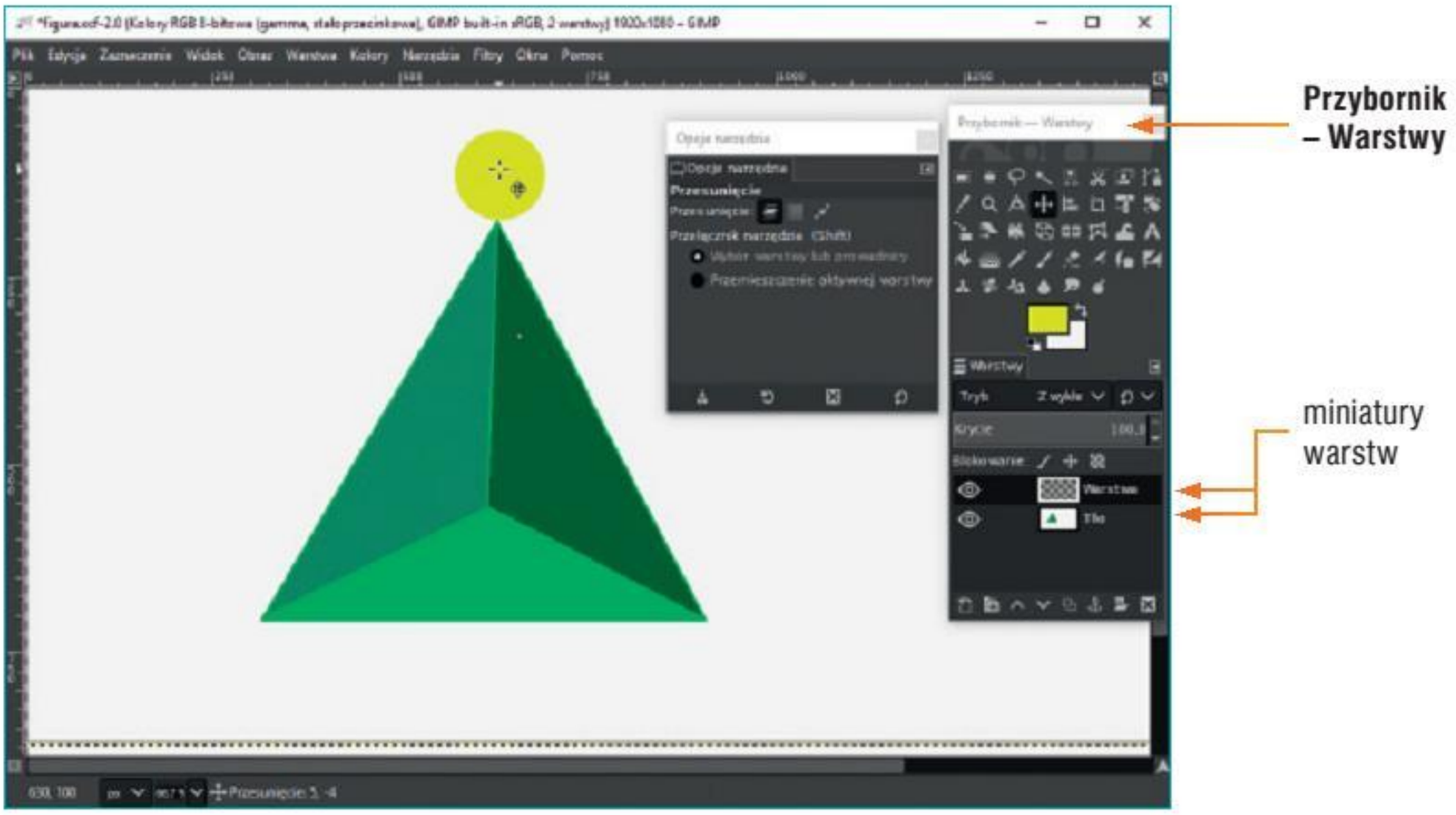
Aby dodać nową warstwę, należy skorzystać z przycisku w oknie z listą warstw lub wybrać opcję **Warstwa/Nowa warstwa**.

Aby usunąć warstwę, należy kliknąć jej nazwę prawym przyciskiem myszy i wybrać opcję **Usuń warstwę**.

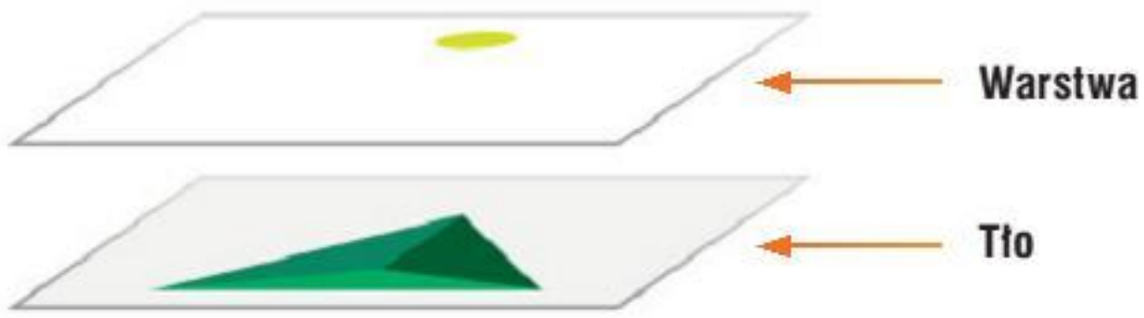
Po dołączeniu okna **Warstwy** do **Przybornika** układ warstw w edytowanym obrazie widzimy w oknie **Przybornik – Warstwy** (rys. 4.). Warstwa **Tło** jest nieprzezroczysta i znajduje się zawsze na końcu listy warstw. Warstwy obrazu są ułożone jedna na drugiej, w takiej kolejności, w jakiej widać je na liście warstw (rys. 4. i 5.). Nie każdy obraz musi zawierać tło – można usunąć tło z listy warstw.

Większość operacji wykonywanych w programie GIMP (np. rysowanie czy dodawanie efektów) dotyczy jedynie aktywnej warstwy, która jest podświetlona na liście warstw. Aby uaktywnić wybraną warstwę, należy kliknąć jej miniaturkę wskaźnikiem myszy.

Warstwy obrazu można względem siebie przesuwać, dzięki czemu zmieniamy położenie elementów obrazu. Do przesuwania warstw służy narzędzie **Przesuwanie** , umieszczone w **Przyborniku**. Warstwy można również obracać o określony kąt, korzystając z opcji **Warstwa/Przekształcenie**.



Rys. 4. *Przykład obrazu wykonanego na dwóch warstwach – praca w trybie wielu okien*



Rys. 5. *Schematycznie pokazany układ warstw*

Ćwiczenie 3. Tworzymy obraz z wykorzystaniem warstw

1. W programie GIMP narysuj na **Tle** figurę składającą się z trzech trójkątów wypełnionych różnym odcieniami koloru zielonego (rys. 4.). Boki trójkąta rysuj **Ołówkiem** z wciśniętym klawiszem **Shift**.
2. Dodaj drugą warstwę i narysuj na niej żółte koło.
3. Umieść koło na jednym z wierzchołków figury, przesuwając odpowiednio jedną z warstw (rys. 4.).
4. Zapisz rysunek w pliku pod nazwą *Figura*.

Wskazówka: Ćwiczenie można wykonać na dwa sposoby: przesuwając warstwę z kołem lub warstwę z zieloną figurą.

Uwaga

Podczas pracy nad obrazem można tymczasowo ukrywać wybrane warstwy – w tym celu wystarczy kliknąć przycisk z okiem  obok nazwy danej warstwy. Ponowne kliknięcie tego przycisku spowoduje, że warstwa znów będzie wyświetlana.

5. Stosowanie narzędzi selekcji w programie GIMP

Chcemy utworzyć ciekawy fotomontaż z dwóch zdjęć – wyciąć z jednego zdjęcia np. pasterza i umieścić na innym zdjęciu (np. wśród wieżowców ☺). Jak to zrobić w programie GIMP?

Jeśli podczas pracy nad obrazem chcemy zmienić jego wybrane fragmenty, korzystamy z **selekcji** (nazywanej również **zaznaczeniem**). Program GIMP udostępnia więcej narzędzi zaznaczania niż program Paint. Poza zaznaczaniem obszaru w kształcie prostokąta (kwadratu)  i zaznaczaniem odręcznym obszarów  umożliwia m.in.: zaznaczanie obszaru w kształcie elipsy (okręgu) , zaznaczanie według kolorów , zaznaczanie przyległego obszaru o zbliżonym kolorze  (**Różdżka**) oraz zaznaczanie obszarów przy użyciu wykrywania krawędzi  (**Inteligentne nożyce**).

Operacje, które w danym momencie wykonujemy na obrazie, dotyczą tylko aktywnej selekcji (aktywnego zaznaczenia). Rozpoczęcie nowej selekcji powoduje usunięcie poprzedniej.



Jeżeli chcemy zaznaczyć kilka fragmentów obrazu równocześnie, należy przed rozpoczęciem zaznaczania kolejnego fragmentu przytrzymać wciśnięty klawisz **Shift**.

Zawartość zaznaczenia można wyciąć (opcja **Wytnij**), a następnie wkleić (opcja **Wklej**) do tego samego lub innego obrazu otwartego w programie GIMP. Możemy również wykorzystać pracę z warstwami.

Stosowanie narzędzi selekcji pokażemy na przykładzie narzędzi **Różdżka** i **Inteligentne nożyce**. Narzędzia te umożliwiają uzyskiwanie ciekawych efektów (rys. 6a-6d, 7a-7b), szczególnie gdy dodatkowo zastosujemy pracę z warstwami obrazu (rys. 8a-8c).

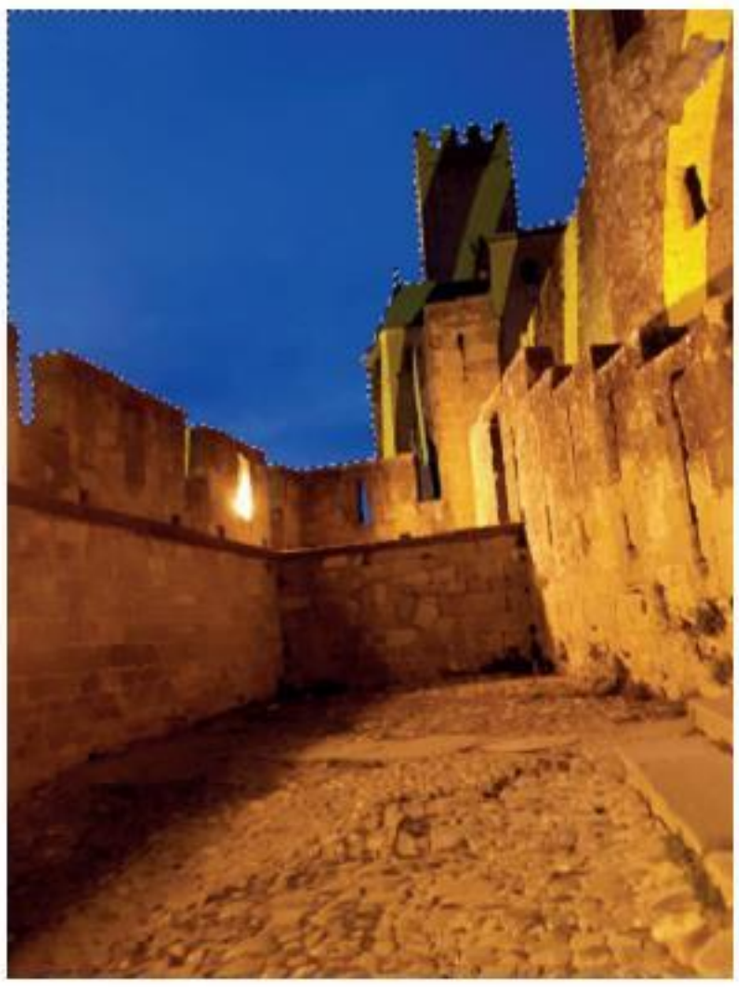
Zwykle **Różdżką** zaznacza się obszary o jednolitej barwie (np. niebo, trawę) po to, aby skopiować inny element, np. postać, samochód, dom, zamek, który znajduje się na tym tle. W tym celu wystarczy zaznaczyć **Różdżką** tło (rys. 6a), odwrócić zaznaczenie i skopiować lub wyciąć zamek (rys. 6b), dom lub postać.



Przykład 1. Stosowanie narzędzia **Różdżka**

1. Wybierz narzędzie **Różdżka**.
2. Trzymając wciśnięty klawisz **Shift**, klikaj tło (miejsca o zbliżonej barwie) wzdłuż krawędzi fragmentu obrazu, aby go zaznaczyć (rys. 6a).
3. Gdy potrzebny fragment (np. niebo) został zaznaczony, z menu kontekstowego obrazu wybierz polecenie **Zaznaczenie/Odwróć**.
4. Wytnij (**Ctrl** + **X**) lub skopiuj (**Ctrl** + **C**) zaznaczony fragment do **Schowka**.

5. Otwórz nowy obraz (np. z innym tłem) i wklej (**Ctrl** + **V**) do niego skopiowany do **Schowka** fragment pierwszego zdjęcia.
6. Kliknij poza zaznaczonym obszarem, aby zrezygnować z **Różdżki**.



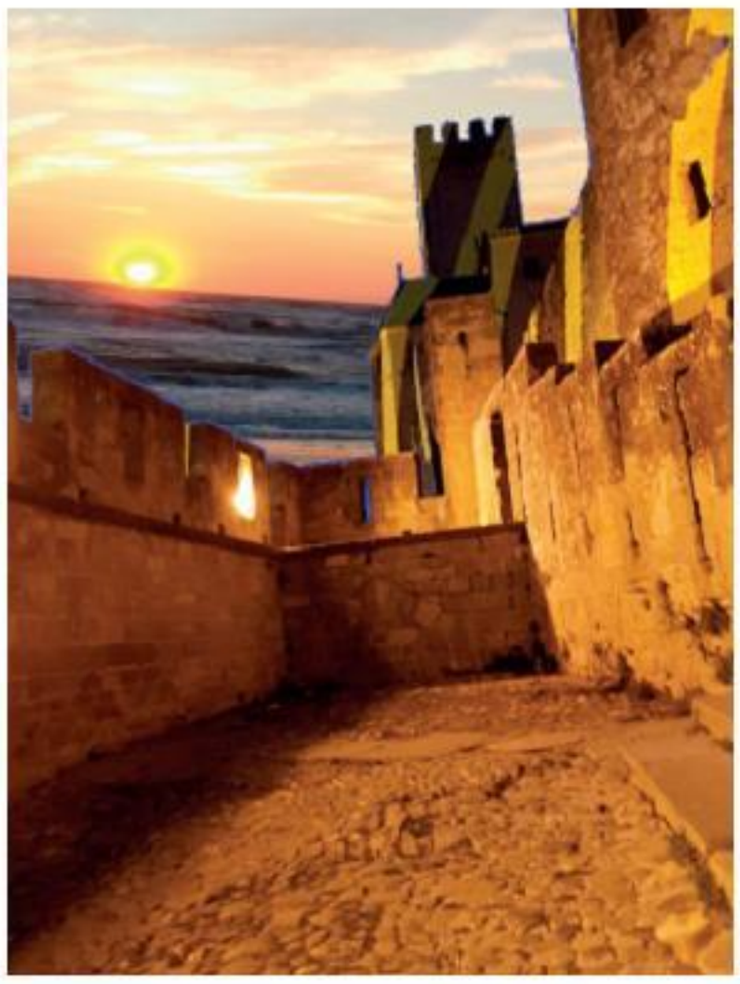
Rys. 6a. *Zamek w Carcassonne (Francja) – do zaznaczenia nieba zastosowano narzędzie* ***Różdżka***



Rys. 6b. *Rysunek 6a po odwróceniu zaznaczenia*



Rys. 6c. *Widok z zachodzącym słońcem*



Rys. 6d. *Zamek z rysunku 6a wklejony na inne tło*

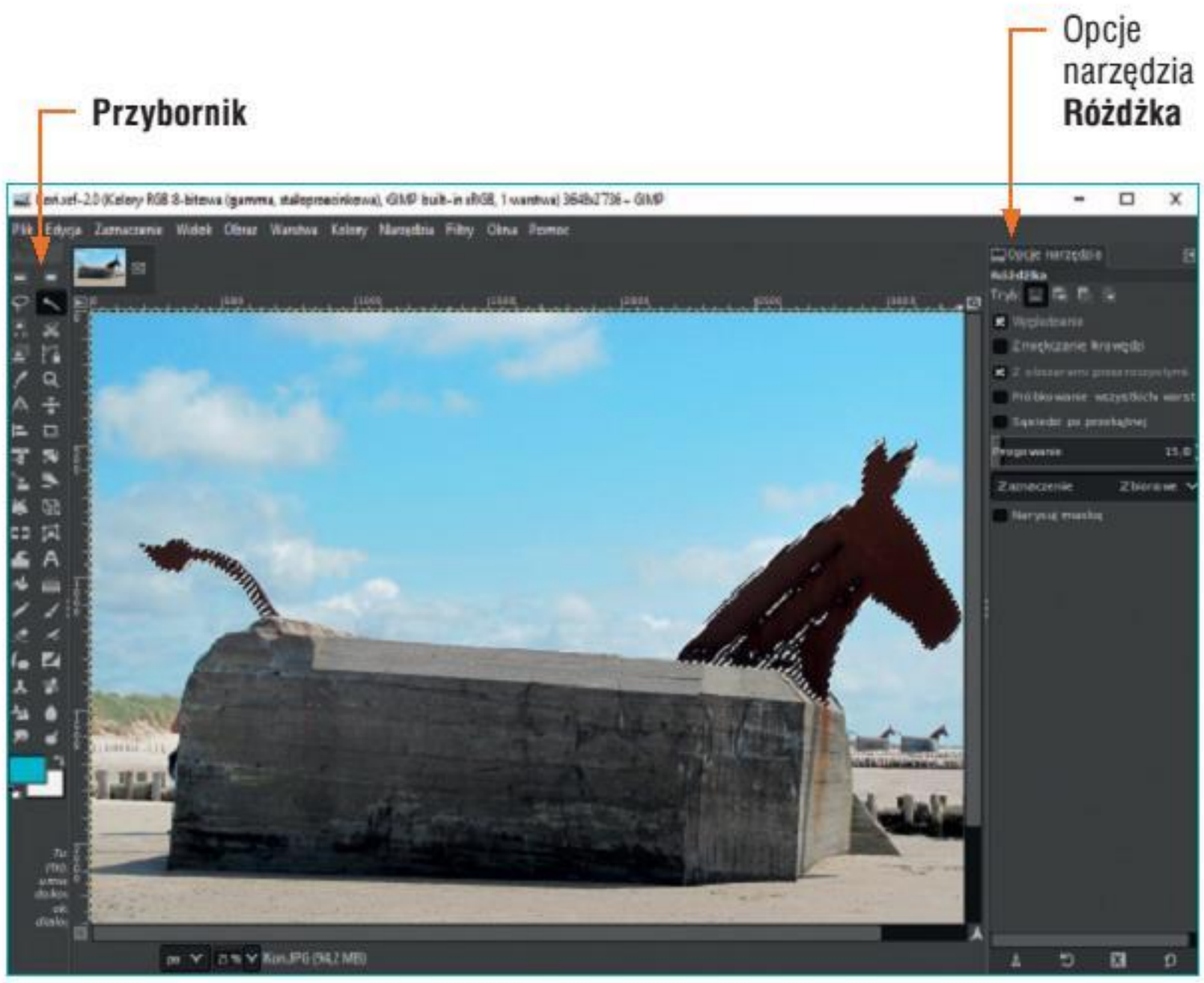
Ćwiczenie 4. Poznajemy narzędzie **Różdżka**

Znajdź w Internecie dwa zdjęcia: zamku na jednolitym tle, np. zamku w Carcasonne na tle niebieskiego nieba, i widoku z zachodzącym słońcem. Stosując narzędzie **Różdżka** (przykład 1.), zaznacz niebo tak, aby zaznaczyć kontur zamku (rys. 6a). Następnie odwróć zaznaczenie i wytnij zamek (rys. 6b). Wycięty fragment wklej na zdjęcie z zachodzącym słońcem (rys. 6d.).

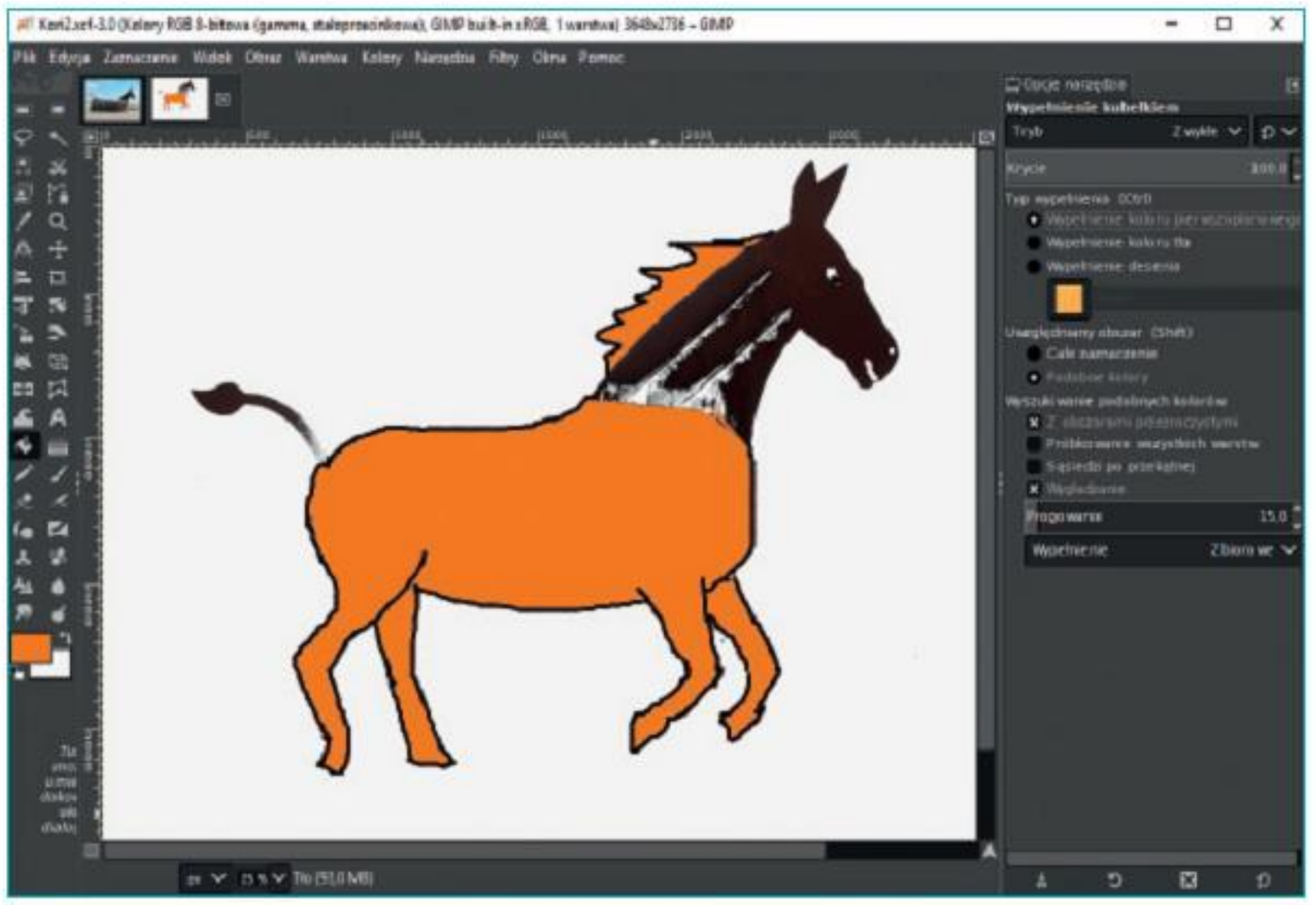
Na rysunkach 7a i 7b pokazano pracę w trybie jednego okna i inne zastosowanie narzędzia **Różdżka** niż na rysunkach 6a-6d. Po zaznaczeniu i skopiowaniu fragmentów zdjęcia z głową i ogonem konia (rys. 7a) wklejono je do nowego obrazu (rys. 7b).

Ćwiczenie 5. Korzystamy z narzędzia **Różdżka**

1. Znajdź w Internecie zdjęcie bunkrów na północnym wybrzeżu Danii, w Blavand. Otwórz je w programie GIMP. Zastosuj do głowy i ogona konia narzędzie **Różdżka**. Skopiuj zaznaczone fragmenty i wklej do nowego obrazu. Uzupełnij obraz, tak aby wklejone fragmenty były częścią rysunku (rys. 7b). Użyj narzędzi do rysowania.
2. Zapisz plik pod nazwą określającą zawartość obrazu.



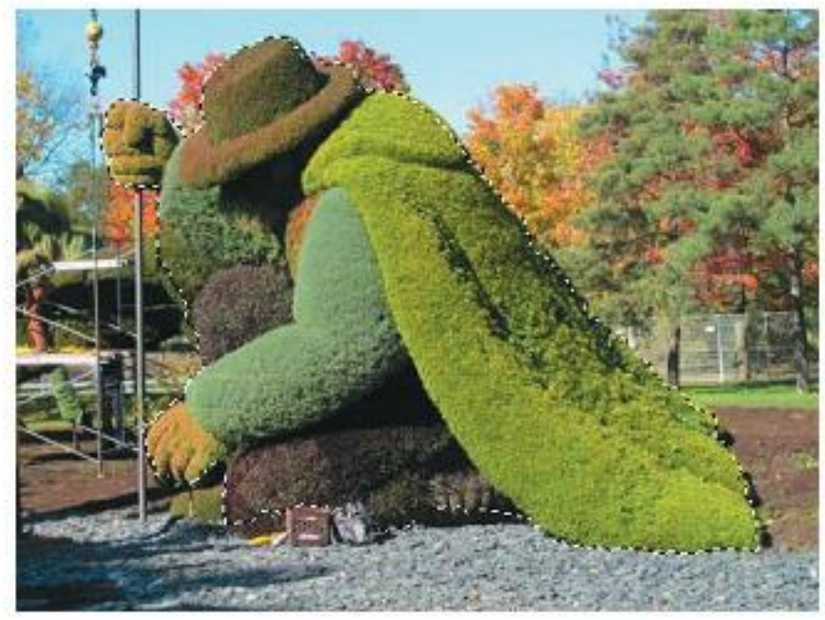
Rys. 7a. *Zastosowanie narzędzia* ***Różdżka*** *do zaznaczenia głowy i ogona konia (Dania, Blavand, bunkry nad Morzem Północnym)*



Rys. 7b. *Fragmenty rysunku 7a z głową i ogonem konia wklejone do nowego obrazu; rysunek konia dokończony za pomocą* ***Pędzla*** *i* ***Wypełnienia kubełkiem***



Rys. 8a. *Zaznaczanie fragmentu zdjęcia z wykorzystaniem narzędzia* ***Inteligentne nożyce*** *(Kanada, Ogród Botaniczny w Montrealu, pasterz wykonany z roślin)*



Rys. 8b. *Zaznaczony fragment zdjęcia po zastosowaniu narzędzia* ***Inteligentne nożyce***



Rys. 8c. *Efekt fotomontażu (Kanada, w tle wieżowce w Mississaudze)*

Przykład 2. Stosowanie narzędzia **Inteligentne nożyce** do zaznaczenia fragmentu obrazu

1. Wybierz narzędzie **Inteligentne nożyce**.
2. Klikaj wzdłuż krawędzi danego fragmentu, aby powstawiać punkty zwane **węzłami** (rys. 8a) – program dorysuje linię łączącą nowy punkt z poprzednim, dopasowując ją do krawędzi fragmentu obrazu.
3. Zakończ stosowanie narzędzia **Inteligentne nożyce**, klikając ponownie pierwszy węzeł.
4. Zmień kontur z wstawionymi węzłami w zaznaczenie, klikając wewnątrz zaznaczonego obszaru (rys. 8b).

Ćwiczenie 6. Stosujemy narzędzie **Inteligentne nożyce** i pracę z warstwami

1. Znajdź w Internecie dwa zdjęcia: jedno z wieżowcami z Warszawy, a drugie przedstawiające instalację z kwiatów z Ogrodu Botanicznego w Montrealu. Korzystając z możliwości pracy z warstwami obrazu i narzędzia selekcji **Inteligentne nożyce**, utwórz w programie GIMP ciekawy fotomontaż (przykład na rys. 8a-8c.).
2. Zapisz plik pod nazwą określającą zawartość obrazu.

6. Edycja zdjęć w programie GIMP

Zrobiliśmy na wakacjach zdjęcia. Niektóre ze zdjęć okazały się zbyt ciemne, inne pozbawione artystycznej ekspresji. W jaki sposób można je poprawić, korzystając z programu GIMP?

6.1. Wybrane filtry i efekty

Uwaga

Zależnie od wersji programu GIMP, filtry i inne polecenia modyfikowania obrazu mogą mieć różne (choć zbliżone) nazwy.

Program GIMP oferuje bogaty zestaw **filtrów**, które można stosować do obrazów, m.in.: **Uwydatnianie**, **Zniekształcenia**, **Szum**, **Artystyczne**.

Stosując filtry, należy pamiętać, że działają one jedynie na aktywnej warstwie i w obszarze zaznaczenia.

Aby zastosować wybrany filtr, należy z menu programu GIMP wybrać opcję **Filtry**, a następnie – odpowiedni filtr.

Po zmianie rozmiarów obrazu często staje się on rozmazany. W takim przypadku może pomóc filtr wyostrzający, który znajdziemy w opcji **Uwydatnianie/Wyostrz**.

Ćwiczenie 7. Poznajemy filtry w programie GIMP

1. Zapoznaj się z możliwościami stosowania wybranych filtrów w programie GIMP.
2. Znajdź w Internecie przykładowe rozmazane zdjęcie i wyostrz je, stosując filtr wyostrzający.
3. Znajdź jeszcze jedno zdjęcie w Internecie lub we własnych zbiorach i zależnie od jakości zdjęcia wypróbuj inny filtr.



Rys. 9. *Po lewej stronie zdjęcie wykonane w Kanadzie, gdzie rośnie około stu gatunków klonów (w Polsce tylko kilka); liść klonu widnieje na fladze Kanady. Po prawej stronie – efekt nałożenia dwóch filtrów –* ***Farba olejna*** *i* ***Film rysunkowy***

Ćwiczenie 8. Stosujemy filtry **Farba olejna** i **Film rysunkowy**

1. Znajdź w Internecie zdjęcia liści klonu z Kanady.
2. Wykonaj przekształcenia zdjęcia, stosując dwa filtry: **Farba olejna** i **Film rysunkowy**.

6.2. Zmiana kontrastu i nasycenia kolorów

Obrazy importowane ze skanera bądź aparatu cyfrowego często są zbyt jasne (prześwietlone), zbyt ciemne (niedoświetlone) albo mają nienaturalne, przekłamane kolory. Używając narzędzi programu GIMP, można dokonać stosownych korekt.

Program GIMP posiada różnego rodzaju narzędzia umożliwiające zmianę parametrów obrazu, dostępne w menu **Kolory**. Dodatkowe informacje można uzyskać w **Pomocy** programu.

Aby zmienić nasycenie kolorów, można użyć opcji **Kolory/Barwa i nasycenie**. Aby dostosować jasność i kontrast obrazu, można skorzystać z regulacji poziomów barwy, nasycenia i jasności.

Ćwiczenie 9. Zmieniamy kontrast i nasycenie kolorów

Znajdź w Internecie niedoświetlone zdjęcie. Korzystając z narzędzi programu GIMP, popraw kontrast i nasycenie kolorów na tym zdjęciu.

6.3. Kadrowanie i skalowanie

Kadrowanie umożliwia przycięcie obrazu w celu wyeksponowania jedynie fragmentu obrazu, który jest dla nas istotny, lub pozbycia się np. niepożądanego obramowania.

Przykład 3. Kadrowanie obrazu

1. W **Przyborniku** kliknij narzędzie **Kadrowanie** .
2. Utwórz prostokąt w obszarze obrazu (podobnie jak rysuje się prostokąt w programie Paint) – w narożnikach i na bokach prostokąta pojawią się małe kwadraty umożliwiające zmianę rozmiarów kadrowanego obszaru.
3. Gdy zaznaczysz potrzebny obszar obrazu, kliknij wewnątrz zaznaczenia, aby przyciąć obraz.

Uwaga: Polecenie kadrowania można również wybrać, klikając opcję **Narzędzia/Narzędzia przekształcania**.



Rys. 10. *Przykład kadrowania zdjęcia*

Ćwiczenie 10. Kadrujemy obraz

Znajdź w Internecie zdjęcie przedstawiające dom. Otwórz je w programie GIMP i wykadruj według własnego pomysłu.

Skalowanie umożliwia powiększenie lub pomniejszenie obrazu. Program GIMP pozwala zmieniać wymiary całego obrazu bądź pojedynczych warstw. Należy jednak pamiętać, że skalowanie powoduje utratę jakości obrazu (dostrzegalną lub nie, zależnie od stopnia skalowania).

Ćwiczenie 11. Skalujemy zdjęcie

1. Odszukaj w **Przyborniku** lub w menu **Narzędzia** polecenie skalowania.
2. Otwórz w programie GIMP zdjęcie wybrane z własnych zbiorów lub wyszukane w Internecie. Stosując skalowanie, zwiększ rozmiar obrazu trzykrotnie, a potem zmniejsz trzykrotnie. Zwróć uwagę na jakość obrazu po każdorazowym skalowaniu. Jakie są twoje spostrzeżenia?

Warto zapamiętać

- Grafikę komputerową dzielimy na grafikę rastrową i grafikę wektorową. Możemy także wyróżnić grafikę dwuwymiarową i trójwymiarową.
- Zapis pliku graficznego w określonym formacie należy dostosować do przeznaczenia danego pliku, uwzględniając wady i zalety wybranego formatu.
- Do zapisywania zdjęć stosujemy najczęściej format JPEG. Format PNG stosujemy między innymi do zapisywania obrazów umieszczanych na stronach WWW.
- Przykładem programu do opracowywania grafiki rastrowej jest program GIMP.
- Program GIMP oferuje wiele narzędzi do rysowania, zaznaczania i przekształcania obrazu. Prostokąty i elipsy można rysować, stosując narzędzia zaznaczania.
- W programie GIMP możemy tworzyć obraz, stosując warstwy. Na każdej można narysować inny element obrazu.

- Warstwy można porównać do nałożonych na siebie przezroczystych folii.
- Dzięki wykorzystaniu różnych narzędzi selekcji (np. **Inteligentnych nożyc** lub **Różdżki**) i pracy na warstwach możemy tworzyć kompozycje graficzne i fotomontaże.
- Narzędzia modyfikowania obrazu w programie GIMP umożliwiają m.in. wyostrzenie obrazu, zmianę kontrastu i nasycenia kolorów, wykadrowanie i zmianę rozmiaru obrazu.

Pytania i polecenia

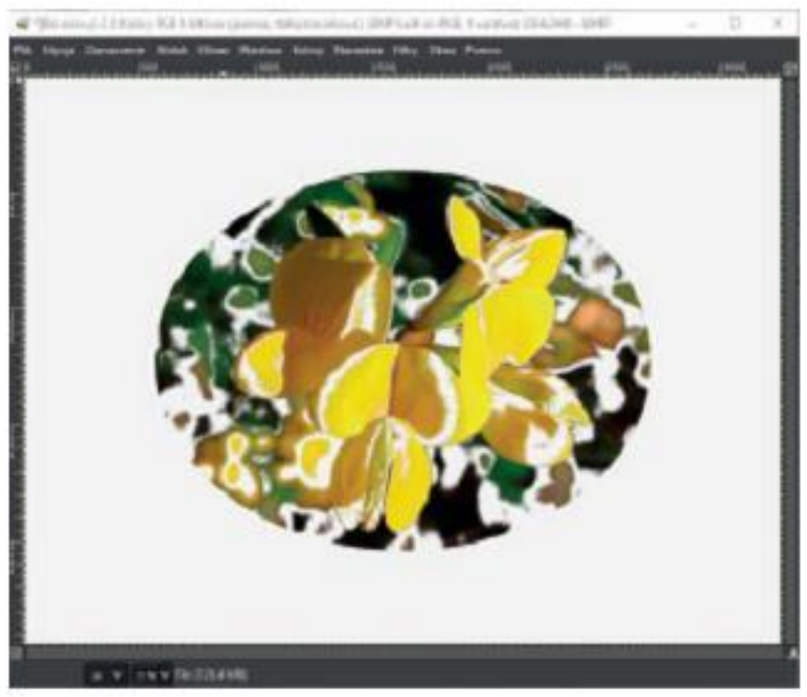
1. Podaj przykłady zastosowań grafiki komputerowej.
2. Czym różni się grafika rastrowa od grafiki wektorowej? W jaki sposób zapisywany jest obraz w pamięci komputera w przypadku każdego z tych rodzajów grafiki?
3. Omów popularne formaty plików graficznych, ich zalety oraz zastosowania.
4. W jakim formacie zapiszesz zdjęcie z aparatu cyfrowego, aby wysłać je jako załącznik w liście e-mail? Uzasadnij odpowiedź.
5. W jaki sposób w programie GIMP narysować kwadrat, a jak – linię prostą?
6. Czym są warstwy obrazu? Wyjaśnij na przykładzie pracy z warstwami w programie GIMP.
7. Omów sposób dodawania nowej warstwy do obrazu.
8. Jaki jest efekt działania narzędzia **Różdżka**? Wyjaśnij na przykładzie.
9. Do czego służą **Inteligentne nożyce**? Wyjaśnij na przykładzie.
10. Wymień i omów przykładowe możliwości poprawy jakości zdjęcia.
11. W jaki sposób wykadrować zdjęcie?

Zadania

1. Otwórz plik *Ziemia* zapisany w ćwiczeniu 2. Dorysuj statek kosmiczny krążący wokół Ziemi. Zapisz plik pod nazwą *Ziemia i pojazd*.
2. W programie GIMP narysuj sześć figur geometrycznych z konturem w kolorze granatowym: prostokąt, kwadrat, elipsę, koło, trójkąt, pięciokąt. Zapisz plik pod nazwą *Figury*.
3. Otwórz plik *Figury* zapisany w zadaniu 2. Odszukaj w **Przyborniku** narzędzie do umieszczania napisów. Podpisz każdą figurę. Dobierz samodzielnie parametry czcionki. Zapisz plik pod tą samą nazwą.
   Wskazówka: W programie GIMP tekst umieszczany jest na oddzielnej warstwie. Gdy skończymy pracę nad tekstem, możemy warstwę tekstową połączyć z obrazem, czyli warstwą leżącą poniżej, wybierając z menu kontekstowego aktywnej warstwy polecenie **Warstwa/Połącz w dół**.
4. Otwórz plik *Figura* zapisany w ćwiczeniu 3. Powiel dwukrotnie warstwę z kołem. Przesuń nowe warstwy tak, aby koła znalazły się na dwóch pozostałych wierzchołkach figury. Zapisz plik pod nazwą *Figura z kołami*.
   Wskazówka: Aby powielić warstwę, należy z menu kontekstowego miniaturki warstwy wybrać polecenie **Powiel warstwę**.
5. W programie GIMP narysuj na pierwszej warstwie (na **Tle**) prostokąt wypełniony kolorem fioletowym, a na drugiej – kwadrat koloru szarego (bok kwadratu musi być mniejszy niż krótszy bok prostokąta). Umieść kwadrat wewnątrz prostokąta, przesuwając odpowiednią warstwę. Odszukaj w **Przyborniku** przyciski służące do przekształcania warstw, m.in. nachylania, obracania. Obróć o 90 stopni w prawo warstwę, na której jest umieszczony prostokąt, a następnie przesuń tak, aby kwadrat był nadal wewnątrz niego. Zapisz rysunek w pliku pod nazwą *Kwadrat w prostokącie*.



***Rys. 11a**. Zastosowanie narzędzia **Zaznaczanie eliptyczne** i **Zaznaczanie według kolorów** – zadanie 6.*



***Rys. 11b**. Fragment zdjęcia z rysunku 11a wklejony do nowego obrazu – zadanie 6.*

6. Na zdjęciu pokazanym na rysunku 11a zastosowano narzędzie **Zaznaczanie eliptyczne**, a następnie w obszarze elipsy użyto narzędzia **Zaznaczanie według kolorów** (klikając kilkakrotnie bez przytrzymywania klawisza **Shift**). Po skopiowaniu zaznaczonych fragmentów z oryginalnego zdjęcia (rys. 11a) wklejono je do nowego obrazu (efekt wklejenia pokazano na rysunku 11b). Poszukaj w swoich zbiorach przykładowego zdjęcia (np. kwiatów, samochodu) i przygotuj w podobny sposób nowy obraz. Zapisz plik pod nazwą określającą zawartość obrazu.
7. Wybierz ze swoich zbiorów dwa zdjęcia. Korzystając z możliwości pracy z warstwami obrazu i narzędzia selekcji **Inteligentne nożyce**, utwórz w programie GIMP ciekawy fotomontaż. Zapisz plik pod nazwą określającą zawartość obrazu.
8. Poszukaj w Internecie lub w swoich zbiorach trzech zdjęć: zbyt ciemnego, zbyt jasnego i nieostrego. Popraw zdjęcia, stosując możliwości modyfikowania obrazu w programie GIMP.

Dla zainteresowanych

9. W programie GIMP narysuj piramidę rozwoju Internetu (temat D1). Na najniższej warstwie (**Tło**) narysuj piramidę (możesz wykorzystać zdjęcie piramidy), a na drugiej warstwie – bloki, do których na kolejnych warstwach wpiszesz odpowiednie teksty z tematu D1 (możesz dodać inne usługi lub wydarzenia związane z rozwojem Internetu). Zapisz plik pod nazwą *Piramida Internetu*.
10. Zaprojektuj w programie GIMP logo szkolnego klubu sportowego.
11. W programie GIMP narysuj na kilku warstwach układ słoneczny.
12. W programie GIMP można przygotować animacje, korzystając z powielania warstw. Przygotuj dowolną animację składającą się z kilku warstw-klatek. Na pierwszej warstwie narysuj rysunek bazowy, następnie powiel warstwę i zmień częściowo rysunek (stosując np. przekształcenia czy inne kolory). Powtarzaj powielanie warstw i nanoszenie zmian tyle razy, ile klatek chcesz utworzyć. Zapisz plik pod nazwą *Animacja*.

    Wskazówki:
    - Animację możemy zapisać jako obraz w formacie *xcf* (opcja **Plik**/**Zapisz jako**). Można ją również wyeksportować do pliku o innym formacie, np. GIF, korzystając z opcji **Plik**/**Wyeksportuj jako**.
    - Aby odtworzyć animację, należy wybrać opcję **Filtry**/**Animacja**/**Odtwarzanie**.
13. Wybierz zdjęcie ze swoich zbiorów lub pobierz z Internetu. Przekształć je w ciekawy sposób, stosując narzędzia programu GIMP.

Temat B4. Zasady tworzenia prezentacji multimedialnej

1. Typy prezentacji multimedialnych
2. Etapy przygotowywania prezentacji multimedialnej
   - 2.1. Planowanie prezentacji
   - 2.2. Tworzenie prezentacji
   - 2.3. Dodawanie dźwięków i filmów
   - 2.4. Zapisywanie prezentacji
   - 2.5. Drukowanie materiałów informacyjnych i notatek
   - 2.6. Ustalenie parametrów pokazu i pokaz
3. Tworzymy prezentację multimedialną o Fryderyku Chopinie
4. Importowanie konspektu prezentacji z dokumentu tekstowego

Warto powtórzyć

1. Jakie znasz zasady przygotowywania prezentacji multimedialnych?
2. W jaki sposób tworzy się konspekt w edytorze tekstu?
3. Czym jest szablon dokumentu?

1. Typy prezentacji multimedialnych

Prezentacje multimedialne są powszechnie stosowane w edukacji i w biznesie. Wykorzystuje się je w celach komunikacyjnych lub marketingowych. Pomagają przybliżyć słuchaczom temat i zwrócić ich uwagę na najważniejsze informacje.

Tworząc prezentację, należy określić jej cel: czy ma być pomocą dla prelegenta, służyć do samodzielnego przeglądania przez odbiorcę, czy też być prezentowana w tle (prezentacja typu kiosk).

P **Prezentacja multimedialna**

Przedstawienie danego zagadnienia w postaci elektronicznego dokumentu, zawierającego zwykle: tekst, obrazy, wykresy, schematy, dźwięki, filmy.

Prezentacja wspomagająca wystąpienie prelegenta

Prezentacja wspomagająca wystąpienie jest uzupełnieniem przekazu prelegenta, np. ucznia lub nauczyciela na lekcji, wykładowcy na uczelni lub podczas konferencji czy członka zarządu firmy przedstawiającego wyniki finansowe pracownikom.

Należy pamiętać, że prezentacja jest w tym przypadku jedynie pomocą w wystąpieniu. Aby skupić uwagę słuchaczy, treść slajdu powinna zawierać tylko główne wątki tematu – pozostałe informacje przekaże prelegent.

Prezentacja do samodzielnego przeglądania przez odbiorcę

Prezentacje, które odbiorca przegląda samodzielnie, stosuje się najczęściej przy broszurach informacyjnych i reklamowych. W tym przypadku prezentacja może zawierać więcej tekstu, ale komunikaty powinny być zwięzłe.

Przygotowując prezentację, należy być kreatywnym – przedstawić informacje w taki sposób, żeby zainteresować odbiorcę. W związku z tym, że odbiorca sam decyduje, w jakim tempie i w jakim zakresie przegląda materiał, treść powinna być interesująca, aby uniknąć sytuacji, w której odbiorca wyłączy prezentację przed jej zakończeniem.

W formie prezentacji do samodzielnego przeglądania można przygotować na przykład ofertę handlową. Prezentacja umożliwia przekazanie informacji w postaci wykresów, schematów. Możemy dodać grafikę (zdjęcia, filmy) i animacje, przedstawiając daną ofertę w zachęcający sposób. Powinniśmy jednak unikać zbyt dużej liczby animacji. Należy też przestrzegać zasady, że tego rodzaju prezentacja powinna zawierać tylko kilka slajdów. Na każdym slajdzie musi być przedstawiona konkretna informacja.

Slajd powinien zawierać pełną informację, choć przedstawioną w skróconej formie. Jeśli konieczne jest wprowadzenie dłuższego tekstu, należy go rozmieścić na kilku slajdach.

Prezentacja typu kiosk – samouruchamiająca się

Prezentacja samouruchamiająca się jest wyświetlana podczas wystawy, konferencji lub w galerii handlowej. Po zakończeniu pokazu prezentacja automatycznie ponownie się uruchamia – stąd jej nazwa „samouruchamiająca".

Podczas przygotowywania takiej prezentacji istotne jest dobranie odpowiedniego sposobu i czasu animacji – slajdy nie mogą zmieniać się zbyt szybko ani zbyt wolno, żeby odbiorca nie stracił zainteresowania. Najlepiej przetestować przygotowaną prezentację na małej grupie osób, które pierwszy raz mają styczność z tematem.

Slajd powinien zawierać pełną informację, ale przedstawioną skrótowo. Podobnie jak w pozostałych typach prezentacji, należy unikać nadmiaru tekstu na jednym slajdzie.



Jeśli w prezentacji wykorzystujemy cudze materiały (teksty, obrazy, pliki dźwiękowe, filmy), należy podać ich źródło (np. adres strony internetowej, tytuł publikacji) oraz nazwisko i imię autora (jeśli je znamy). Źródło można umieścić na danym slajdzie lub na końcu prezentacji w postaci listy źródeł.

Niezależnie od typu prezentacji ważne jest przedstawienie informacji w przystępny sposób, tak by nie przytłoczyć odbiorcy nadmiarem treści.



Slajdy powinny zawierać jak najwięcej treści przy jak najmniejszej ilości tekstu.

2. Etapy przygotowywania prezentacji multimedialnej

Chcemy przygotować prezentację multimedialną o zasadach tworzenia prezentacji. Prezentacja ma wspomagać nasze wystąpienie. Jak ją najlepiej wykonać i nie przytłoczyć słuchaczy nadmiarem treści?

Prezentację będziemy przygotowywać w sześciu etapach.

1. Planowanie prezentacji:
   - określenie celu i tytułu prezentacji,
   - ustalenie liczby slajdów,
   - wybór tekstów i ilustracji,
   - ustalenie kolorystyki i rozmieszczenia elementów (np. wybranie szablonu).
2. Tworzenie prezentacji:
   - przygotowanie slajdu tytułowego,
   - przygotowanie pozostałych slajdów,
   - wpisanie lub wstawienie tekstów (m.in. z wykorzystaniem grafiki SmartArt, ikon),
   - umieszczenie obrazów,
   - dodanie animacji i ustalenie sposobu wyświetlania slajdów.
3. Dodawanie dźwięków i filmów.
4. Zapisanie prezentacji.
5. Drukowanie materiałów informacyjnych i notatek (opcjonalnie).
6. Ustalenie parametrów pokazu i pokaz.

W kolejnych punktach pokażemy zasady przygotowania prezentacji multimedialnej w programie PowerPoint. Wykorzystamy rysunki (m.in. grafikę **SmartArt**, ikony dostępne w programie PowerPoint), animacje, zdjęcia i wykresy, które urozmaicą prezentację oraz przyciągną uwagę odbiorców.



Rys. 1. *Okno* ***Wybieranie grafiki SmartArt*** *(PowerPoint 2016)*

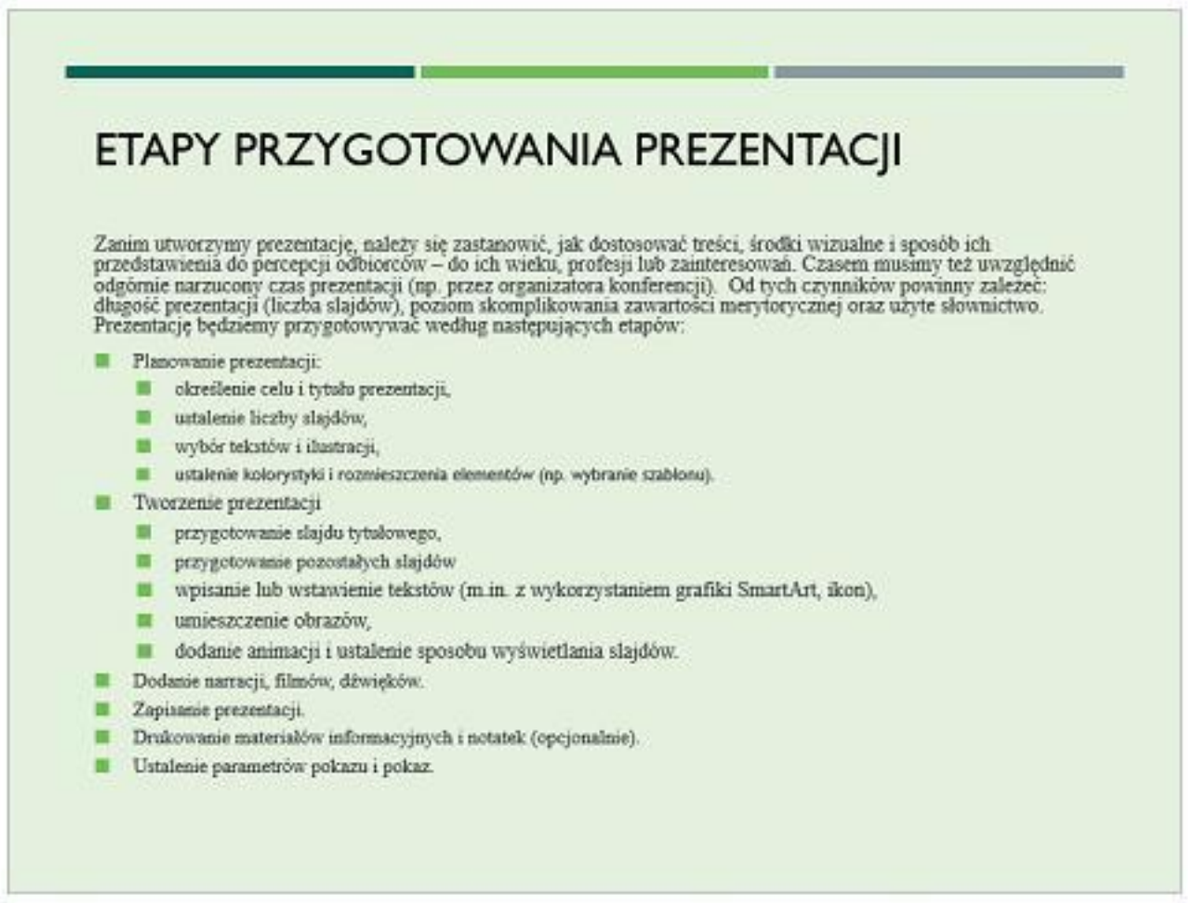


Rys. 2. *Okno* ***Wstawianie ikon*** *(PowerPoint 2016)*

Na rysunkach 3a i 3b pokazano przykładowy sposób przedstawienia na slajdach etapów tworzenia prezentacji. W punktach 2.1-2.6 omówiono poszczególne etapy, a na rysunkach 4-9 pokazano slajdy prezentujące odpowiednio treści każdego punktu.



Rys. 3a. *Slajd „Etapy przygotowywania prezentacji" – na slajdzie wykorzystano grafikę SmartArt i ikony*



Rys. 3b. *Slajd „Etapy przygotowania prezentacji" – na slajdzie umieszczono cały tekst i zastosowano wypunktowanie*

Ćwiczenie 1. Zapoznajemy się z zasadami tworzenia prezentacji, korzystając ze slajdów

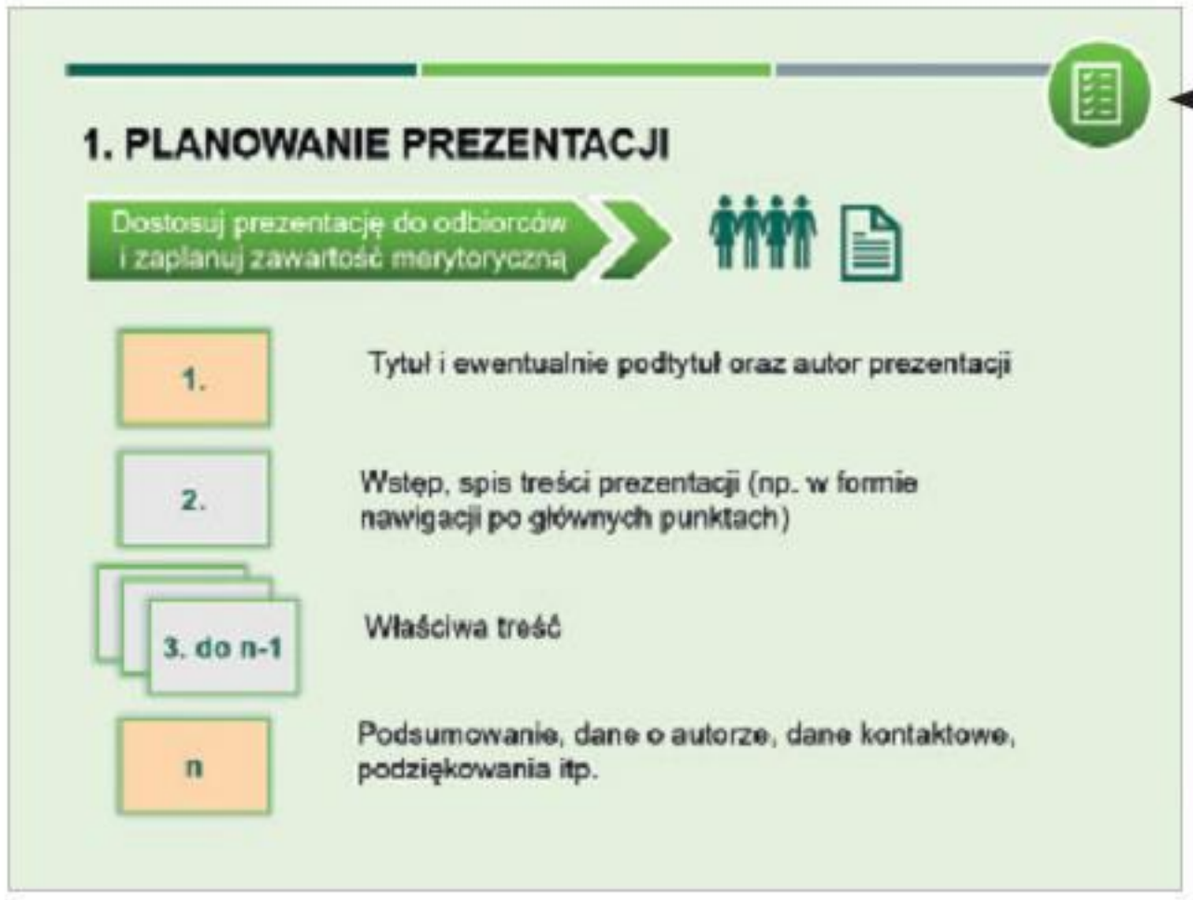
Na rysunkach 3a i 3b pokazano dwie wersje tego samego slajdu. Jak oceniasz przedstawienie tych samych treści na obydwu slajdach? Który slajd można wybrać jako plan (spis treści) prezentacji? Uzasadnij odpowiedź. Zapoznaj się z treścią pozostałych slajdów pokazanych na rysunkach 4-9.

Ćwiczenie 2. Zapoznajemy się z zasadami tworzenia prezentacji, korzystając z treści punktów

Przeczytaj treść punktów 2., 2.1-2.6 i porównaj ją z treścią slajdów z odpowiednich rysunków.

Zanim utworzymy prezentację, należy się zastanowić, jak dostosować treści, środki wizualne i sposób ich przedstawienia do percepcji odbiorców – do ich wieku, profesji lub zainteresowań. Czasem musimy też uwzględnić odgórnie narzucony czas prezentacji (np. przez organizatora konferencji). Od tych czynników powinny zależeć: długość prezentacji (liczba slajdów), poziom skomplikowania zawartości merytorycznej oraz użyte słownictwo.

2.1. Planowanie prezentacji



Na kolejnych slajdach można umieścić taki sam element graficzny jak wstawiony przy odpowiadającym mu punkcie planu prezentacji (rys. 3a). Takie elementy mogą pełnić funkcję nawigacji po punktach prezentacji.

Rys. 4. *Slajd „Planowanie prezentacji"*

Istotnym etapem tworzenia prezentacji jest planowanie zawartości merytorycznej. Należy określić sposób rozmieszczenia informacji na slajdach oraz elementy graficzne. Warto poświęcić czas na planowanie, ponieważ im lepiej przemyślimy koncepcję, tym mniej czasu zajmie sama praca nad tworzeniem slajdów.

Przykładowy plan:

- **pierwszy slajd** – tytuł (ewentualnie podtytuł) oraz imię i nazwisko autora prezentacji;
- **drugi slajd** – wstęp, spis treści prezentacji (widzowie od razu będą mogli się zorientować, jaka jest jej zawartość);
- **kolejne slajdy** – poszczególne wątki, czyli właściwa treść;
- **ostatni slajd** – podsumowanie (może to być streszczenie informacji, które widzowie mają zapamiętać), dane o autorze, dane kontaktowe, podziękowania itp.

Jeśli mamy ograniczony czas na przedstawienie prezentacji, podczas planowania najlepiej przewidzieć od pół minuty do kilku minut na każdy slajd. Prezentacja nie może zawierać np. 50 slajdów, jeśli wystąpienie ma trwać 20 minut. Odbiorca musi mieć czas na zapoznanie się z treścią każdego slajdu.

2.2. Tworzenie prezentacji



Rys. 5. *Slajd „Tworzenie prezentacji"*

W polach tekstowych na prezentacji możemy umieszczać teksty, obrazy i animacje. Możemy też określić sposób wyświetlania animacji, dodawać przejścia slajdów, przyciski akcji i hiperłącza. Nie każda prezentacja musi zawierać wszystkie te elementy.

Prezentacja powinna mieć jednolitą oprawę graficzną. Slajdy należy tworzyć w podobnym stylu i spójnej kolorystyce.

Aby na wszystkich slajdach odpowiadające sobie pola tekstowe i inne obiekty były jednakowo rozłożone i jednolicie pokolorowane, najlepiej wybrać gotowy szablon (**Plik/Nowy**) lub motyw (**Projektowanie/Motywy**) oraz kolor proponowany przez program i konsekwentnie je stosować.

Przedstawiając wybrane treści, zwłaszcza ukazujące pewne procesy, np. rozwój w czasie, etapy tworzenia – warto wykorzystać grafikę SmartArt i ikony (rys. 3a).



Materiały umieszczane na slajdach powinny pomóc odbiorcy skupić uwagę, a nie ją rozpraszać.

Zasady dodawania tekstu

- Nie wolno przesadzać z ilością tekstu. Warto kierować się zasadą „mniej znaczy więcej" – odbiorca nie powinien czuć się przytłoczony nadmiarem informacji, należy skupić się na najważniejszych kwestiach.
- Jeśli nie korzystamy z gotowego szablonu lub motywu, należy tak dobrać rozmiar i krój czcionki, aby tekst był czytelny. Nie należy używać wielu różnych atrybutów czcionek (zwłaszcza na jednym slajdzie) – najlepiej zdecydować się na co najwyżej dwa kroje pisma. Rozmiar czcionki powinien być odpowiednio duży (trzeba pamiętać, że prezentacja może być oglądana przez widzów z daleka). Aby zmienić czcionkę na kilku slajdach, należy je zaznaczyć i wybrać opcję **Narzędzia główne/Zamień/Zamień czcionki**.
- Należy umiejętnie stosować kolory (czcionki i tła) – tekst nie powinien zlewać się z tłem.

- Powinniśmy dbać o poprawność gramatyczną i ortograficzną tekstu – warto korzystać ze słownika.
- Prezentację warto urozmaicać odpowiednio dobranymi ilustracjami.
- Nie należy przesadzać z liczbą ozdobników.
- Warto dopracować szczegóły:
  - Przenieść spójniki pozostające na końcach wierszy do kolejnego wiersza. W tym celu stosujemy ręczny podział wiersza (**Shift+Enter**) – w programie PowerPoint nie ma możliwości dodawania spacji nierozdzielającej.
  - Wyrównać na slajdach pola tekstowe i obrazy. Najlepiej włączyć linie siatki (**Widok/Linie siatki**), a następnie wybrać sposób wyrównywania na karcie **Formatowanie** (w grupach **Rozmieszczanie** i **Rozmiar**) lub w menu kontekstowym zaznaczonego obiektu (polecenie **Rozmiar i położenie**). Aby precyzyjnie umieścić pole tekstowe lub obraz, można je zaznaczyć i użyć klawiszy strzałek. Największą precyzję uzyskamy, wykorzystując kombinację klawiszy **Ctrl+strzałka**.

Zasady dodawania animacji

Do slajdów można dodać animacje, korzystając z opcji na karcie **Animacje**. Możemy animować poszczególne elementy slajdu: tekst, grafikę, wykresy, schematy i inne obiekty. Wyróżniony w ten sposób element slajdu przyciąga uwagę i wzbudza zainteresowanie odbiorcy.

- Należy umiejętnie stosować animacje – tempo ich wyświetlania nie powinno być zbyt wolne ani zbyt szybkie. Nieodpowiednie tempo i nadmierna różnorodność sprawiają wrażenie chaosu. Animacje nie muszą być przypisywane każdemu elementowi slajdu. Najlepiej korzystać z nich jedynie w celu podkreślania ważnych informacji.
- Jeśli nie odpowiada nam wybrana animacja, możemy ją zmienić. Można również zmienić sposób animowania elementu. W tym celu należy w **Okienku animacji** kliknąć prawym przyciskiem myszy nazwę na liście efektów i wybrać polecenie **Chronometraż**. Możemy m.in. zmienić: sposób rozpoczęcia animacji (animacja pojawi się po kliknięciu myszą, po poprzedniej animacji lub równocześnie z poprzednią), szybkość przebiegu animacji, kierunek pojawiania się elementu, (np. przylatuje z góry ekranu, wsuwa się od lewej strony).

Dodawanie przejść slajdów

Kolejne slajdy prezentacji mogą się pojawiać po kliknięciu myszą lub automatycznie, po upływie ustalonego czasu. Niezależnie od tego można ustalić sposób pojawiania się każdego slajdu na ekranie. Polecenia zmiany przejścia slajdu znajdziemy na karcie **Przejścia** lub po wybraniu opcji **Pokaz slajdów/Przejście slajdu** (zależnie od wersji programu).

Sposób przełączania slajdu wybieramy w zależności od sposobu wyświetlania prezentacji. Jeśli chcemy sami decydować, kiedy ma się wyświetlić następny slajd, wybieramy przełączanie po kliknięciu myszą; jeśli nie – ustalamy czas, po jakim nastąpi automatyczna zmiana slajdu. Można dodatkowo wybrać dźwięk, który będzie towarzyszył przejściu slajdu.

Wybrane ustawienia przejścia slajdu możemy zastosować do wszystkich slajdów prezentacji (przycisk **Zastosuj do wszystkich**). Jeśli tego nie zrobimy, przejście zostanie zastosowane tylko do aktualnego slajdu.



Tempo wyświetlania slajdów powinno być takie, aby odbiorca mógł zapoznać się z ich treścią i jednocześnie nie musiał zbyt długo czekać na pojawienie się kolejnego slajdu.



Aby zmienić kolejność wybranych efektów animacji, należy wskazać nazwę efektu w **Okienku animacji** i, trzymając wciśnięty lewy przycisk myszy, **przeciągnąć i upuścić** ją w innym miejscu na liście.

Dodawanie hiperłączy i przycisków akcji



Aby wstawić hiperłącze, należy zaznaczyć obiekt lub fragment tekstu, kliknąć go prawym przyciskiem myszy i wybrać opcję **Hiperłącze**. Następnie należy określić obiekt docelowy, np. slajd albo adres URL.



Aby wstawić przycisk akcji, należy:

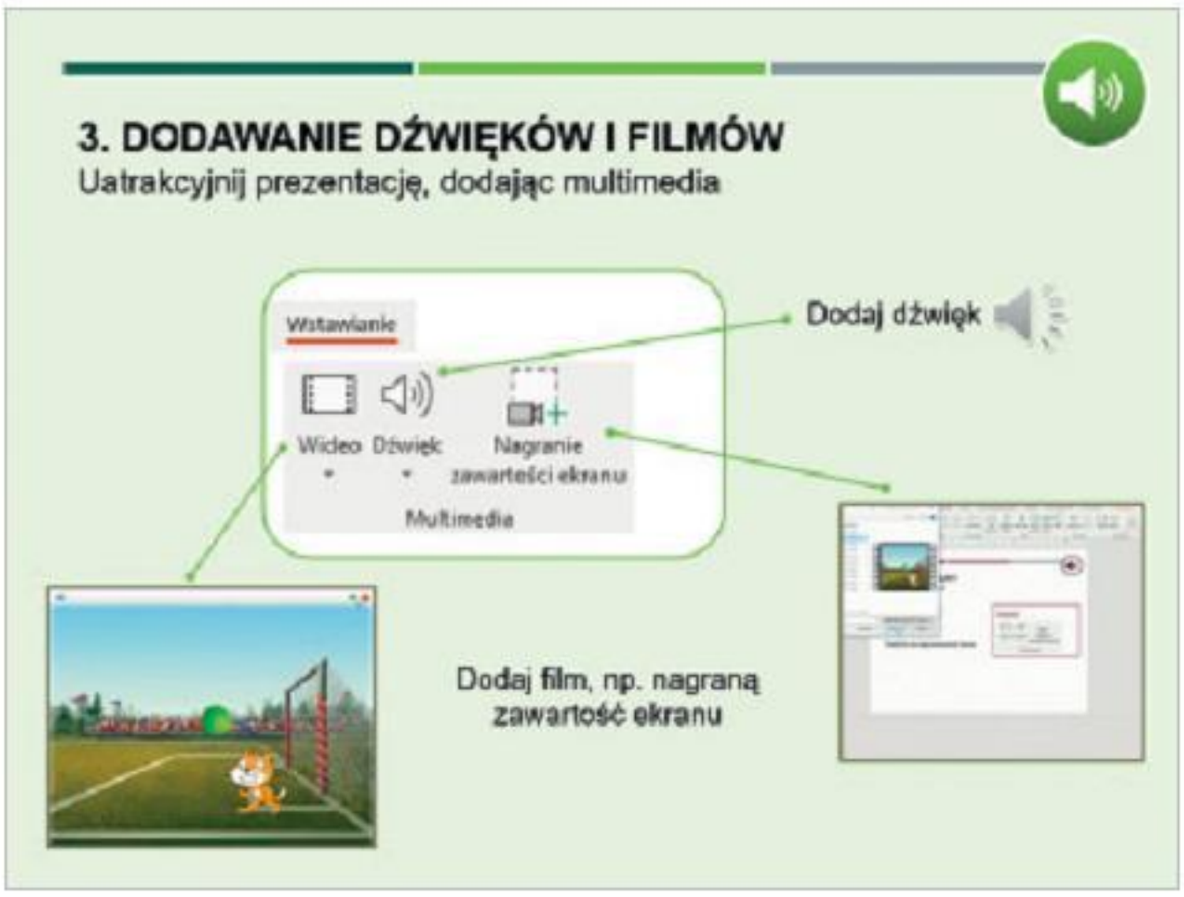
- wybrać opcję **Pokaz slajdów/Przyciski akcji** (lub na karcie **Wstawianie**, w grupie **Ilustracje** kliknąć strzałkę pod przyciskiem **Kształty**), a następnie – przycisk, który ma zostać dodany,
- kliknąć wybrane miejsce na slajdzie i przeciągnąć wskaźnik myszy, aby utworzyć przycisk,
- w oknie dialogowym **Ustawienia akcji** określić, jaką akcję programu spowoduje kliknięcie przycisku lub wskazanie go kursorem myszy.

2.3. Dodawanie dźwięków i filmów

W celu podniesienia walorów poznawczych prezentacji lub jej uatrakcyjnienia można do niej dodać dźwięki (muzykę, efekty dźwiękowe) i filmy. Narracja to komentarz do prezentacji, nagrany przed jej uruchomieniem lub podczas pokazu (można wtedy dołączyć wypowiedzi odbiorców).

Na każdym slajdzie, do którego dodano dźwięk, pojawia się ikona dźwiękowa. Dźwięk może być odtwarzany automatycznie lub po kliknięciu tej ikony. Pliki dźwiękowe można **połączyć** z prezentacją lub w niej **osadzić**.

W przypadku połączenia zachowuje się trwałe połączenie między obiektem wstawionym a obiektem źródłowym i jego aplikacją macierzystą. Każdą zmianę danych w pliku źródłowym widać we wstawionym obiekcie. Plik dźwiękowy jest przechowywany poza prezentacją, więc rozmiar pliku prezentacji jest mniejszy.



Rys. 6. *Slajd „Dodawanie dźwięków i filmów”*

Osadzać można tylko pliki dźwiękowe w formacie WAVE – pozostałe typy plików multimedialnych są z prezentacją łączone. Osadzenie pliku dźwiękowego powoduje zwiększenie rozmiaru pliku prezentacji (plik ten staje się częścią pliku z prezentacją).

Do prezentacji można dodać film, który wzbogaci pokaz o dodatkowe informacje, np.: instruktaż wykonywania czynności (montaż lub demontaż części rowerowych), sceny ilustrujące tezy przedstawiane przez autora prezentacji, wypowiedzi osób nieobecnych podczas przedstawiania prezentacji itp.

Przed dodaniem do prezentacji plików (dźwiękowych, filmowych) najlepiej skopiować je do tego samego folderu, w którym zapisano prezentację.

Aby dodać do prezentacji film, należy:

- na karcie **Wstawianie** lub w menu **Wstaw** wybrać opcję wstawiania filmu (np. **Film**, **Wideo**),
- wybrać źródło (np. z pliku, z kolekcji filmów).

Film może być wyświetlany podczas prezentacji automatycznie, po kliknięciu myszą lub po czasie określonym w programie. Film może też być odtwarzany na kilku slajdach lub przez cały czas trwania prezentacji.

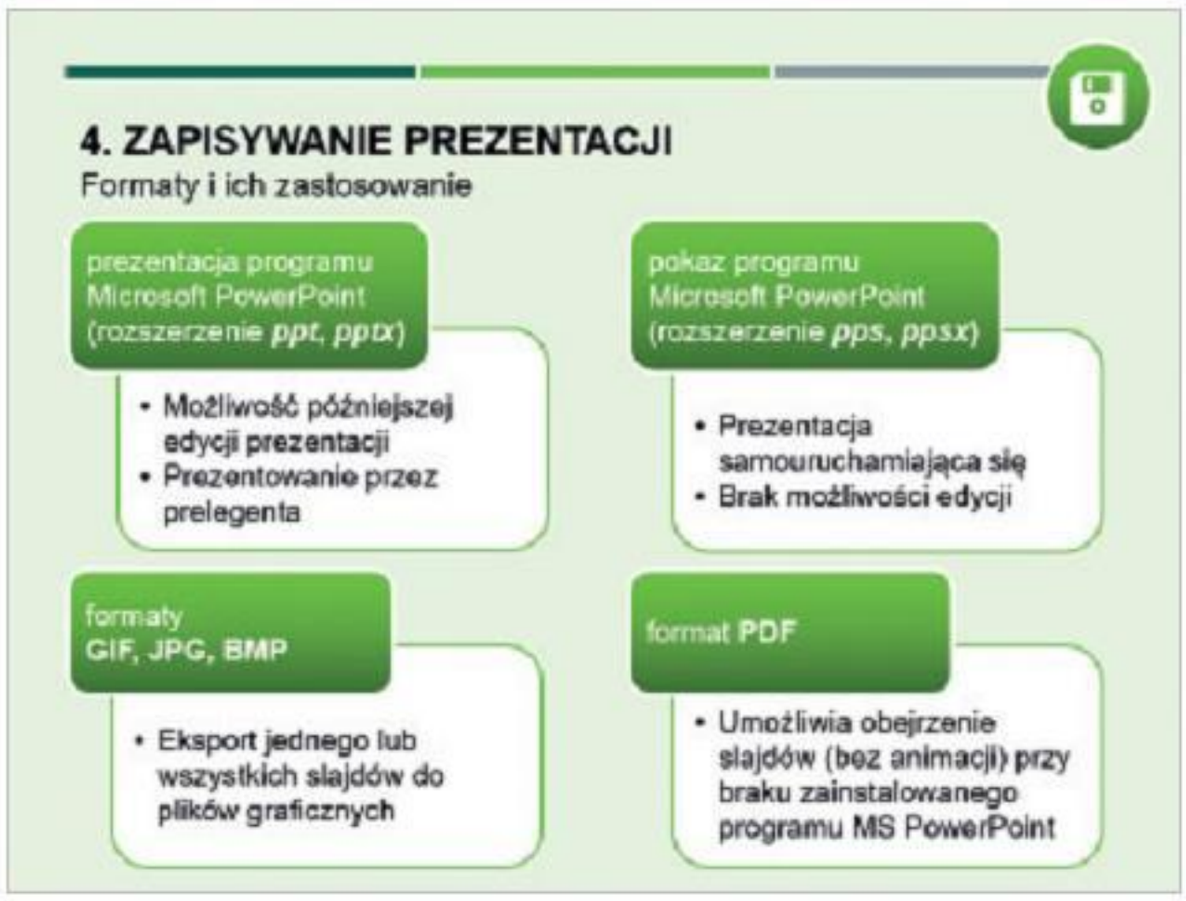
Po dodaniu filmu i wybraniu opcji automatycznego odtwarzania w **Okienku animacji** pojawiają się dodatkowo dwa efekty: wstrzymania i odtwarzania. Efekt wstrzymania pozwala na zatrzymanie filmu kliknięciem i kontynuowanie odtwarzania po ponownym kliknięciu. Jeśli nie skorzystamy z efektu wstrzymania, każde kliknięcie filmu będzie powodowało rozpoczęcie odtwarzania od początku. Film można odtwarzać tak, aby wypełniał część slajdu lub cały ekran (tryb pełnoekranowy).

Przydatną funkcją programu PowerPoint jest nagranie zawartości ekranu. Możemy na przykład nagrać ciąg czynności pokazujących, jak wykonać daną operację w programie Scratch. Nagrają się ruchy wskaźnika myszy, kliknięcia, wybór opcji itp. W ten sposób możemy również utworzyć nagranie z uruchomienia i wykonania programu np. w języku C++ lub Python, a następnie wstawić je na slajd prezentacji dotyczącej programowania.

Aby wstawić na slajd prezentacji nagranie zawartości ekranu, należy:

- otworzyć plik, np. z programem w języku Scratch,
- nie zamykając okna programu Scratch, przejść do okna programu PowerPoint i uaktywnić slajd, do którego chcemy dodać film,
- na karcie **Wstawianie** z grupy **Multimedia** wybrać opcję **Nagranie zawartości ekranu**,
- w menu wyświetlonym na ekranie kliknąć przycisk **Zaznacz obszar** i zaznaczyć fragment ekranu (np. okno z uruchomionym programem), następnie kliknąć przycisk **Nagraj**,
- aby zakończyć nagrywanie, należy nacisnąć kombinację klawiszy **Logo systemu Windows** + **Shift** + **Q** – komputer automatycznie umieści zarejestrowany film na aktualnym slajdzie.

2.4. Zapisywanie prezentacji



Rys. 7. *Slajd „Zapisywanie prezentacji”*

Program Microsoft PowerPoint umożliwia wiele sposobów zapisywania prezentacji. Wyboru formatu dokonujemy w oknie **Plik/Zapisz jako**. Najczęściej wykorzystywane formaty:

- **Prezentacja programu Microsoft PowerPoint** (rozszerzenie *ppt*, *pptx*) – zwykła prezentacja. Plik można edytować w programie Microsoft PowerPoint lub innym obsługującym ten format.
- **Pokaz programu Microsoft PowerPoint** (rozszerzenie *pps*, *ppsx*) – po otwarciu prezentacji w systemie Windows (z zainstalowanym programem Microsoft PowerPoint lub Microsoft PowerPoint Viewer) zostanie ona od razu wyświetlona na pełnym ekranie, a po jej zakończeniu program Microsoft PowerPoint zostanie automatycznie zamknięty. Pliku tego typu nie można edytować.

- **Konspekt/RTF** – format przydatny szczególnie na etapie projektowania prezentacji. Konspekt można również wykorzystać w czasie prelekcji.
- **Format PDF** – umożliwia obejrzenie slajdów prezentacji (bez animacji) osobom, które na swoim komputerze nie mają zainstalowanego programu Microsoft PowerPoint. Prezentację w formacie PDF można np. umieścić na stronie internetowej.
- **Formaty GIF, JPEG, BMP** – umożliwiają zapisanie jednego lub wszystkich slajdów jako plików graficznych w jednym z popularnych formatów.

Prezentację utworzoną w programie Microsoft PowerPoint możemy wyeksportować do dokumentu tekstowego programu Microsoft Word przez wybranie w menu **Plik** opcji **Eksportuj/Utwórz materiały informacyjne**.

Uwaga

Jeżeli wykorzystywany przez nas program nie obsługuje eksportu do formatu PDF, dokument PDF można utworzyć, korzystając z jednego z wielu darmowych programów, np. PDFCreator, które tworzą w systemie Windows wirtualną drukarkę. Wysłanie zlecenia drukowania na takiej drukarce (z dowolnego programu, także z przeglądarki internetowej) powoduje utworzenie dokumentu PDF.



Aby możliwe było odtworzenie prezentacji na każdym komputerze, nawet takim, na którym nie zainstalowano programu Microsoft PowerPoint lub Microsoft PowerPoint Viewer, można w menu **Plik** wybrać opcję **Eksportuj/Spakuj prezentację na dysk CD** lub **Spakuj na dysk CD**.

Wybranie opcji spakowania prezentacji na dysk CD powoduje skopiowanie prezentacji i wszystkich używanych w niej czcionek, połączonych plików dźwiękowych, sekwencji wideo itp. do jednej lokalizacji (na nośnik lub do folderu) i automatyczne zaktualizowanie w pliku z prezentacją wszystkich łączy do plików zewnętrznych.



Aby udostępnić prezentację multimedialną w Internecie, wykorzystując chmurę, należy w menu **Plik** wybrać opcję **Zapisz jako/OneDrive** lub **Udostępnij** (należy mieć utworzone konto w usłudze **OneDrive**).

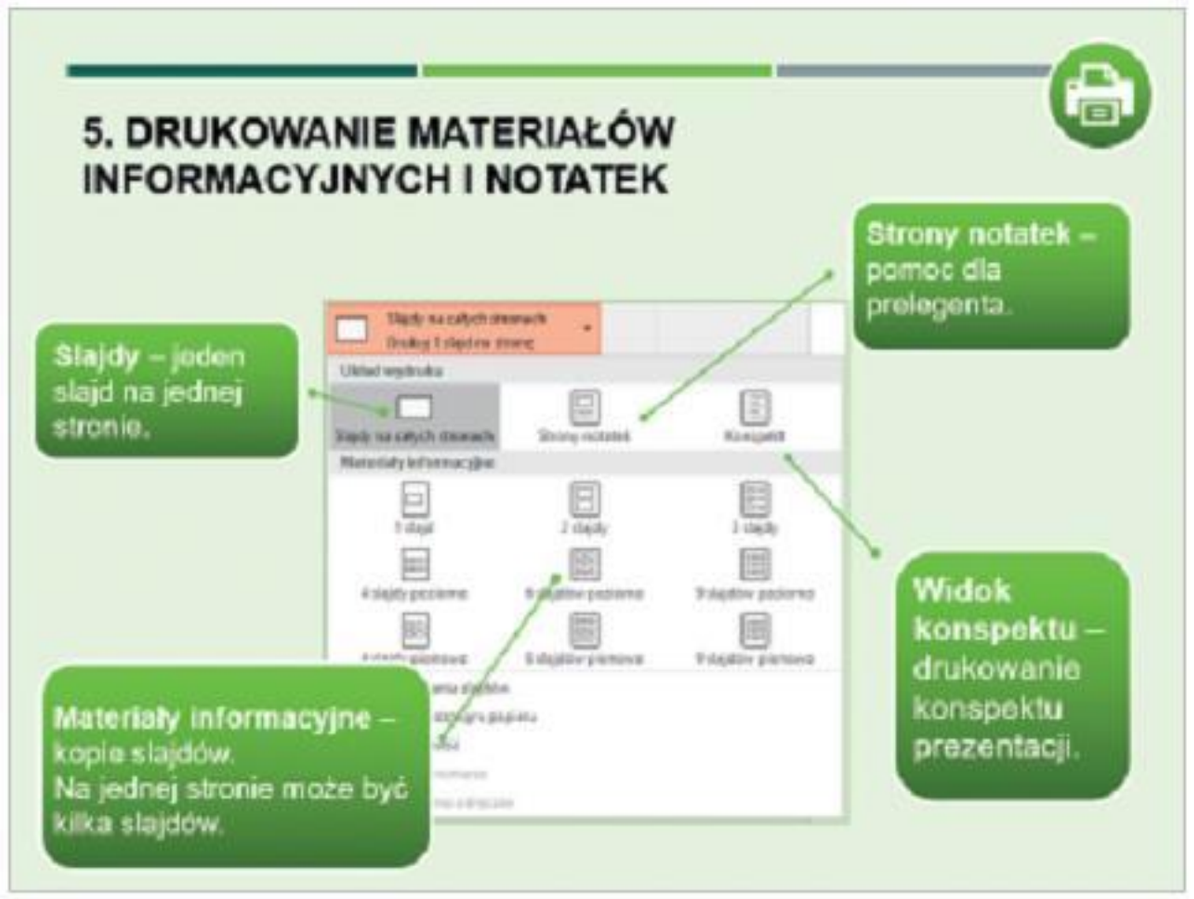
Korzystając z opcji **Zapisz jako**, można również zapisać prezentację w innej lokalizacji. W tym celu należy wybrać polecenie **Dodaj miejsce** lub **Inne lokalizacje w sieci Web** (zależnie od wersji programu).

W starszych wersjach programu (np. Microsoft PowerPoint 2003), aby zapisać prezentację jako stronę sieci Web i utworzyć folder, w którym będą przechowywane pliki pomocnicze (np. tła, obrazy i dźwięki), należy wybrać typ pliku **Strona sieci Web**. Zapisanie prezentacji jako strony sieci Web powoduje przekształcenie tekstu umieszczonego na karcie **Konspekt** w spis treści, który umożliwia poruszanie się między slajdami.

2.5. Drukowanie materiałów informacyjnych i notatek

Uwaga

Bądź ekologiczny, oszczędzaj papier – nie drukuj materiałów, jeżeli nie będą ci niezbędne.



Rys. 8. *Slajd „Drukowanie materiałów informacyjnych i notatek”*

W programie Microsoft PowerPoint możliwe jest drukowanie prezentacji w formie materiałów, takich jak:

- **Slajdy** – dzięki tej funkcji można wydrukować slajdy, np. na przezroczystej folii lub na papierze. Slajdy drukowane są w takiej postaci, w jakiej pojawiają się na ekranie – jeden slajd na jednej stronie.
- **Materiały informacyjne** – kopie slajdów. Na jednej stronie wydruku może znajdować się kilka slajdów. Wygląd materiałów można modyfikować, wybierając na karcie **Widok** polecenie **Wzorzec materiałów informacyjnych** lub polecenie **Widok/Wzorzec/Wzorzec materiałów informacyjnych**. Można np. dodać własny nagłówek i stopkę oraz zmienić tło.
- **Strony notatek** – na jednej stronie można wydrukować pomniejszony slajd wraz z notatkami do niego. Tego rodzaju materiały mogą być pomocą dla prelegenta lub zawierać dodatkowe informacje dla odbiorców.
- **Widok konspektu** – drukowanie konspektu prezentacji (w postaci, w jakiej widoczny jest na ekranie).

Warto też zwrócić uwagę na opcje drukowania dostępne w programie Microsoft PowerPoint, np. **Dołącz animacje**, dzięki której można wydrukować kilka wersji slajdu, uzyskując efekt symulacji animacji dodanych do slajdu.

W widoku **Normalnym** pod wyświetlanym slajdem znajduje się pole na notatki. Notatki mogą być dodatkowymi informacjami dla prelegenta (na które kwestie zwrócić uwagę, co dodatkowo uzupełnić itd.). Notatki można wydrukować razem z miniaturkami slajdów i udostępnić tak przygotowane materiały odbiorcom. Mogą zastąpić narrację dla osób niedosłyszących oraz dla użytkowników, których komputerów nie wyposażono w kartę dźwiękową. Po zapisaniu prezentacji w formacie HTML notatki będą wyświetlane razem ze slajdami.

2.6. Ustalenie parametrów pokazu i pokaz



Rys. 9. *Slajd „Ustalenie parametrów pokazu i pokaz"*

Parametry pokazu dostępne są po wybraniu karty **Pokaz slajdów**.

Po wybraniu opcji **Przygotuj pokaz slajdów** otworzy się okno **Przygotowanie pokazu**. W zależności od wybranego typu pokazu dostępne będą różne opcje, m.in.: przełączania slajdów (ręczne czy zgodnie z chronometrażem), określenia, które slajdy mają być pokazywane. Jeśli nie chcemy, aby podczas pokazu wyświetlały się animacje bądź było słychać narrację, możemy je w tym oknie wyłączyć.

Dzięki opcji **Próba tempa** możemy wypróbować tempo zmieniania się slajdów. Program rejestruje czas potrzebny na omówienie danego slajdu. W ten sposób możemy precyzyjnie określić czas wyświetlania każdego slajdu.

Opcja **Rejestruj pokaz slajdów** umożliwia nagranie pokazu slajdu łącznie z **narracją**. Narracja odtwarzana jest podczas wyświetlania slajdów. Może ona w pewnym stopniu zastąpić prelegenta – przydaje się, gdy odbiorca będzie przeglądał prezentację samodzielnie. Najlepiej stosować narrację do prezentacji typu kiosk. Narracja może zostać nagrana do wybranych slajdów lub do całej prezentacji. Do nagrania narracji potrzebne są karta dźwiękowa i mikrofon. Należy też pamiętać o głośnikach do odtwarzania dźwięku.

Podczas pokazu można na slajdach rysować linie lub inne znaki, podkreślając najważniejsze informacje. Notatki tego typu możemy robić za pomocą **Pióra**.



Aby włączyć Pióro, należy uruchomić prezentację, kliknąć ekran prawym przyciskiem myszy i z menu kontekstowego wybrać **Opcje wskaźnika** (lub nacisnąć klawisze **Ctrl+P**).

Gdy **Pióro** jest aktywne, kliknięcie myszą nie powoduje przejścia do następnego slajdu. Aby przejście było możliwe, należy nacisnąć klawisz **Enter** lub klawisz sterujący ↓. W **Opcjach wskaźnika** znajdują się również polecenia zmiany koloru pióra i **Gumka**.

O czym należy pamiętać przy wygłaszaniu prezentacji?

- Przed pokazem dobrze jest przejrzeć konspekt prezentacji i notatki.
- Ważne jest pierwsze wrażenie, dlatego należy dopracować wstęp prezentacji. Ma on za zadanie przygotować odbiorców do obejrzenia pokazu. Wstęp nie powinien być zbyt długi.
- Kluczową sprawą jest utrzymanie kontaktu z odbiorcami. Jest to niezbędne, aby zaciekawić ich swoim pokazem.
- Trzeba dobrać tempo wyświetlania pokazu do percepcji odbiorców.
- Nie warto czytać tekstu umieszczonego na slajdach. Każdy ma go przecież przed oczami. Powinno się go raczej komentować i uzupełniać, aby widzowie mogli lepiej zrozumieć prezentowaną tematykę.
- Jeśli pokaz nie jest zbyt długi ani skomplikowany, najlepiej poprosić słuchaczy, aby zadawali pytania dopiero po jego zakończeniu.
- Pod koniec prezentacji zawsze warto jeszcze raz powtórzyć jej najważniejsze punkty.

Co można wręczać widzom?

Jako dodatek do prezentacji można wręczyć widzom materiały drukowane. Mogą to być ulotki z podsumowaniem najważniejszych punktów pokazu albo np. foldery reklamowe. Dobrze przygotowane materiały będą przypominać treść pokazu.

Na przykład w przypadku prezentacji z geografii na temat Morza Martwego słuchaczom przyda się wydrukowany konspekt prezentacji. Można im go wręczyć jeszcze przed rozpoczęciem pokazu, aby mogli robić notatki uzupełniające, korzystając z informacji usłyszanych podczas prelekcji. Jeśli pokaz zawiera tabele czy wykresy, również warto je wydrukować. Takie podejście jest szczególnie ważne, jeśli prezentacja zawiera wiele liczb (np. danych statystycznych). Dzięki konspektowi słuchacze nie będą musieli gorączkowo notować niezbędnych informacji.

Ćwiczenie 3. Omawiamy zasady tworzenia prezentacji

Jesteś prelegentem, który ma przedstawić zasady tworzenia prezentacji, korzystając ze slajdów na rysunkach 3a, 4-9. Przygotuj komentarz prelegenta do każdego ze slajdów, korzystając z treści punktów 2., 2.1-2.6.

3. Tworzymy prezentację multimedialną o Fryderyku Chopinie

W kolejnych ćwiczeniach przygotujemy prezentację na temat pobytu Fryderyka Chopina w miejscowości Valldemossa na Majorce. Wykonamy ją zgodnie z omówionymi etapami.
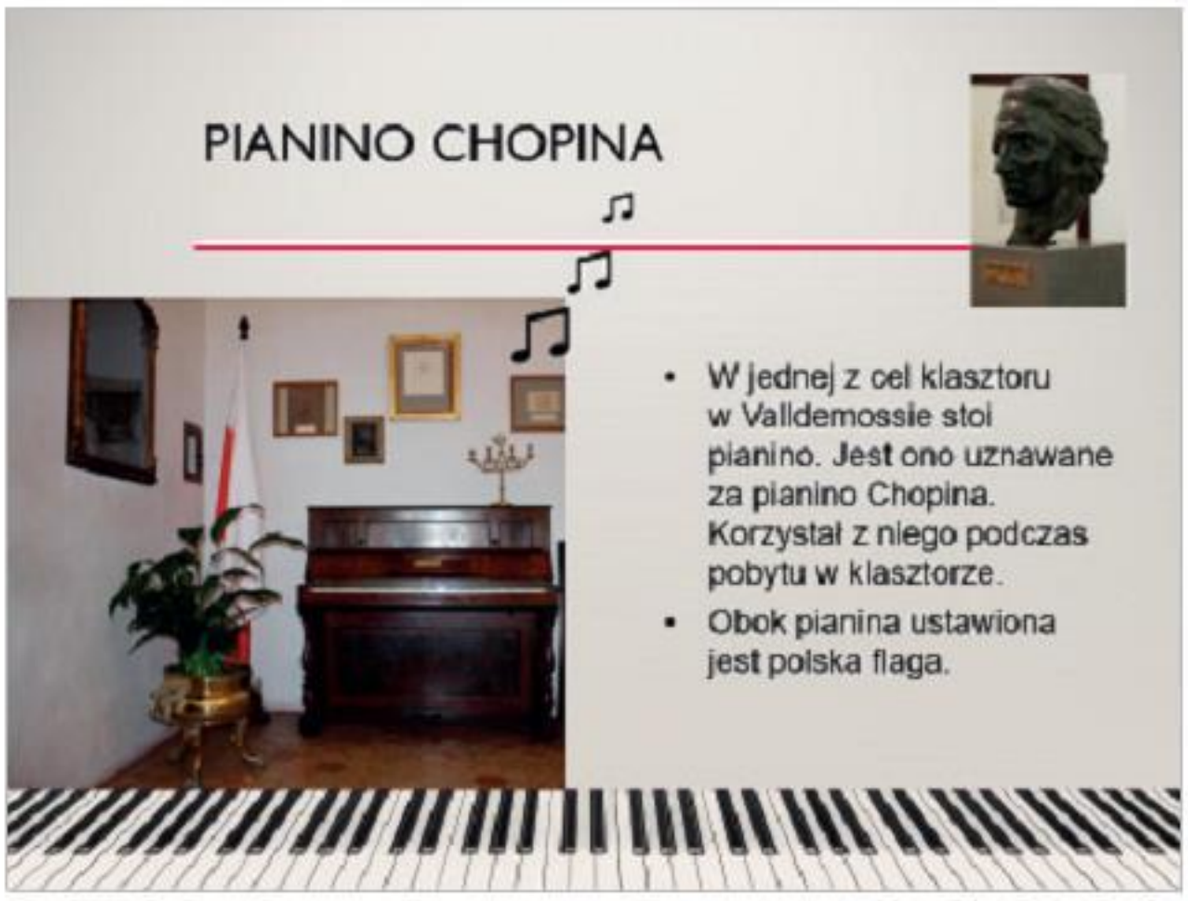


Rys. 10. *Przykładowy slajd prezentacji o Chopinie – spis treści*

Ćwiczenie 4. Planujemy prezentację

Przygotuj plan prezentacji na temat pobytu Fryderyka Chopina w miejscowości Valldemossa na Majorce. Prezentacja ma za zadanie wspomagać wystąpienie prelegenta.

1. W prezentacji należy ująć krótkie informacje dotyczące Fryderyka Chopina, George Sand, Majorki, Valldemossy i klasztoru, w którym mieszkał Chopin. Poszukaj informacji na ten temat. Skorzystaj z Internetu lub innych źródeł.
2. Zastanów się, jakie informacje będą najistotniejsze dla odbiorców prezentacji. Zaplanuj wstępnie zawartość poszczególnych slajdów.
3. Zastanów się, jakie pomoce wizualne zastosujesz (m.in. obrazy, dźwięki). Znajdź odpowiednie materiały w Internecie.
4. Ustal wstępnie liczbę slajdów.



Rys. 11. *Przykładowy slajd prezentacji o Chopinie*

! Uwaga

Mimo że o zapisywaniu prezentacji piszemy dopiero w etapie 4., to tworzoną prezentację (w formacie roboczym – *ppt*, *pptx*) powinno się zapisywać co jakiś czas – tak jak podczas pracy nad każdym dokumentem komputerowym.

Ćwiczenie 5. Tworzymy prezentację

1. Wybierz motyw lub zmodyfikuj wybrany motyw (szablon projektu) według własnego pomysłu i zastosuj go do slajdów prezentacji.
2. Po slajdzie tytułowym dodaj nowy slajd i umieść na nim spis treści prezentacji.
3. Dodaj kolejne slajdy zgodnie ze spisem treści – wstaw teksty, zdjęcia.
4. Wykorzystaj grafikę SmartArt do przedstawienia chronologii zdarzeń, korzystając np. ze schematu pokazanego na rysunku 12. Pod konkretnymi datami możesz wpisać najważniejsze wydarzenia z życia Chopina i wyróżnić innym kolorem okres przedstawiany w prezentacji.
5. Do każdej pozycji w spisie treści dodaj przycisk akcji pozwalający przechodzić do odpowiedniego slajdu. Na pozostałych slajdach dodaj przyciski akcji pozwalające przechodzić do następnego/poprzedniego slajdu oraz do slajdu ze spisem treści.
6. Dodaj animacje do wybranych obiektów na slajdach.
7. Dodaj efekty przejścia między slajdami.



Rys. 12. *Wykorzystanie grafiki SmartArt do przedstawienia chronologii zdarzeń*

Ćwiczenie 6. Dodajemy dźwięki i filmy

1. Wyszukaj w Internecie preludium *Des-dur Deszczowe*, skomponowane przez Fryderyka Chopina na Majorce. Wstaw na wybranym slajdzie hiperłącze do strony internetowej z nagraniem utworu.
2. Znajdź w Internecie krótki film z koncertu chopinowskiego i dodaj go do prezentacji.

Ćwiczenie 7. Zapisujemy prezentację

Zapisz prezentację w pliku pod nazwą *Chopin* z rozszerzeniem *pptx*.

Ćwiczenie 8. Drukujemy materiały informacyjne

1. Otwórz prezentację *Chopin* zapisaną w ćwiczeniu 7. Wyobraź sobie, że masz ją przedstawić przed dużym gronem osób. Opisz poszczególne slajdy za pomocą notatek, tak aby po wydrukowaniu uzyskać materiały, które pomogą w przygotowaniu prelekcji.
2. Zapisz plik pod tą samą nazwą.

Ćwiczenie 9. Ustawiamy parametry pokazu i przeprowadzamy pokaz

1. Ustaw wybrane opcje pokazu i wypróbuj tempo przejścia slajdów.
2. Zapisz plik pod tą samą nazwą.
3. Dodatkowo zapisz prezentację w formatach *ppsx* i *pdf*.

4. Importowanie konspektu prezentacji z dokumentu tekstowego

Konspekt można zaimportować z dokumentu tekstowego zapisanego z rozszerzeniem *rtf*, *doc* lub *docx*. Dokument powinien być sformatowany z wykorzystaniem stylów nagłówkowych (temat B1), np.: stylem **Nagłówek1** tytuły (punkty), stylem **Nagłówek2** – treść (podpunkty).

Aby zaimportować konspekt z dokumentu tekstowego zapisanego w pliku, należy:

- otworzyć nową prezentację multimedialną,
- wybrać opcję **Otwórz** i w oknie otwierania pliku wybrać typ pliku **Wszystkie pliki** (lub **Wszystkie konspekty**),
- odnaleźć i otworzyć dokument zawierający konspekt, na podstawie którego chcemy utworzyć prezentację.

Jeżeli chcemy wstawić konspekt do utworzonej wcześniej prezentacji, należy wybrać polecenie **Nowy slajd/Slajdy z konspektu** (lub **Wstaw/Slajdy z konspektu**). Dokument tekstowy nie powinien być równolegle otwarty w edytorze tekstu.

Ćwiczenie 10. Importujemy konspekt prezentacji

1. Otwórz plik *Komputery* zapisany w ćwiczeniu 13. z tematu B1. Punkty powinny być sformatowane stylem *Nagłówek 1*, a podpunkty – stylem *Nagłówek 2*. Zapisz plik pod nazwą *Konspekt*. Zamknij okno programu Microsoft Word.
2. Otwórz nową prezentację multimedialną i wstaw do niej konspekt z pliku *Konspekt*.
3. Zapisz prezentację w pliku pod nazwą *Komputery*.

Warto zapamiętać

- Zależnie od sposobu przedstawiania można przygotować prezentację wspomagającą wystąpienie prelegenta, do samodzielnego przeglądania lub prezentację typu kiosk. W każdym przypadku należy stosować ogólne zasady przygotowywania prezentacji.
- W przygotowaniu prezentacji wyróżniliśmy następujące etapy: planowanie, tworzenie (w tym m.in.: dodawanie tekstów, obrazów, animacji, przejść slajdów), dodawanie dźwięków i filmów, zapisywanie, drukowanie materiałów informacyjnych, ustawianie parametrów (w tym dodanie narracji) i przeprowadzanie pokazu.
- Nie należy wypełniać całego slajdu samym tekstem. Lepiej przedstawić treści z wykorzystaniem schematów, rysunków, ikon, a komentarz dopowiedzieć, zwłaszcza w prezentacji wspomagającej wystąpienie prelegenta. W przypadku prezentacji do samodzielnego przeglądania lub typu kiosk można dodać narrację.
- Do prezentacji multimedialnej można zaimportować konspekt dokumentu tekstowego – punkty sformatowane stylem *Nagłówek 1* zostaną umieszczone na kolejnych slajdach jako tytuły slajdów, a tekst sformatowany stylem *Nagłówek 2* – jako treść slajdu.

Pytania i polecenia

1. Wyjaśnij, kiedy stosujemy poszczególne typy prezentacji.
2. Na jakie etapy możemy podzielić przygotowywanie prezentacji?
3. Jakie czynności należy wykonać podczas planowania prezentacji?
4. O czym należy pamiętać, umieszczając tekst na slajdach?
5. Na co powinniśmy zwracać uwagę podczas dodawania animacji?
6. Do czego mogą służyć hiperłącza i przyciski akcji umieszczane na slajdach?
7. Wyjaśnij sposób dodawania dźwięków i filmów na slajd prezentacji.
8. Jak można wstawić na slajd nagranie zawartości ekranu?
9. W jakich formatach można zapisać prezentację?
10. Jakie materiały można wręczyć widzom? Dlaczego warto to zrobić?
11. Jakie parametry pokazu możemy ustalić?
12. Omów najważniejsze zasady przedstawiania prezentacji.
13. W jaki sposób zaimportować konspekt z dokumentu tekstowego do prezentacji?

Zadania

1. Zaproponuj sposób rozmieszczenia informacji z punktu 1. tego tematu na jednym slajdzie. Odbiorcami materiału będą twoi koledzy i koleżanki z klasy, a prezentacja ma wspomagać twoje wystąpienie. Zapisz prezentację w pliku pod nazwą *Typy prezentacji*.
2. Utwórz własną prezentację o tworzeniu prezentacji. Możesz się wzorować na wskazówkach z punktu 2. tego tematu, ale na co najmniej dwóch slajdach zastosuj inne pomysły na przedstawienie treści (np. z wykorzystaniem innej grafiki). Pamiętaj o przestrzeganiu zasad tworzenia prezentacji. Zapisz plik pod nazwą *Etapy*.
3. Utwórz nowy dokument tekstowy. Opisz w nim cztery miejsca, do których chcesz pojechać w przyszłości (po trzy zdania). Zastosuj style nagłówkowe do tytułów (np. nazwa miejsca) i treści (**Nagłówek1** i **Nagłówek2**), aby można było zaimportować plik jako konspekt do programu PowerPoint. Zapisz plik pod nazwą *Moja podróż*.
4. Otwórz nowy dokument w programie PowerPoint i zaimportuj przygotowany konspekt *Moja podróż* zapisany w zadaniu 3. Dodaj zdjęcia wybranych miejsc. Znajdź je w Internecie (pamiętaj o podaniu źródła).
5. Korzystając z opisanych w tym temacie zasad, zaprojektuj i utwórz prezentację multimedialną na temat znanej osoby ze świata kultury, która według ciebie ma szczególne osiągnięcia. Materiały znajdź w Internecie. Zacznij od utworzenia planu prezentacji. Zapisz prezentację w pliku pod nazwą *Kultura*.

Dla zainteresowanych

6. Odbywasz praktykę lub staż w sklepie firmy produkującej odzież. Jako pierwsze zadanie pracodawca zlecił ci wykonanie prezentacji reklamującej firmę. Prezentacja będzie pokazywana na targach branżowych. Powinna zawierać cztery slajdy (slajd tytułowy, dwie strony z ofertą i podsumowanie). Korzystając z Internetu, sprawdź, jakie istotne informacje o firmie i jej produktach powinno się przedstawiać. W swojej prezentacji umieść fikcyjne dane.
7. Jesteś pracownikiem firmy sprzedającej tablety. Twoim zadaniem jest przygotowanie oferty handlowej, która zostanie wysyłana do potencjalnych odbiorców produktów firmy. Oferta ma być przygotowana w postaci prezentacji do samodzielnego przeglądania przez odbiorcę i zawierać pięć slajdów. Korzystając z Internetu, sprawdź, w jaki sposób przygotowuje się ofertę handlową i jakie istotne informacje o produktach powinno się przedstawiać. W swojej prezentacji umieść fikcyjne dane.

Wokół algorytmiki i programowania

C

Moduł

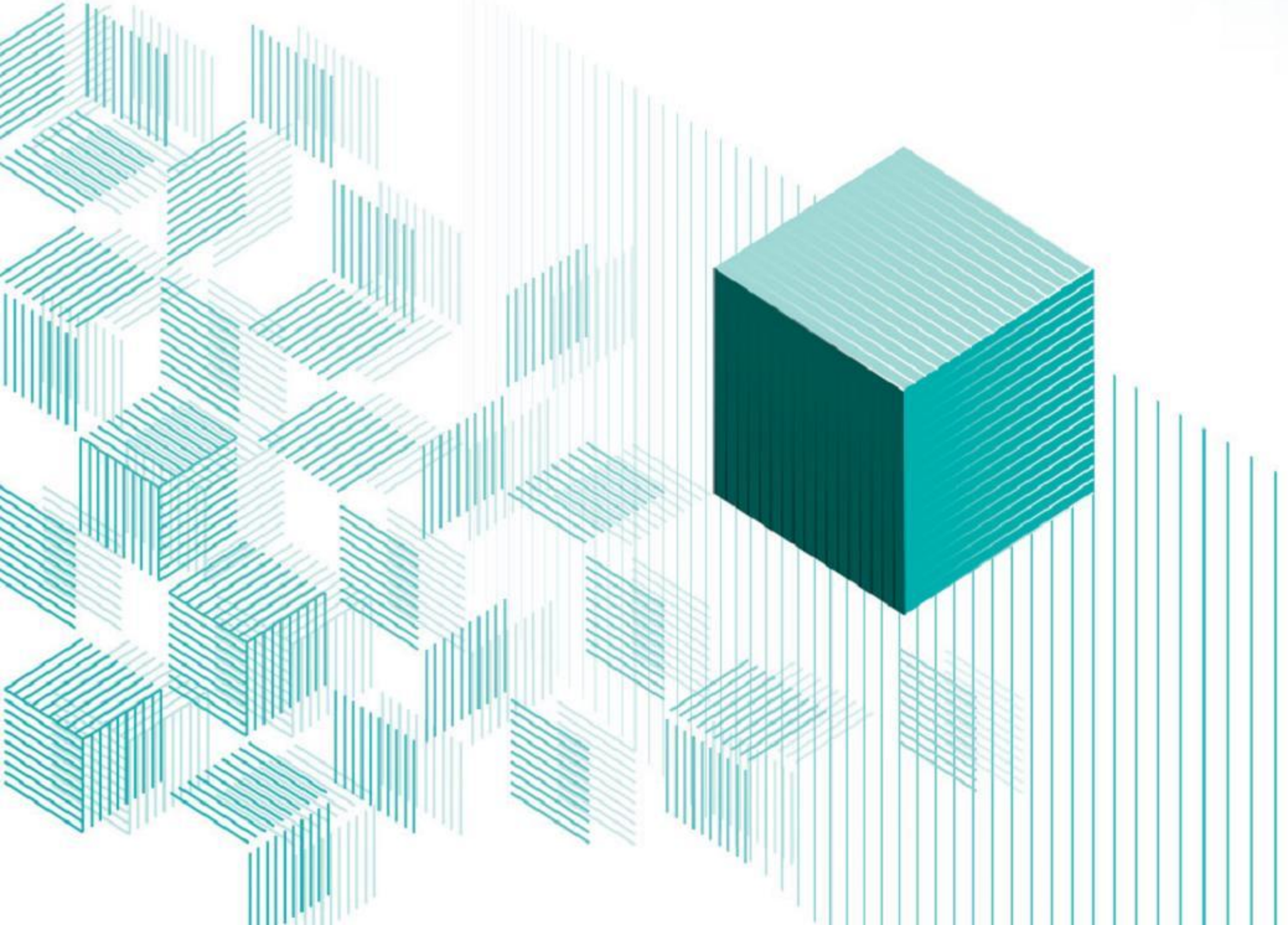
Temat C1. Wprowadzenie do programowania i rozwiązywania problemów z wykorzystaniem komputera

Temat C2. Tworzenie programów w języku C++

Temat C3. Tworzenie programów w języku Python

Temat C4. Stosowanie instrukcji warunkowych w językach C++ i Python

Temat C5. Stosowanie instrukcji iteracyjnych w językach C++ i Python



Temat C1. Wprowadzenie do programowania i rozwiązywania problemów z wykorzystaniem komputera

1. **Sytuacje problemowe**
2. **Określanie specyfikacji zadania (problemu)**
3. **Języki programowania**
4. **Na czym polega programowanie?**

Warto powtórzyć

1. Jakie znasz rodzaje oprogramowania komputerowego?
2. Z jakich programów użytkowych korzystaliśmy? Jakie było ich przeznaczenie?
3. Jakie są zastosowania języków programowania?
4. Z jakich środowisk programowania korzystaliśmy w szkole podstawowej?
5. Na czym polega programowanie w środowiskach poznanych w szkole podstawowej?

1. Sytuacje problemowe

Programy komputerowe umożliwiają rozwiązywanie problemów.

Algorytm A

Uporządkowany i uściślony sposób rozwiązywania danego problemu, zawierający szczegółowy opis wykonywanych czynności w skończonej liczbie kroków.

Program komputerowy (program źródłowy, kod źródłowy) P

Ciąg instrukcji języka programowania, realizujący algorytm.

Gdy mamy do czynienia z **problemem**, szukamy sposobu rozwiązania (**algorytmu**), a następnie dobieramy odpowiednie narzędzie – **program komputerowy** (np. użytkowy lub język programowania).

Najczęściej korzystamy z programów użytkowych (np. arkuszy kalkulacyjnych, edytorów tekstu lub grafiki, programów do odtwarzania muzyki), programów narzędziowych (np. antywirusowych, archiwizujących) i systemu operacyjnego. Nie jest dla nas ważne, w jaki sposób dany program został napisany. Wystarczy, że znamy jego działanie – niezbędne funkcje i możliwości. Zazwyczaj nie mamy w ogóle wglądu do „wnętrza" programu, czyli do tzw. **programu (kodu) źródłowego**.

Najpierw pojawia się problem, dotyczący np. opracowania wyników zawodów sportowych czy ułożenia planu lekcji. Informatycy zapoznają się z problemem i zastanawiają nad sposobem jego rozwiązania – następnie układają lub wybierają gotowy algorytm.

W podręczniku często podajemy przykłady rozwiązywania problemów dotyczących różnych zagadnień, m.in. edycji tekstu (w punkcie 4. z tematu B1 i w punkcie 1. z tematu B2). Problemy i cele do osiągnięcia wyróżniono nawiasami klamrowymi. Po krótkiej analizie problemu opracowujemy rozwiązanie – w podręczniku podane w przykładzie lub wyróżnione ramką z kluczykiem.

Przykład 1. Przykład sytuacji problemowej i algorytmu jej rozwiązania

Jesteś sprzedawcą komputerów. Cena komputera (*cenak*) zależy m.in. od ceny płyty głównej (*cenap*). Jeśli cena płyty wzrośnie o *w*%, to trzeba podnieść cenę komputera. Przygotowując kalkulację ceny, musisz zastanowić się, o jaki procent możesz podnieść cenę, aby nie stracić klienta.

Przyjmiemy następujący algorytm kalkulacji ceny komputera:

a. jeśli *w* mieści się w przedziale ⟨0%; 10%⟩, cena komputera pozostaje bez zmian (zmniejszasz swój zysk, aby zadowolić klienta);
b. jeśli *w* mieści się w przedziale (10%; 20%⟩, cenę sprzedawanego komputera zwiększasz o 5% nowej ceny płyty głównej (przewidujesz, że klient zaakceptuje wzrost ceny komputera);
c. jeśli $w > 20\%$, nie ustalasz ceny tego komputera (szukasz innego producenta płyty głównej) i wyprowadzasz komunikat: *za wysoka cena*.

W życiu codziennym pewne podobieństwo do algorytmów cechuje np. przepisy kulinarne, określające, z jakich składników i w jaki sposób należy przyrządzić potrawę. Przepisów nie można jednak nazwać algorytmami w sensie informatycznym, choćby z racji tego, że pojawiają się w nich określenia nie do końca precyzyjne, np. „gotować przez chwilę". Zależnie od tego, ile ta chwila będzie trwała, możemy otrzymać różne wyniki (np. budyń o różnej konsystencji).

W algorytmach informatycznych dane wejściowe powinny być precyzyjnie określone, a wszystkie operacje – dokładnie opisane. Dla tych samych danych powinniśmy otrzymać te same wyniki. W algorytmach nieinformatycznych wynik może zależeć od danych wejściowych i różnych czynników (np. od czasu gotowania potrawy).

Wymyślony (lub wybrany) algorytm programiści zapisują w postaci możliwej do wykonania przez komputer, czyli w formie programu komputerowego. Korzystają przy tym z wybranego **języka programowania**. Komputer, wykonując program, realizuje dany algorytm. Ten uproszczony schemat postępowania pokazuje rysunek 1.

J **Język programowania**
Specjalny język służący do pisania programów komputerowych. Jest on zbiorem określonych instrukcji i zasad składni.

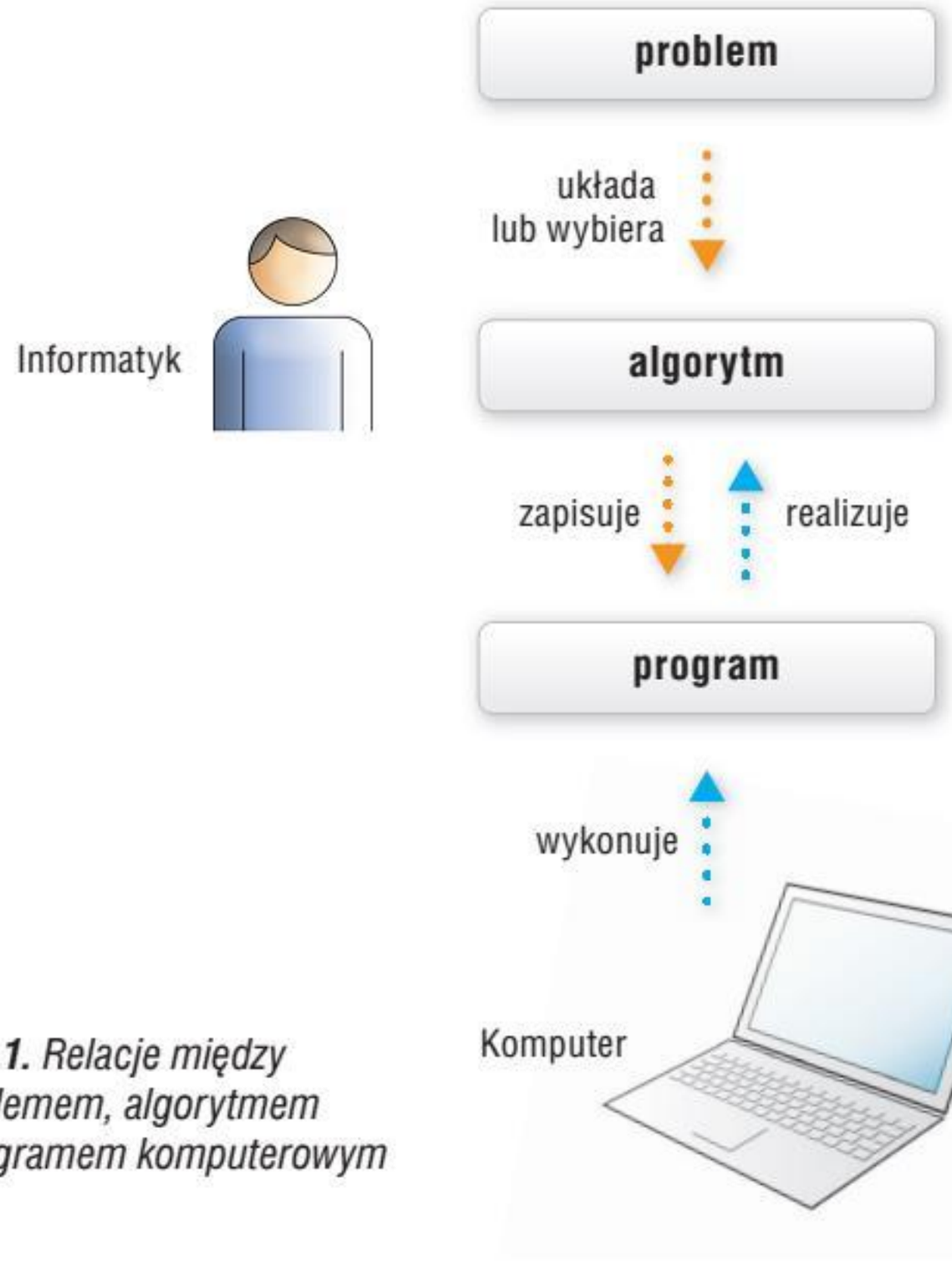
Wybór algorytmu i jego sformułowanie powinny być zawsze podporządkowane problemowi, który ma zostać rozwiązany.

Lista kroków L

Przedstawienie algorytmu w kolejnych punktach (krokach). Każdy punkt takiej listy zawiera opis wykonywanej czynności. Kolejność punktów nie jest przypadkowa – musi być zgodna z działaniem algorytmu.

Schemat blokowy S

Przedstawienie poszczególnych operacji algorytmu za pomocą odpowiednio połączonych figur (bloków). Połączenia określają, w jakiej kolejności i w jaki sposób będą wykonywane operacje realizujące dany algorytm.



***Rys. 1.** Relacje między problemem, algorytmem i programem komputerowym*

W rzeczywistości tworzenie programów jest złożonym procesem – tym bardziej, im trudniejszy jest problem. Bardziej skomplikowane zadania dzieli się na mniejsze części i często dla każdej z nich oddzielnie układa bądź dobiera algorytmy. W efekcie powstają duże projekty informatyczne, w których tworzeniu uczestniczy wiele osób, nie tylko programistów, ale również projektantów.

Ćwiczenie 1. Wybieramy algorytm do rozwiązania problemu

Jaki znany ci algorytm można zastosować do znalezienia największego wspólnego dzielnika (NWD) dla dwóch niezerowych liczb naturalnych?

Przykład 2. Przykład sytuacji problemowej

W komputerze gromadzimy wiele danych. Pojawia się zatem problem ich porządkowania, np. w systemach zarządzania plikami: według dat utworzenia, według nazw lub rozszerzeń; w systemach baz danych: sortowanie artykułów według nazw, cen czy dat dostaw.
Aby użytkownik komputera mógł oglądać dane uporządkowane w wybranym przez siebie układzie, informatycy musieli wymyślić odpowiednie sposoby rozwiązania problemu (algorytmy sortowania). Dopiero potem mogli napisać programy do konkretnych zastosowań.

Program komputerowy jest realizacją wybranego wcześniej algorytmu lub wielu algorytmów.

Istniejące programy są usprawniane – pojawiają się ich kolejne wersje, oferujące nowe możliwości, które pozwalają rozwiązywać coraz więcej problemów. Ciągle powstają też nowe programy, dostosowane do zmieniających się potrzeb użytkowników.

Czasem tworzy się programy dla indywidualnych odbiorców (osób prywatnych, firm), jeśli mają oni wymagania użytkowe, których nie spełnia żaden z istniejących programów. Użytkownik może np. oczekiwać innych od oferowanych w gotowym programie sposobów wprowadzania danych (formularzy), odmiennych końcowych zestawień (raportów) czy dodatkowych wyliczeń. Możemy też wyobrazić sobie, że firma użytkownika to swoisty „magazyn osobliwości", zawierający artykuły, które trudno byłoby klasyfikować i przetwarzać według ogólnych kryteriów.

Nierzadko tworzy się też programy do zadań, które wprawdzie można wykonać za pomocą istniejącego oprogramowania, np. arkusza kalkulacyjnego, ale łatwiej i wygodniej jest napisać odrębny program. W nowym programie zadanie będzie wymagało jedynie kilku prostych instrukcji, podczas gdy w arkuszu kalkulacyjnym byłoby bardziej pracochłonne – wymagałoby większej liczby czynności, np. kopiowania tej samej formuły.

Postępowanie mające na celu rozwiązanie dowolnego problemu możemy podzielić na następujące etapy:

1. Sformułowanie zadania.
2. Określenie danych wejściowych.
3. Określenie celu, czyli wyniku.
4. Określenie metody rozwiązania, czyli wybór algorytmu.
5. Przedstawienie algorytmu w wybranej postaci: opisu słownego, listy kroków, schematu blokowego lub programu w wybranym języku programowania.
6. Analiza poprawności rozwiązania.
7. Testowanie rozwiązania dla różnych danych – ocena efektywności przyjętej metody.

! Uwaga

W kolejnych tematach modułu C algorytmy będziemy opisywać i przedstawiać w postaci programów komputerowych, a na lekcjach realizowanych w zakresie rozszerzonym – w postaci list kroków, schematów blokowych i programów komputerowych.

2. Określanie specyfikacji zadania (problemu)

W zadaniach z różnych dziedzin (np. z fizyki czy matematyki) określa się dane wejściowe i wynik (wyniki), a także związek pomiędzy danymi a wynikami, czyli warunki, jakie muszą spełniać wyniki – określamy w ten sposób specyfikację zadania. Potem szukamy odpowiednich rozwiązań, dobieramy wzory, twierdzenia, definicje. Podobnie na zajęciach z informatyki – opis sposobu rozwiązania problemu będziemy poprzedzać **specyfikacją zadania**.

Określając specyfikację zadania, warto nadać danym i wynikom nazwy, którymi będziemy się posługiwać w dalszych etapach rozwiązywania zadania.

S Specyfikacja zadania (problemu)

Szczegółowy opis zadania, w którym określa się dane wejściowe i oczekiwane wyniki oraz związek między danymi a wynikami (warunki, jakie muszą spełniać wyniki).

Przykład 3. Formułowanie zadania i określanie jego specyfikacji – przykłady

Zadanie 1: Oblicz wartość bezwzględną dowolnej liczby całkowitej.
Dane: dowolna liczba całkowita: *x*.
Wynik: wartość bezwzględna liczby *x* równa *w*.

Zadanie 2. Uporządkuj alfabetycznie rosnąco (od A do Ż) zbiór fikcyjnych nazwisk uczniów klas pierwszych fikcyjnej szkoły.
Dane: nieuporządkowany zbiór fikcyjnych nazwisk uczniów klas pierwszych.
Wynik: uporządkowany alfabetycznie zbiór fikcyjnych nazwisk uczniów klas pierwszych.

Zadanie 3. Sprawdź, czy dany wyraz (ciąg znaków składający się z liter) jest palindromem. Palindrom to wyraz, który czyta się tak samo od lewej do prawej i od prawej do lewej, np. *potop, kajak*.
Dane: dowolny wyraz.
Wynik: jeżeli wyraz jest palindromem, określenie „tak", w przeciwnym wypadku – określenie „nie".

Ćwiczenie 2. Piszemy specyfikację zadania

Napisz specyfikacje do zadań:
1. Oblicz drogę przebytą w czasie *t* przez pojazd poruszający się ze średnią prędkością *v*.
2. Oblicz sumę liczb ujemnych i sumę liczb dodatnich w *n*-elementowym zbiorze liczb rzeczywistych różnych od zera.
3. Oblicz liczbę znaków różnych od spacji w dowolnym tekście.

Przykład 4. Formułowanie zadania i określanie jego specyfikacji

Sformułujemy zadanie i specyfikację do sytuacji problemowej omówionej w przykładzie 1.

Zadanie. Oblicz i wyprowadź aktualną cenę komputera w zależności od wartości procentowego wzrostu ceny płyty głównej (*w*). Jeśli wartość *w* mieści się w przedziale ⟨0%; 10%⟩, cenę komputera pozostaw bez zmian; jeśli mieści się w przedziale (10%; 20%⟩, cenę komputera zwiększ o 5% nowej ceny płyty głównej; jeśli $w > 20\%$, wyprowadź komunikat: „za wysoka cena".
Dane: Cena komputera *cenak*, cena płyty głównej *cenap*, wartość *w* – procent, o jaki wzrosła cena płyty głównej. Przyjmujemy założenia: $cenak > 0$, $cenap > 0$, $w \geq 0$.
Wynik: aktualna wartość ceny komputera *cenak* (jeśli $0 \leq w \leq 20$) lub napis „za wysoka cena" (jeśli $w > 20$).

Program komputerowy może zostać napisany na podstawie listy kroków, schematu blokowego lub samej specyfikacji problemu.

Przykład 5. Projektowanie rozwiązania i wybór narzędzi komputerowych

W przykładzie 1. opisaliśmy przykładowy problem, a w przykładzie 4. sformułowaliśmy zadanie i jego specyfikację, podając dane i wyniki oraz warunki, jakie muszą spełniać dane. W przyjętym rozwiązaniu występują sytuacje warunkowe. Kolejne kroki algorytmu będą zatem realizowane zależnie od spełnienia warunku.
Zaczynamy od sprawdzenia, czy $0 \leq w \leq 10$ (gdzie *w* oznacza procent, o jaki wzrośnie cena płyty głównej). Jeśli warunek jest spełniony, wyprowadzamy wynik: aktualną niezmienioną cenę komputera (*cenak*).

W przeciwnym wypadku sprawdzamy, czy $w \leq 20$. Jeśli warunek jest spełniony, obliczamy aktualną cenę płyty głównej

$$cenap = cenap \cdot (1 + \frac{w}{100})$$

i aktualną cenę komputera według wzoru:

$$cenak = cenak + \frac{5}{100} \cdot cenap$$

i wyprowadzamy wynik: aktualną cenę komputera (*cenak*). W przeciwnym wypadku (czyli dla $w > 20$) wyprowadzamy komunikat „za wysoka cena".

Po opisaniu algorytmu należy sprawdzić, czy wszystkie sformułowania i wartości podane w specyfikacji zadania i opisie algorytmu są wystarczająco precyzyjnie określone, by można było zrealizować rozwiązanie na komputerze.

Do rozwiązania zadania wykorzystamy arkusz kalkulacyjny, ponieważ możemy w nim realizować algorytmy z warunkami za pomocą funkcji JEŻELI. Można również stosować zagnieżdżoną funkcję JEŻELI, w której argumentem funkcji JEŻELI jest wartość będąca wynikiem kolejnej funkcji JEŻELI.

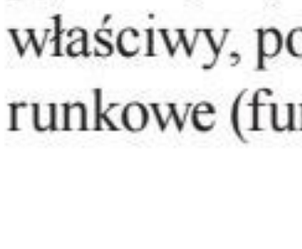
Uwaga: Zadanie sformułowane w przykładzie 4. możemy również rozwiązać w wybranym języku programowania. W temacie C4 zrealizujemy je w językach C++ i Python.

Ćwiczenie 3. Rozwiązywanie zadania za pomocą arkusza kalkulacyjnego

Korzystając z opisu przedstawionego w przykładach 4. i 5., zaprojektuj w arkuszu kalkulacyjnym tabelę do wprowadzania danych. Utwórz formułę obliczającą i wyprowadzającą odpowiedni wynik.

Każde rozwiązanie należy przetestować dla różnych danych. Danymi do zadania rozwiązywanego w ćwiczeniu 3. są liczby, określające odpowiednio: cenę komputera, cenę płyty głównej i procent, o jaki zostanie zmieniona cena płyty głównej. Przyjęliśmy odpowiedni algorytm ustalania ceny (przykład 1.). Określiliśmy też związek między danymi a wynikami (przykład 4.).

Testując rozwiązanie, będziemy oceniać **poprawność** zastosowanego algorytmu.



Algorytm jest **poprawny**, jeśli rozwiązuje problem zgodnie ze specyfikacją (czyli dla poprawnych danych daje poprawne wyniki) oraz dla poprawnych danych zawsze kończy swoje działanie (nie zapętla się).

Ćwiczenie 4. Testowanie rozwiązania

Przetestuj rozwiązanie z ćwiczenia 3. dla różnych danych. Pokaż na przykładach, że dla poprawnych danych (różnych wartości *w*, spełniających podane warunki) otrzymujesz poprawne wyniki.

Na zakończenie pracy nad omawianym problemem przeprowadzamy prezentację rozwiązania zadania i omawiamy zastosowane narzędzie. Odpowiadamy m.in. na pytanie, czy zastosowaliśmy właściwe narzędzie. Wybór arkusza kalkulacyjnego jest właściwy, ponieważ arkusz posiada mechanizmy obliczeń zawierających sytuacje warunkowe (funkcję JEŻELI).

Ćwiczenie 5. Przeprowadzanie prezentacji rozwiązania

Przedstaw i omów rozwiązanie zastosowane w ćwiczeniu 3., korzystając z projektora multimedialnego.

3. Języki programowania

Aby przedstawić algorytm w postaci programu komputerowego, należy zapisać go jako ciąg instrukcji języka programowania. Każda instrukcja odpowiada określonej operacji lub ciągowi operacji.

Słowo kluczowe S

Słowo mające szczególne znaczenie w danym języku programowania, np. oznaczające określony rozkaz, instrukcję lub polecenie.

Każdy język programowania posiada swój zbiór instrukcji, w tym **słowa kluczowe**. Podobnie jak języki naturalne, którymi posługują się ludzie, języki programowania posiadają odpowiednie **zasady składni** oraz właściwe **słownictwo**.

Każdy język naturalny dysponuje własnymi słowami, służącymi określaniu czynności i nazywaniu rzeczy. Podobnie jest z językami programowania.

Instrukcje każdego języka programowania realizują czynności takie, jak: wprowadzanie danych, wyprowadzanie wyników, wykonywanie obliczeń, sprawdzanie warunków czy powtarzanie operacji.

Niezależnie od stosowanego języka programowania należy pamiętać, że:

1. Język programowania jest językiem formalnym, co oznacza, że podlega jednoznacznym regułom. Języki naturalne nie posiadają tej cechy, ponieważ czasem sens zdania zależy od kontekstu, w którym zostało ono użyte, lub od naszej interpretacji.
2. Postać instrukcji, w tym słów kluczowych, musi być bardzo precyzyjna – zgodna z zasadami składni. W programie nie może zabraknąć ani jednego koniecznego znaku (np. dwukropka, przecinka, średnika, nawiasu). Tłumaczenie programu na kod maszynowy połączone jest ze sprawdzaniem poprawności składniowej zapisanych instrukcji. Instrukcje błędnie zapisane nie będą mogły zostać przetłumaczone, a tym samym komputer nie będzie mógł wykonać programu.
3. Kolejność zapisywania instrukcji powinna odpowiadać kolejności operacji realizujących dany algorytm.

Ze względu na **poziom wykonywania programu** języki programowania dzielimy na:

- **języki wysokiego poziomu**, np. Java, C, C++, C# (C Sharp), Python, Visual Basic, Pascal, PHP, JavaScript;
- **języki niskiego poziomu** (wewnętrzne), np. assemblery.

4. Na czym polega programowanie?

Zapisanie algorytmu w postaci ciągu instrukcji języka programowania wysokiego poziomu nazywamy **implementacją**. Powstaje wówczas tzw. **program (kod) źródłowy**.

Kod źródłowy można zapisać w dowolnym edytorze tekstu, najwygodniej jest jednak skorzystać ze specjalnych edytorów dla programistów lub z edytora wbudowanego w **środowisko programistyczne** danego języka.

Program źródłowy (który jest zapisany jako dokument tekstowy) nie może być wykonywany przez procesor, który potrafi wykonywać jedynie **kod maszynowy**. Pisząc program w języku programowania, przygotowujemy jedynie źródło do utworzenia kodu maszynowego, czyli programu rozumianego przez komputer. Komputer, a dokładniej – procesor, wykonuje program w kodzie maszynowym (w języku wewnętrznym procesora).

Program komputerowy może występować w dwóch postaciach:

- jako **program (kod) źródłowy** – program zrozumiały dla programisty, zapisany jako ciąg instrukcji tekstowych języka programowania wysokiego poziomu, np. C++ (rys. 4.), Python (rys. 5.),
- jako **program (kod) wynikowy (maszynowy)** – program zrozumiały dla komputera, napisany w języku wewnętrznym komputera w postaci ciągu instrukcji dla procesora (rys. 6.).

Program napisany w języku wysokiego poziomu (np. Python, C++) musi zostać przetłumaczony na język niskiego poziomu (język wewnętrzny komputera). Proces ten nazywamy **translacją**. Może ona przebiegać w formie **kompilacji** lub **interpretacji**, wykonywanych w **translatorze**, czyli programie do tłumaczenia programu na kod maszynowy.

Przykładami języków kompilowanych są: C, C++, C# (C Sharp), Pascal. Przykładami języków interpretowanych są: LOGO, Scratch oraz Python. W zadaniach rozwiązywanych w tym podręczniku będziemy używać języków C++ i Python.

W programowaniu bardzo istotna jest prawidłowa kolejność zapisu poleceń.

I Implementacja

Zapisanie algorytmu w postaci kodu źródłowego.

K Kod maszynowy (program maszynowy, kod wynikowy, program wynikowy)

Program napisany w języku wewnętrznym (maszynowym), wykonywany przez procesor.

I Interpretacja

Tłumaczenie programu utworzonego w języku programowania instrukcja po instrukcji, tak aby komputer mógł wykonać każdą z nich. Tłumaczenie następuje każdorazowo przy uruchomieniu programu.

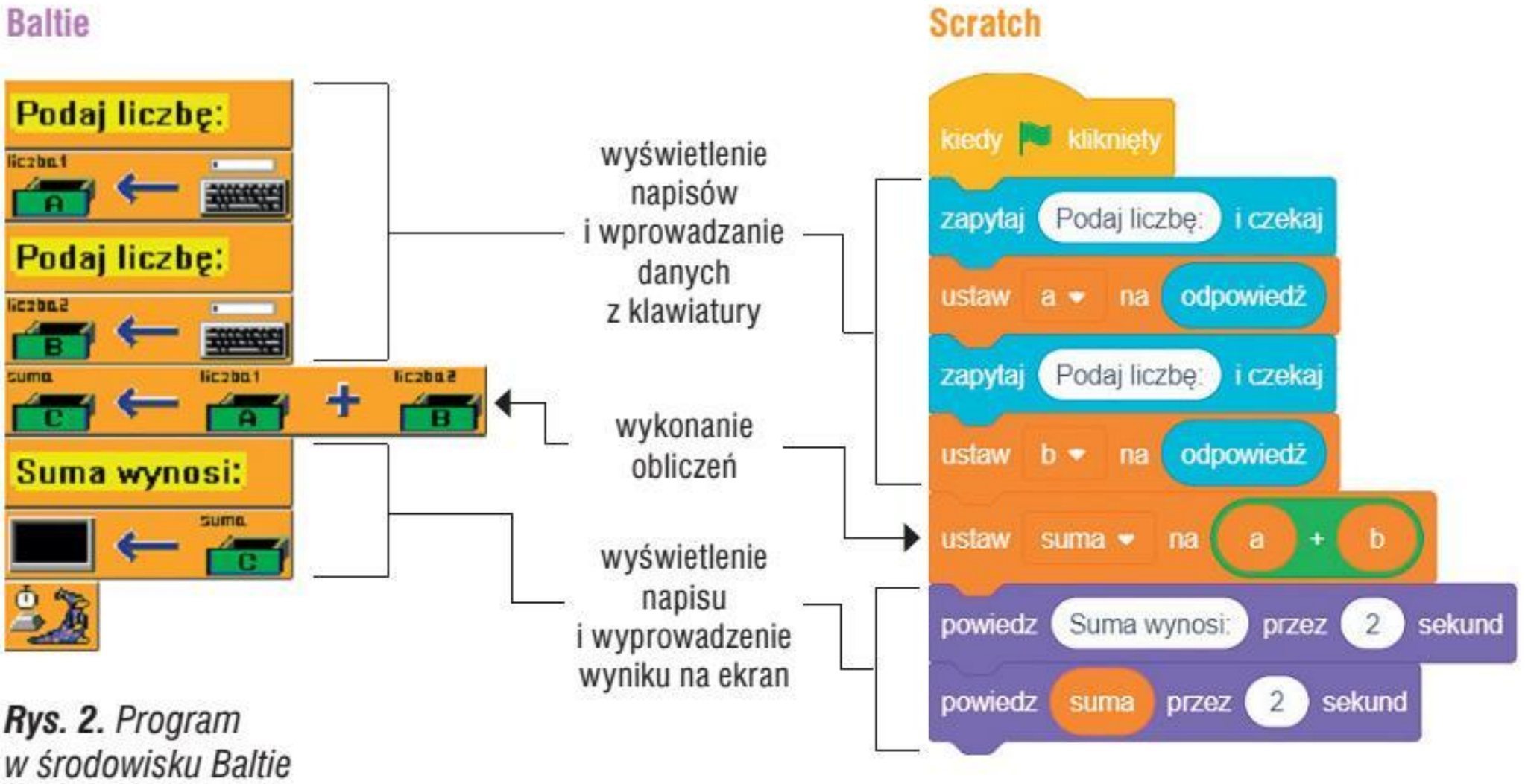
K Kompilacja

Przetłumaczenie programu w całości na język zrozumiały dla procesora, tak by mógł go wykonać komputer. Jeśli kompilacja przebiegnie poprawnie, można uruchomić program. Raz skompilowany program nie wymaga już powtórnej operacji tłumaczenia.

Przykład 6. Przedstawienie algorytmu obliczenia sumy dwóch liczb w języku programowania

Zadanie: Napisz w wybranym języku programowania program umożliwiający wprowadzenie z klawiatury dwóch liczb całkowitych, obliczający ich sumę i wyprowadzający wynik obliczeń na ekran. Przed wprowadzeniem każdej danej wyświetlaj napis „Podaj liczbę", a przed wyprowadzeniem wyniku – „Suma wynosi:".
Dane: dwie liczby całkowite: *a*, *b*.
Wyniki: wartość sumy liczb *a* i *b*: *suma*.
Na rysunkach 2. i 3. pokazano realizację zadania w środowiskach dydaktycznych Baltie i Scratch, a na rysunkach 4. i 5. – w językach C++ i Python. Na pierwszy rzut oka wydaje się, że są to cztery różne programy. Każde środowisko dydaktyczne ma inaczej wyglądające elementy graficzne, inny zapis poleceń, a każdy z języków programowania może mieć inną postać instrukcji, odmienne zasady składni, w tym stosowanie znaków interpunkcyjnych itp.
Zauważmy jednak, że w każdym rozwiązaniu występują te same części programu, ułożone w takiej samej kolejności:

1. wyświetlenie napisu i wprowadzanie danych z klawiatury,
2. wykonanie obliczeń,
3. wyświetlenie napisu i wyprowadzenie wyniku na ekran.



Rys. 2. *Program w środowisku Baltie*

Rys. 3. *Program w środowisku Scratch*

Ćwiczenie 6.

1. Porównaj programy pokazane na rysunkach 2. i 3. – podaj różnice i podobieństwa. Wskaż odpowiadające sobie polecenia i elementy (przedmioty). Na przykład: *Jakiemu elementowi w języku Scratch odpowiada przedmiot z klawiaturą w środowisku Baltie?*
2. Porównaj programy pokazane na rysunkach 4. i 5. – podaj różnice i podobieństwa. Na przykład: Czym różni się zapis instrukcji `suma = a + b` w obydwu językach? Wskaż odpowiadające sobie instrukcje.
3. W wybranym środowisku programowania (Baltie lub Scratch) utwórz program realizujący zadanie opisane w przykładzie 6. Zapisz program w pliku pod nazwą *Suma*.

Rys. 4. *Program w języku C++*

C++

[*] Suma.cpp

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a, b, suma;
6      cout << "podaj liczbe: ";
7      cin >> a;
8      cout << "podaj liczbe: ";
9      cin >> b;
10     suma = a + b;
11     cout << "Suma wynosi: " << suma;
12     return 0;
13 }
```

wyświetlenie napisów i wprowadzanie danych z klawiatury

wykonanie obliczeń

Python

```
a = int(input("Podaj liczbę: "))
b = int(input("Podaj liczbę: "))

suma = a + b

print("Suma wynosi:", suma)
```

wyświetlenie napisu i wyprowadzenie wyniku na ekran

Rys. 5. *Program w języku Python*

```
0x00401374      call    0x416610 <__main>
0x00401379      lea     -0x20(%ebp),%eax
0x0040137C      mov     %eax,(%esp)
0x0040137F      movl    $0x1,-0x58(%ebp)
0x00401386      mov     $0x477900,%ecx
0x0040138B      call    0x446984 <std::istream::operator>>(int&)>
0x00401390      sub     $0x4,%esp
0x00401393      lea     -0x24(%ebp),%eax
0x00401396      mov     %eax,(%esp)
0x00401399      mov     $0x477900,%ecx
0x0040139E      call    0x446984 <std::istream::operator>>(int&)>
0x004013A3      sub     $0x4,%esp
```

Rys. 6. *Program (kod) maszynowy, czyli wynik przetłumaczenia programu napisanego w języku C++ pokazany w symbolicznej formie*

Warto zapamiętać

- Komputery wykonują programy komputerowe, a programy realizują algorytmy (rys. 1.).
- Aby rozwiązać dowolny problem (zadanie), trzeba go poprawnie sformułować, określić dane i wyniki oraz warunki, jakie muszą zostać spełnione. Następnie należy ułożyć lub wybrać odpowiedni algorytm i określić sposób jego realizacji oraz wybrać narzędzie, które to umożliwi.
- Programowanie polega na utworzeniu kodu źródłowego, czyli zapisaniu (implementacji) algorytmu w postaci programu w wybranym języku programowania.
- Aby program był zrozumiały dla komputera, musi zostać przetłumaczony z języka wysokiego poziomu na język wewnętrzny komputera. Powstaje wówczas program (kod) maszynowy (wynikowy).
- Proces tłumaczenia kodu źródłowego na kod maszynowy nazywamy translacją. Może ona przebiegać w formie kompilacji lub interpretacji.

- Wyróżniamy następujące etapy programowania: implementację, kompilację (lub interpretację), uruchomienie i wykonanie oraz testowanie.
- Każdy język programowania posiada odpowiedni zestaw instrukcji, w tym słów kluczowych, umożliwiający zapisanie algorytmów opracowanych przez programistów.

Pytania i polecenia

1. Omów zależności przedstawione na rysunku 1.
2. Wyjaśnij pojęcia algorytmu i specyfikacji problemu (zadania).
3. Jakie są cechy algorytmu informatycznego?
4. Wymień te cechy języka programowania, które odróżniają go od języka naturalnego.
5. Czym jest implementacja?
6. Czym różni się kod źródłowy od kodu maszynowego?
7. Przedstaw i omów na konkretnym przykładzie etapy programowania.
8. Na czym polega kompilacja programu? Czym różni się od interpretacji?

Zadania

1. Napisz specyfikacje do zadań:
   a. Oblicz średnią arytmetyczną trzech liczb.
   b. Określ, czy dana litera jest samogłoską czy spółgłoską. Zależnie od wyniku wyprowadzaj napisy: „samogłoska" lub „spółgłoska".
   c. Zebrano dane o wzroście uczniów klas pierwszych w twojej szkole. Uporządkuj informacje o wzroście malejąco.
   d. Znajdź wśród danych o wzroście uczniów w twojej klasie najmniejszą i największą liczbę.
2. Wybierz jedno z zadań od 1a do 1d i wykonaj je w arkuszu kalkulacyjnym. Przetestuj rozwiązanie dla różnych danych. Czy arkusz kalkulacyjny jest odpowiednim narzędziem do wykonania każdego zadania? Uzasadnij odpowiedź.
3. W wybranym środowisku dydaktycznym (Baltie lub Scratch) utwórz program realizujący zadanie 1a. Zapisz plik pod nazwą *Średnia*.

Dla zainteresowanych

4. Opisz (w edytorze tekstu lub zeszycie przedmiotowym) przykładową sytuację problemową, w której występują sytuacje warunkowe (możesz wzorować się na opisie problemu z przykładu 1.). Sformułuj zadanie i zapisz jego specyfikację. Zaprojektuj rozwiązanie zadania, wzorując się na przykładzie 5.
5. Wyjaśnij pochodzenie słowa *algorytm*. Informacje znajdź w Internecie.

Przeczytaj, jeśli chcesz wiedzieć więcej…

Pierwszy język wysokiego poziomu powstał w 1954 roku. Nazywał się **FORTRAN** (nazwa pisana wielkimi literami). Kompilator FORTRAN-u był również pierwszym kompilatorem języka wysokiego poziomu. Język ten doczekał się wielu wersji. Do wersji 77. nazywał się FORTRAN (nazwa pisana wielkimi literami), a od wersji 90. do dziś – **Fortran** (nazwa jedynie zaczyna się wielką literą). Co ciekawe, język ten nadal jest rozwijany i używany. Uznaje się go za najpopularniejszy język do obliczeń numerycznych, naukowo-inżynierskich, symulacji. Kolejne wersje umożliwiają programowanie strukturalne, obiektowe, modularne i równoległe.
W Polsce Fortran stosowano w komputerach typu ODRA i Mera.

Temat C2. Tworzenie programów w języku C++

1. Środowisko programistyczne języka C++
   1.1. Struktura programu w języku C++
   1.2. Etapy tworzenia programu w języku C++
2. Stosowanie zmiennych
   2.1. Deklarowanie zmiennych
   2.2. Nadawanie zmiennym wartości
   2.3. Wykonywanie obliczeń
3. Wyprowadzanie komunikatów i wyników na ekran monitora
4. Zapisywanie rozwiązania problemu w języku C++

Warto powtórzyć

1. Jaką funkcję pełni pamięć operacyjna (RAM)?
2. Jakie informacje należy umieścić w specyfikacji problemu?
3. Wyjaśnij pojęcia: *język programowania*, *program komputerowy*, *słowo kluczowe*.
4. Dlaczego o języku programowania mówimy, że jest językiem formalnym?
5. Czym jest kod źródłowy?
6. Z czego powinna wynikać kolejność instrukcji programu?
7. Na czym polega kompilacja programu?
8. Dlaczego błędnie napisane instrukcje nie zostaną wykonane?

1. Środowisko programistyczne języka C++

Aby pisać programy w języku C++, należy zainstalować wybraną aplikację zawierającą edytor i kompilator języka C++, np., Code::Blocks, Dev-C++. Są one dostępne bezpłatnie w Internecie. Można także skorzystać z rozwiązań komercyjnych, np. Visual Studio, C++Builder.

Wyspecjalizowany pakiet programistyczny zawiera **edytor kodu źródłowego**, kompilator i inne narzędzia wspomagające programowanie. Taką aplikację (lub zespół aplikacji) określa się też jako **zintegrowane środowisko programistyczne**. W tym podręczniku będziemy korzystać ze środowiska Dev-C++.

Z **Zintegrowane środowisko programistyczne**
(IDE – z ang. *Integrated Development Environment*)

Aplikacja lub zespół aplikacji do tworzenia, modyfikowania, testowania i utrzymywania oprogramowania.

1.1. Struktura programu w języku C++

Prosty program w języku C++ składa się z **poleceń preprocesora** (tzw. **dyrektyw**) oraz **funkcji głównej** `main()`.

W programie musi wystąpić funkcja `main()`, od której rozpoczyna się wykonanie programu. Pozostałe elementy programu są opcjonalne. Instrukcje języka C++ kończymy średnikami.

W języku C++ **blok programu** wyznaczają nawiasy klamrowe {}. **Komentarze** (ignorowane przez kompilator) umieszczamy po znakach // w tej samej linii lub wewnątrz znaków /* (początek bloku komentarza) i */ (koniec bloku komentarza).

Przykładowe elementy programu, ułożone w zwykle stosowanej kolejności, pokazujemy na rysunku 1.

Preprocesor P

Program, który przetwarza wstępnie kod źródłowy przed rozpoczęciem zasadniczej kompilacji. Polecenia preprocesora (zwane dyrektywami) rozpoczynają się od znaku #. Np. dyrektywa `#include` nakazuje dołączenie do kodu źródłowego kodu z innych plików źródłowych.

C++

```
#include <iostream>
using namespace std;
// deklaracja zmiennych globalnych (jeśli występują)
// deklaracja funkcji (jeśli występują)

int main()
{
      // deklaracja zmiennych lokalnych (jeśli występują)
      // część wykonawcza programu
   return 0;
)
```

Rys. 1. *Ogólna struktura programu w języku C++*

1.2. Etapy tworzenia programu w języku C++

W każdym języku kompilowanym możemy wyróżnić następujące etapy tworzenia programu:

1. W edytorze wbudowanym w środowisko programistyczne piszemy kod źródłowy programu (rys. 2.).
2. Zapisujemy kod źródłowy programu w pliku (w języku C++ zwykle w pliku z rozszerzeniem *cpp*).
3. Kompilujemy program, korzystając z opcji **Kompiluj** (**Compile**). Przebieg kompilacji obserwujemy w dolnej części okna programu (rys. 2.). Jeśli program został skompilowany, zostanie utworzony tzw. **plik wykonywalny** z rozszerzeniem *exe*.
4. Skompilowany program uruchamiamy, zazwyczaj korzystając z opcji **Uruchom** (**Run**). Wynik działania programu pokaże się w osobnym oknie (rys. 3.).
5. Jeżeli w trakcie kompilacji wystąpiły błędy, należy je poprawić (kompilator wskazuje prawdopodobne miejsce popełnienia błędu – rys. 4a i 4b) i ponownie przejść do kroku 2.

W programowaniu w języku C++, jak w każdym języku, bardzo istotna jest prawidłowa kolejność zapisu poleceń. Instrukcje języka C++, podobnie jak innych języków programowania, realizują takie czynności, jak: wprowadzanie danych, wyprowadzanie wyników, wykonywanie obliczeń, sprawdzanie warunków czy powtarzanie operacji. Poznamy i zastosujemy te instrukcje w ćwiczeniach i zadaniach w tematach C3, C4 i C5.



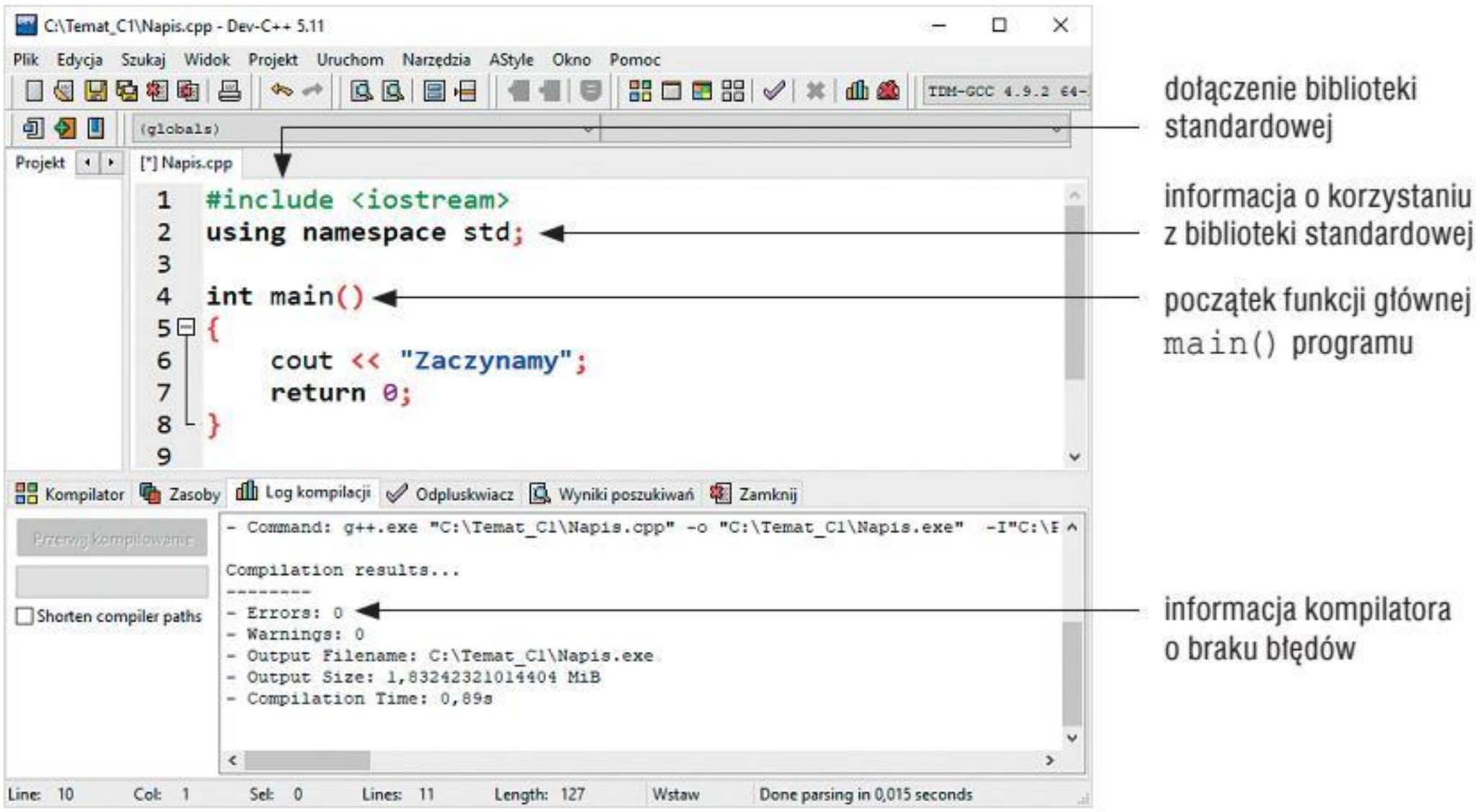
Uruchomiony i wykonywany może być program, który został poprawnie skompilowany. Jeśli kompilacja nie przebiegła prawidłowo, należy poprawić program i ponownie go skompilować. Raz skompilowany program nie wymaga już powtórnego tłumaczenia.

W trakcie wykonywania programu procesor rozpoznaje i wykonuje instrukcje swojego wewnętrznego języka. Po wykonaniu programu wyniki pojawią się na wybranym urządzeniu zewnętrznym, np. na ekranie monitora (rys. 3.).

Po uruchomieniu środowiska programistycznego języka C++ zobaczymy okno z głównym menu programu. Okno edytora wbudowanego do środowiska programistycznego otwieramy w menu **Plik** – możemy otworzyć istniejący plik (plik źródłowy) lub utworzyć nowy.



Aby utworzyć nowy plik źródłowy, należy w menu **Plik** wybrać opcję **Nowy/Plik źródłowy**.
Aby skompilować program, należy w menu **Uruchom** wybrać opcję **Kompiluj**.
Aby uruchomić program, należy w menu **Uruchom** wybrać opcję **Uruchom**.



***Rys. 2.** Okno środowiska programistycznego języka C++ z programem źródłowym – kompilacja przebiegła prawidłowo*

Uwagi:

- Funkcja `main()` zawiera główny program. Przed nazwą funkcji powinno się znajdować słowo kluczowe **`int`**, informujące, że funkcja `main()` zwraca wartość liczbową.
- Każda instrukcja programu powinna być zakończona średnikiem.
- Instrukcja **`return`** `0` na końcu funkcji powoduje zwrócenie do systemu operacyjnego wartości 0, informującej o prawidłowym wykonaniu programu.

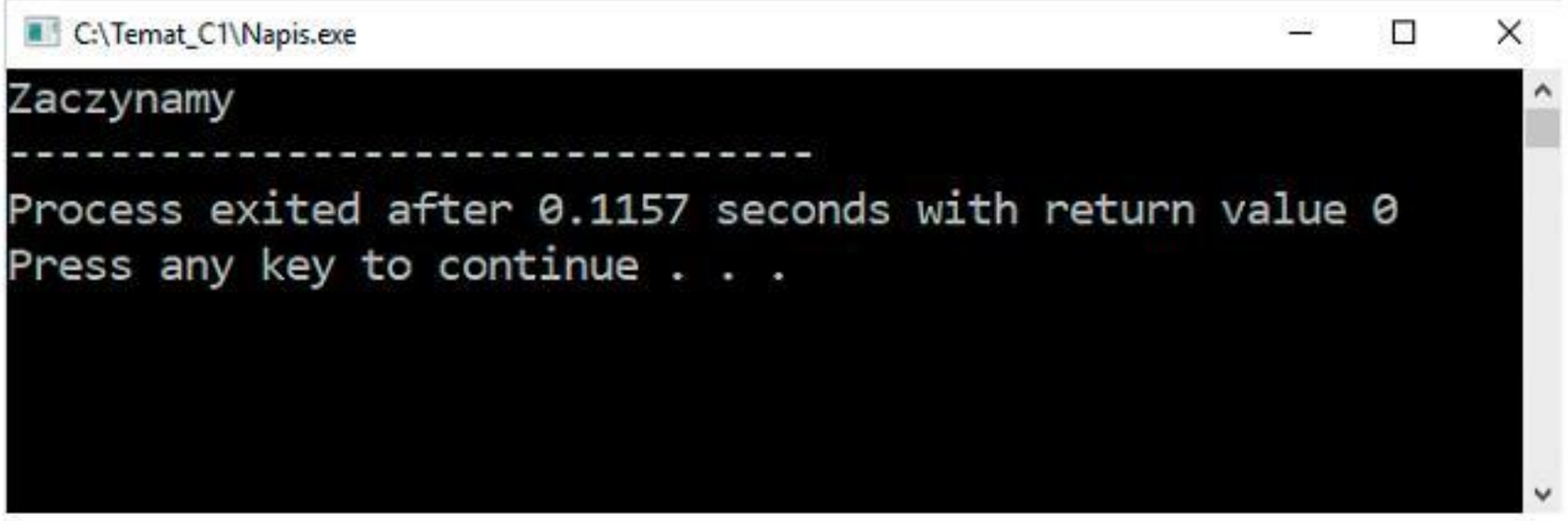
Ćwiczenie 1. Poznajemy etapy tworzenia programu w języku C++

1. Utwórz nowy plik źródłowy i napisz program wyświetlający na ekranie napis „Zaczynamy" – przepisz instrukcje pokazane na rysunku 2.
2. Zapisz program w pliku pod nazwą *Napis*.
3. Skompiluj program.
4. Uruchom program.
5. Jeśli kompilator wykrył błędy, popraw je. Zapisz program w pliku pod tą samą nazwą. Ponownie skompiluj i uruchom program.

Aby powiększyć czcionkę w oknie wykonania programu, należy nacisnąć przycisk w lewym górnym rogu okna i wybrać polecenie **Właściwości/Czcionka**.

Dobra rada

Staraj się samodzielnie poprawiać program do momentu, gdy da się go skompilować i program będzie można uruchomić. W nauce programowania umiejętność wyszukiwania i poprawiania błędów w programie jest bardzo ważna.

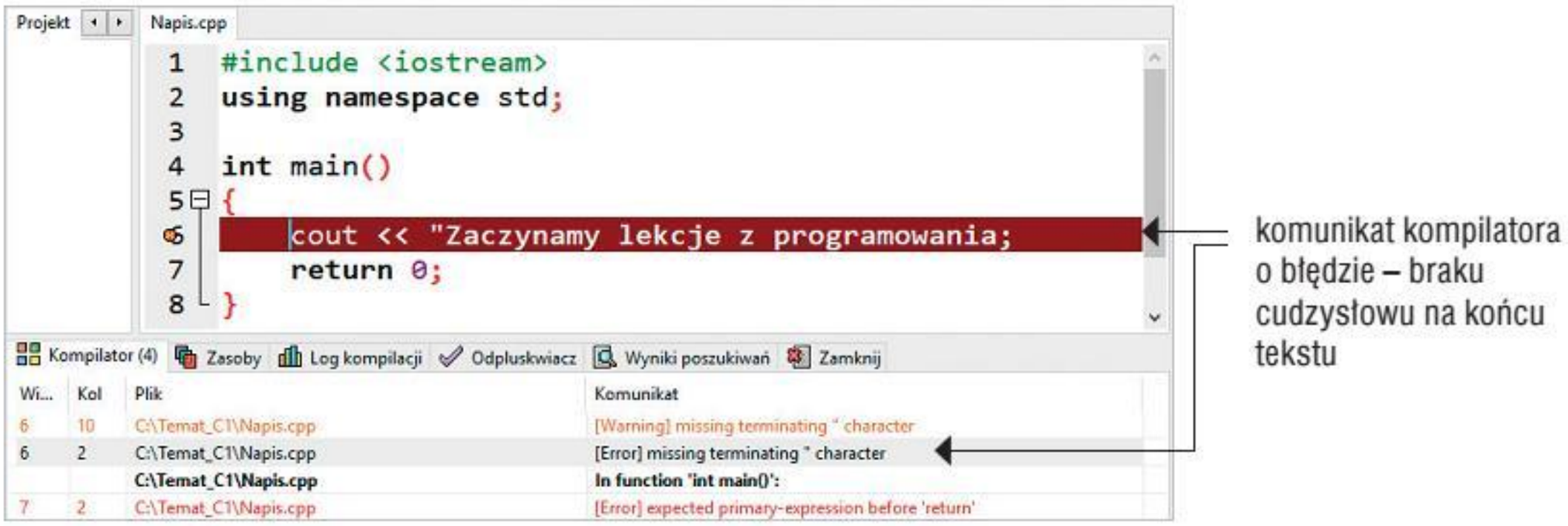


Rys. 3. *Wynik działania programu – ćwiczenie 1.*

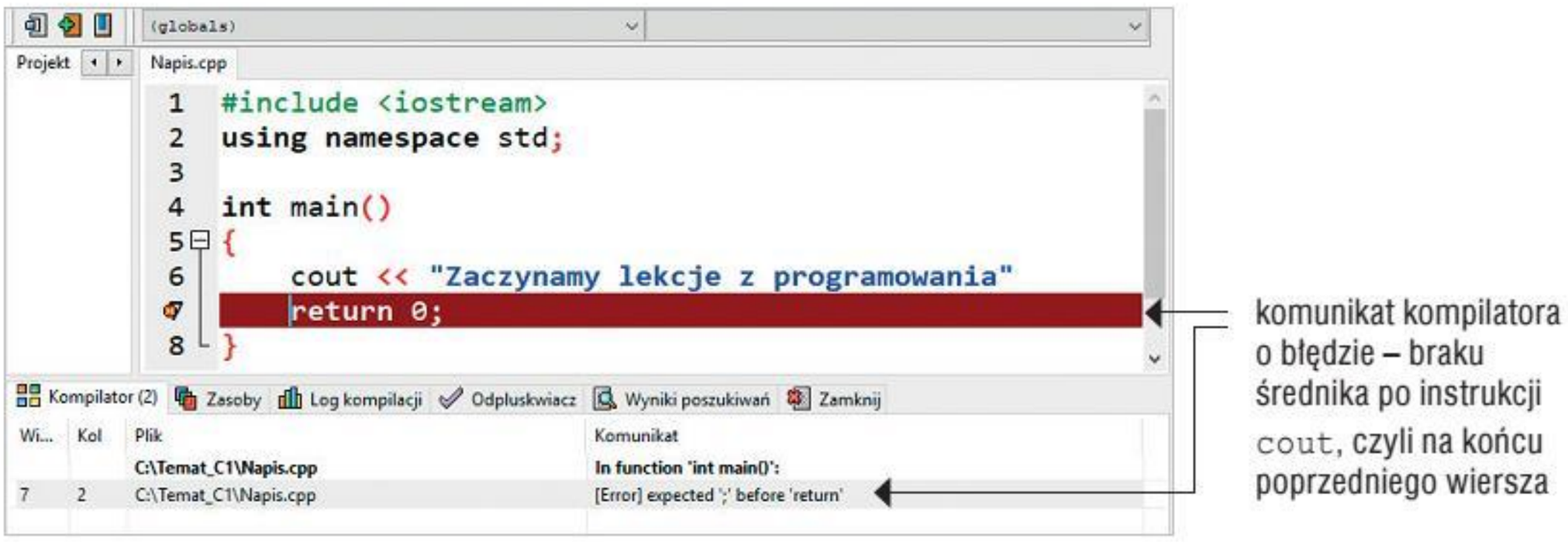
Ćwiczenie 2. Modyfikujemy program

1. Zmodyfikuj program utworzony w ćwiczeniu 1., dodając dalszą część napisu: „lekcje z programowania".
2. Zapisz plik pod tą samą nazwą.
3. Jeśli kompilator wykrył błędy, popraw je. Ponownie zapisz plik i skompiluj program.
4. Jeśli kompilacja przebiegła pomyślnie, uruchom program.

Wskazówka: Na rysunkach 4a i 4b pokazano przykładowe błędy wykryte przez kompilator. Oba błędy są w szóstym wierszu programu, ale na rysunku 4b kompilator podświetlił wiersz siódmy. W tym przypadku bowiem brakuje średnika na końcu poprzedniego wiersza.

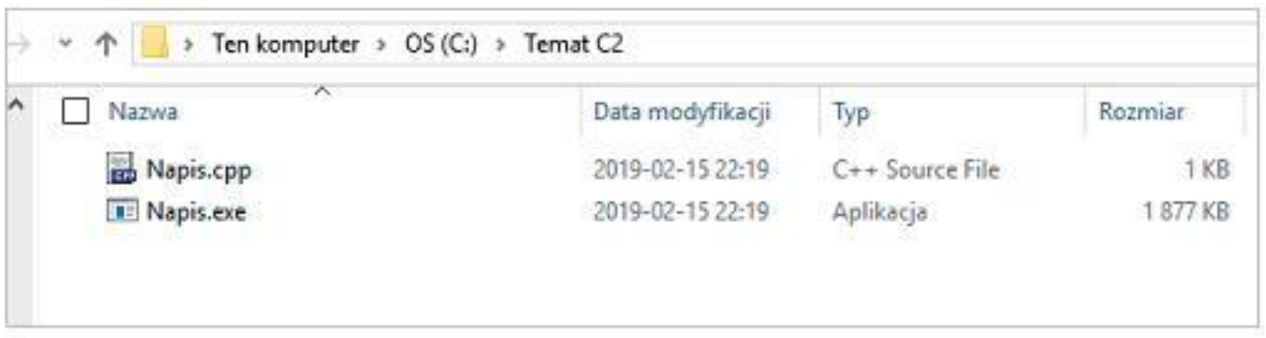


Rys. 4a. *Okno środowiska programistycznego języka C++ z programem źródłowym – kompilator wykrył błąd w programie*



Rys. 4b. *Okno środowiska programistycznego języka C++ z programem źródłowym – kompilator wykrył błąd w programie*

Po skompilowaniu programu źródłowego zapisanego w pliku *Napis.cpp* zostanie utworzony plik wykonywalny *Napis.exe* (aplikacja) o takiej samej nazwie własnej (*Napis*), ale z rozszerzeniem *exe* (rys. 5.). Aby uruchomić program po zamknięciu okna środowiska programistycznego, wystarczy kliknąć dwukrotnie nazwę lub ikonę *Napis.exe*. Program uruchomi się w oknie pokazanym na rysunku 3.



Rys. 5. *W oknie Eksploratora plików widać dwa pliki – źródłowy i wykonywalny*

2. Stosowanie zmiennych

Chcemy obliczyć iloczyn dwóch liczb całkowitych. Jak napisać program w języku C++ umożliwiający wprowadzenie dwóch liczb całkowitych z klawiatury, obliczający ich iloczyn i wyprowadzający wynik obliczeń na ekran?

Program, który utworzyliśmy w ćwiczeniach 1. i 2., wyświetlał na ekranie napis – nie używaliśmy w nim zmiennych i nie wykonywaliśmy obliczeń. Struktura kolejnych programów będzie bardziej rozbudowana, m.in. dodamy **deklarację zmiennych**.

2.1. Deklarowanie zmiennych

Komórka pamięci K

Część pamięci operacyjnej komputera (pamięci RAM). Pamięć operacyjna jest podzielona logicznie na komórki pamięci. Każda komórka pamięci ma swój unikatowy adres. W komórkach pamięci mogą być przechowywane dane reprezentujące instrukcje procesora lub dane dla tych instrukcji.

Typ zmiennej T

Określa rodzaj danych przechowywanych w zmiennej – mogą to być np. liczby, znaki, wartości logiczne (**typy proste**) lub teksty, tablice, rekordy i obiekty (**typy złożone**).

W wielu językach programowania, w tym w języku C++, zmienne przed użyciem należy **zadeklarować**. Deklaruje się zmienne, w których będą przechowywane dane wejściowe dla programu, wyniki działania programu, a także dane pomocnicze niezbędne do wykonania obliczeń. W deklaracji zmiennej podaje się jej **typ** oraz nazwę. Na tym etapie programowania będziemy korzystać wyłącznie z **prostych typów danych**.

Wykorzystywanym w programie zmiennym typu prostego przyporządkowane są określone **komórki pamięci**.

Dla każdej zadeklarowanej zmiennej rezerwowany jest fragment pamięci o określonym adresie i wielkości odpowiadającej danemu typowi danych. Na przykład dla zmiennych typu całkowitego `int` to cztery bajty lub osiem bajtów (zależnie od używanego kompilatora), dla zmiennych typu rzeczywistego `float` – cztery bajty, a wartość zmiennej znakowej `char` pamiętana jest w jednym bajcie lub w dwóch bajtach.



Użycie w programie niezadeklarowanej zmiennej zostanie zgłoszone podczas kompilacji jako błąd (program nie zostanie skompilowany).

Języki programowania oferują różne typy danych, które dobieramy zależnie od zadania, np. liczby całkowite, gdy mamy obliczyć sumę liczb całkowitych, lub znaki, gdy mamy sprawdzić, czy wprowadzona litera jest samogłoską.

Deklarowanie zmiennych	`opis_typu lista_zmiennych;`	C++

Przykład 1. Deklarowanie zmiennych w języku C++

Deklarowanie zmiennych typu całkowitego **(int)**

```
int liczba;
int a, b, Suma;
int a, b, iloczyn;
int liczba_sztuk, dlugosc;
```

Deklarowanie zmiennych typu rzeczywistego (**float**)

```
float minimum;
float a, b, c, suma, srednia;
float Cena_jedn, Dlugosc;
```

Zasady dotyczące nazw zmiennych w języku C++

1. Wielkie i małe litery w nazwach traktowane są odmiennie (np. `suma` i `Suma` oznaczać będą różne zmienne). Pisząc program w języku C++, należy zwracać uwagę na poprawne używanie małych i wielkich liter.
2. W nazwach zmiennych można stosować litery (bez polskich znaków diakrytycznych), cyfry i znak podkreślenia. Nazwa nie może zaczynać się od cyfry.
3. W nazwach zmiennych nie wolno stosować spacji.

! Uwaga

W nazwach plików, w których zapisujemy programy tworzone w języku C++, nie będziemy stosować polskich znaków diakrytycznych i spacji (w nazwach wieloczłonowych użyjemy znaku podkreślenia).

W kilku kolejnych ćwiczeniach napiszemy program obliczający iloczyn dwóch liczb całkowitych wprowadzanych z klawiatury. Zaczniemy od zadeklarowania zmiennych.

Ćwiczenie 3. Zapisujemy dane i wyniki do zadania

Wzorując się na opisie podanym w przykładzie 6. z tematu C1, zapisz w zeszycie przedmiotowym dane i wyniki do zadania: *Napisz w wybranym języku programowania program umożliwiający wprowadzenie z klawiatury dwóch liczb całkowitych, obliczający ich iloczyn i wyprowadzający wynik obliczeń na ekran. Przed wprowadzeniem pierwszej danej wyświetlaj napis „Podaj pierwsza liczbe", przed wprowadzaniem drugiej – „Podaj druga liczbe", a przed wyprowadzeniem wyniku – „Iloczyn wynosi:".*

! Uwaga

Aby uniknąć problemów z poprawnym wyświetlaniem komunikatów, nie będziemy w nich używać polskich znaków diakrytycznych.

Ćwiczenie 4. Deklarujemy zmienne

1. Utwórz nowy plik źródłowy. Zgodnie z zapisaną specyfikacją zadeklaruj zmienne do obliczenia iloczynu dwóch liczb całkowitych.
2. Zapisz program w pliku pod nazwą *Iloczyn* i skompiluj. Jeśli kompilator wykrył błędy, popraw program, ponownie zapisz, skompiluj i uruchom.

2.2. Nadawanie zmiennym wartości

Aby nadać zmiennej wartość, można wprowadzić wartość ze **standardowego wejścia** (zwykle klawiatury), stosując polecenie `cin >>`, lub zastosować **instrukcję przypisania**.

W języku C++ `cin` jest **obiektem** reprezentującym standardowe wejście programu. **Operator** >> oznacza wprowadzenie danych do odpowiedniej zmiennej, podanej po prawej stronie. Jeżeli chcemy wprowadzić dane do kilku zmiennych, należy operatora >> użyć wielokrotnie.

Wprowadzanie danych z klawiatury	`cin >> nazwa_zmiennej;`	C++

Przykład 2. Wprowadzanie danych

```
cin >> dlugosc;
cin >> a >> b;
cin >> Liczba1 >> Liczba2 >> Liczba3;
```

W **instrukcji przypisania** zmiennej podanej po lewej stronie instrukcji zostanie przypisana obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji.

Instrukcja przypisania	`zmienna = wyrażenie;`	C++

Wyrażeniem mogą być m.in. konkretna wartość, zmienna, wyrażenie algebraiczne.

Przykład 3. Stosowanie instrukcji przypisania

```
Rok = 2019;
a = 7.58;
x = y;
Obwod = 2 * a + 2 * b;
```

Uwaga: W języku C++ operacji przypisania można użyć wielokrotnie w ramach jednej instrukcji, np. zapis: `a = b = c = 150;` oznacza, że wartość 150 zostanie przypisana najpierw zmiennej `c`, potem wartość zmiennej `c` zostanie przypisana zmiennej `b`, a na koniec wartość zmiennej `b` zostanie przypisana zmiennej `a`. Po wykonaniu instrukcji zmienne `a`, `b` i `c` będą miały wartość 150.

Ćwiczenie 5. Wprowadzamy dane z klawiatury

1. Otwórz plik *Iloczyn* zapisany w ćwiczeniu 4.
2. Dodaj polecenie wprowadzania danych z klawiatury, zgodnie z programem pokazanym na rysunku 6.
3. Zapisz plik pod tą samą nazwą i skompiluj program. Jeśli kompilator wykrył błędy, popraw je, ponownie zapisz plik i skompiluj program.

[*] Iloczyn.cpp

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a, b, iloczyn;
6
7      cin >> a;
8      cin >> b;
9
10     return 0;
11 }
```

Rys. 6. *Program umożliwiający wprowadzenie dwóch liczb z klawiatury – ćwiczenie 5.*

2.3. Wykonywanie obliczeń

Aby zapisać w programie obliczenia, możemy skorzystać z instrukcji przypisania.

W języku C++ stosujemy następujące podstawowe **operatory arytmetyczne**: + (dodawania), - (odejmowania), * (mnożenia), / (dzielenia) oraz % (reszty z dzielenia). Operator / oznacza dzielenie, w zależności od typu danych – całkowite (z obcięciem części ułamkowej) lub zmiennoprzecinkowe (z zachowaniem części ułamkowej), a operator % – obliczenie reszty z dzielenia dwóch liczb całkowitych.

Operator	Działanie	Przykład instrukcji przypisania	Wynik działania dla danych: a = 11 i b = 4
+	dodawanie	`suma = a + b;`	11 + 4 = 15
-	odejmowanie	`roznica = a - b;`	11 – 4 = 7
*	mnożenie	`iloczyn = a * b;`	11 * 4 = 44
/	dzielenie w zależności od typu danych: • całkowite (z obcięciem części ułamkowej) – dla wartości typu całkowitego	`iloraz = a / b;`	11 / 4 = 2
	• zmiennoprzecinkowe (z zachowaniem części ułamkowej) – jeśli przynajmniej jedna liczba jest wartością typu rzeczywistego	`iloraz = a / b;`	11 / 4 = 2.75
%	obliczenie reszty z dzielenia dwóch liczb całkowitych	`reszta = a % b;`	11 % 4 = 3

Tabela 1. *Podstawowe operatory arytmetyczne w języku C++*

Ćwiczenie 6. Stosujemy instrukcję przypisania do zapisania obliczeń

1. Otwórz plik *Iloczyn* zapisany w ćwiczeniu 5.
2. Dodaj obliczanie iloczynu tak jak pokazano na rysunku 7.
3. Zapisz plik pod tą samą nazwą. Skompiluj i uruchom program. Dlaczego nie widzisz wyniku obliczeń na ekranie? Uzasadnij odpowiedź.

[*] Iloczyn.cpp

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a, b, iloczyn;
6
7      cin >> a;
8      cin >> b;
9      iloczyn = a * b;
10
11     return 0;
12 }
```

Rys. 7. *Program obliczający iloczyn dwóch liczb wprowadzanych z klawiatury – ćwiczenie 6.*

3. Wyprowadzanie komunikatów i wyników na ekran monitora

W ćwiczeniach 1. i 2. wyprowadzaliśmy komunikaty na ekran monitora, używając polecenia `cout <<`. Za pomocą tego polecenia można wyprowadzać również wyniki obliczeń.

Wyprowadzanie wyników i komunikatów na ekran monitora	`cout << wartość;`	C++



W języku C++ `cout` jest obiektem reprezentującym standardowe wyjście programu (zwykle okno systemu operacyjnego na ekranie monitora).
Operator << oznacza wyprowadzenie wartości podanej po prawej stronie.
Jeżeli chcemy wyprowadzić kilka wartości, należy użyć operatora << wielokrotnie.

Wartością może być zmienna, wyrażenie, a także napis. Napis umieszczany w programie, który ma zostać wyświetlony na ekranie monitora, musi być ujęty w górne cudzysłowy "".



Przykład 4. Wyprowadzanie komunikatów i wyników

```
cout << "Zaczynamy lekcje z programowania";
cout << P;
cout << a + b;
cout << "Obwod = " << 2 * a + 2 * b;
cout << "Obwod wynosi: " << obwod;
```

Umieszczenie w wierszu z instrukcją wyprowadzania danych zapisu `endl` lub znaków `"\n"` powoduje przejście kursora na początek następnego wiersza po wyświetleniu komunikatu lub wyniku, za którym umieszczono te znaki.

Polecenia:

```
cout << "Liczba dni wolnych" << endl;
cout << liczba;
```

lub

```
cout << "Liczba dni wolnych" << endl << liczba;
```

lub

```
cout << "Liczba dni wolnych" << "\n";
cout << liczba;
```

lub

```
cout << "Liczba dni wolnych\n" << liczba;
```

wyświetlą w taki sam sposób te same wyniki

```
Liczba dni wolnych
12
```

Uwaga: Jeśli wprowadzamy z klawiatury podwójny cudzysłów, edytor środowiska Dev-C++ od razu wypisuje obydwa górne cudzysłowy i ustawia pomiędzy nimi kursor: `""`. Podobnie dzieje się, gdy wprowadzamy otwierający nawias okrągły po nazwie funkcji, np. `main`: `int main()`, czy otwierający nawias klamrowy zaczynający blok.

Ćwiczenie 7. Wyprowadzamy napisy i wynik na ekran

1. Otwórz plik zapisany w ćwiczeniu 6.
2. Dodaj wyświetlanie komunikatów, podobnie jak pokazano na rysunku 8a.
3. Zapisz plik pod tą samą nazwą. Skompiluj i uruchom program.
4. Co sądzisz o programach pokazanych na rysunkach 8a i 8b? Wskaż kilka zasadniczych różnic.

[*] Iloczyn.cpp

```cpp
#include <iostream>
using namespace std;
int main()
{
    int a, b, iloczyn;

    cout << "podaj pierwsza liczbe: ";
    cin >> a;
    cout << "podaj druga liczbe: ";
    cin >> b;
    iloczyn = a * b;
    cout <<"Iloczyn wynosi: "<< iloczyn;

    return 0;
}
```

DOBRZE

Dobra rada

Mimo że kompilator pomija dodatkowe spacje i wcięcia, kod programu powinien być napisany czytelnie, przejrzyście i jednolicie, (rys. 8a). Innej osobie, trudniej jest pracować dalej nad programem napisanym w złym stylu (rys. 8b).

Rys. 8a. *Program napisany czytelnie i przejrzyście – ćwiczenie 7.*

[*] Iloczyn.cpp

```cpp
#include <iostream>
using namespace std;
int main()
{int a , b,iloczyn ;
cout << "podaj pierwsza liczbe: " ;cin>> a;
    cout << "podaj druga liczbe: ";
cin >>b;iloczyn =a * b;cout <<"Iloczyn wynosi: "<<iloczyn;
        return 0;
}
```

ŹLE

Rys. 8b. *Program mniej czytelny i przejrzysty – ćwiczenie 7.*

Ćwiczenie 8. Testujemy działanie programu obliczającego średnią dla danych różnego typu

1. Do programu zapisanego w ćwiczeniu 7. dodaj obliczanie średniej arytmetycznej liczb *a* i *b*. Zadeklaruj zmienną *srednia* typu całkowitego `int`. Wynik wyprowadź na ekran. Dodaj odpowiednie komunikaty.
2. Zapisz program w pliku pod nazwą *Srednia_c*. Skompiluj go i uruchom. Przetestuj program dla par liczb (wartości zmiennych *a* i *b*): (2; 3), (14; 8); (7; 6).
3. Zmień typ wszystkich danych na `float`. Zapisz program pod nazwą *Srednia_r*, skompiluj i uruchom. Przetestuj program dla tych samych danych. Porównaj otrzymane wyniki z wynikami działania programu *Srednia_c*. Wyjaśnij, dlaczego dla tych samych danych otrzymujemy różne wyniki. Dodatkowo przetestuj program *Srednia_r* dla liczb: (342,3; 25,7) (3763,82; 109,87).

Wskazówka: Pamiętaj, aby liczby dziesiętne wprowadzać z kropką, a nie z przecinkiem.

Uwaga

W większości języków programowania, w tym w języku C++, część dziesiętną liczby zapisujemy po kropce, a nie po przecinku, jak na matematyce. W ten sam sposób liczby dziesiętne są również wyświetlane na ekranie.

4. Zapisywanie rozwiązania problemu w języku C++

Chcesz przygotować tradycyjną sałatkę jarzynową na imprezę urodzinową. W Internecie są przepisy, ale w proporcjach dla dwóch, pięciu czy dziesięciu osób. Nie możesz znaleźć przepisu dla trzynastu osób – uczestników spotkania. Jak napisać program w języku C++, który obliczy, ile potrzebujemy danego składnika sałatki i wyświetli wynik na ekranie?

Przykład 5. Sformułowanie zadania, opis specyfikacji i projekt rozwiązania

Zadanie: Napisz program obliczający, ile potrzeba danego składnika sałatki dla *o2* osób, gdy znasz ilość składnika dla *o1* osób.

Dane: liczba całkowita $o1 > 0$ oznaczająca liczbę osób uwzględnioną w przepisie, liczba rzeczywista $s1 > 0$, określająca ilość składnika dla *o1* osób, liczba całkowita $o2 > 0$ oznaczająca liczbę uczestników spotkania.

Wynik: liczba rzeczywista $s2 > 0$ oznaczająca ilość składnika dla *o2* osób.

W rozwiązaniu skorzystamy z zależności:
jeśli

s1 ⟶ dla *o1* osób
s2 ⟶ dla *o2* osób

to

$$s2 = \frac{s1 \cdot o2}{o1}$$

Aby zapisać te obliczenia w języku C++, zastosujemy instrukcję przypisania:

```
s2 = (s1 * o2) / o1;
```

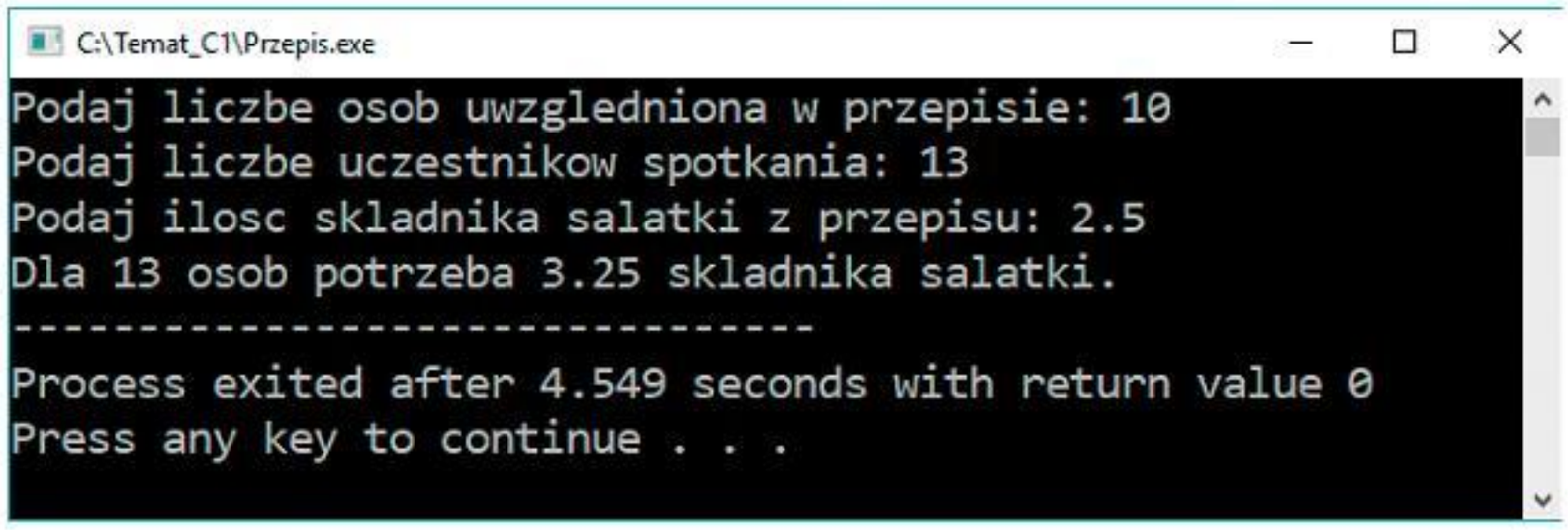
Uwaga: Z matematycznego punktu widzenia nawiasy są zbędne – umieściliśmy je w programie dla zwiększenia czytelności.

Ćwiczenie 9. Realizujemy algorytm obliczania ilości składnika sałatki w języku C++

Uwaga

Na razie w programach nie sprawdzamy poprawności danych wprowadzanych przez użytkownika. Zakładamy, że użytkownik podaje poprawne dane. Problem poprawności danych omówimy w temacie C4.

1. Utwórz program realizujący zadanie sformułowane w przykładzie 5. Zadeklaruj zmienne zgodnie ze specyfikacją zadania. Dodaj przed wprowadzaniem zmiennych komunikaty. Dodaj również komunikat przed wyprowadzeniem wyniku. Komunikaty i wynik wyprowadź na ekran monitora tak jak pokazano na rysunku 9.
2. Zapisz program w pliku pod nazwą *Przepis*. Skompiluj i uruchom program. Przetestuj go dla następujących trójek liczb (wartości zmiennych *o1, o2*, *s1*): (10; 13; 2,5), (3; 12; 0,45), (15; 4; 3,20).
3. Przetestuj program dla trzech innych trójek liczb.



Rys. 9. *Wynik działania programu – ćwiczenie 9.*

Wskazówki:

- Jeśli musimy zadeklarować zmienne różnych typów, warto dla czytelności programu zapisać deklaracje w dwóch kolejnych wierszach, np.:

```
int o1, o2;
float s1, s2;
```

- Aby wyświetlić wyniki tak jak pokazano na rysunku 9., należy wpisać instrukcję:

```
cout << "Dla " << o2 << " osob potrzeba " << s2
<< " skladnika salatki.";
```

Zauważmy, że program utworzony w ćwiczeniu 9. oblicza ilość tylko jednego składnika sałatki. Jeśli chcemy obliczyć ilość kolejnego składnika, musimy ponownie uruchomić program. W temacie C5 zmodyfikujemy ten program tak, aby obliczał i wyświetlał na ekranie kolejno ilości wszystkich składników.

Warto zapamiętać

- Aby utworzyć program w języku C++, piszemy kod źródłowy w specjalnym edytorze wbudowanym w środowisko programistyczne, zapisujemy program w pliku i kompilujemy. Jeśli kompilacja przebiegła prawidłowo, uruchamiamy program. Wynik działania programu pokaże się w osobnym oknie i zostanie utworzony plik wykonywalny z rozszerzeniem *exe*.
- Funkcja główna `main()`, rozpoczynająca działanie programu, musi wystąpić w każdym programie w języku C++. Inne elementy są opcjonalne.
- W języku C++ każda zmienna używana w programie musi zostać zadeklarowana przed pierwszym użyciem poprzez określenie jej typu i nazwy.
- Wartość zmiennej możemy wprowadzić z klawiatury (stosując polecenie `cin >> zmienna`) lub nadać zmiennej wartość za pomocą instrukcji przypisania.
- Obliczenia zapisujemy, korzystając z instrukcji przypisania. Zmiennej po lewej stronie znaku = zostanie przypisana wartość wyrażenia podanego po prawej stronie.
- Komunikaty i wyniki wyprowadzamy na ekran monitora, stosując polecenie `cout <<`.

Pytania i polecenia

1. Przedstaw i omów na konkretnym przykładzie etapy tworzenia programu komputerowego.
2. Jaka jest ogólna struktura programu w języku C++?
3. Czym jest komórka pamięci?

4. Na czym polega deklaracja zmiennych w programie komputerowym? Wyjaśnij sposób deklarowania zmiennych w języku C++.
5. Przedstaw na przykładzie sposób wprowadzania danych z klawiatury.
6. Do czego służy instrukcja przypisania?
7. Jaka jest rola zapisów `endl` i `"\n"` umieszczonych w programie?
8. Wyjaśnij zapisy:
   a. `int liczba1, liczba2;`
   b. `float iloraz;`
   c. `cin >> liczba1;`
   d. `iloraz = liczba1 / liczba2;`
   e. `cout << "Iloraz wynosi: " << iloraz << endl;`
9. Wskaż nieprawidłowo zapisane instrukcje. Wyjaśnij, na czym polegają błędy:
   a. `cin >> 78;`
   b. `cin >> droga >> czas;`
   c. `cout >> "Podaj liczbę: ";`
   d. `cout << "Suma wynosi: " << a + b;`



Zadania

Uwagi:

- W edytorze języka C++ można kopiować (**Edycja**/**Kopiuj**) i wklejać (**Edycja**/**Wklej**) fragmenty kodu programu – w tym samym dokumencie i pomiędzy dokumentami (tak jak w edytorze tekstu). Ta możliwość jest przydatna, gdy ponownie wykorzystujemy te same lub podobne fragmenty kodu.
- Staraj się dodawać do programu komunikaty (również komentarze) oraz pisać przejrzyste programy.
- Każdy program należy skompilować, uruchomić i przetestować dla różnych danych, nawet jeśli w zadaniu nie ma takiego polecenia.

1. Napisz specyfikację zadania i program obliczający pole powierzchni prostokąta dla danych długości boków *a* i *b*, wprowadzanych z klawiatury. Po uruchomieniu programu na ekranie powinny pojawiać się w kolejnych wierszach komunikaty: „Podaj dlugosc boku a:", „Podaj dlugosc boku b:". Po podaniu danych w trzecim wierszu powinny wyświetlić się: napis „Pole prostokąta wynosi" oraz wartość pola. Zapisz program w pliku pod nazwą *Pole*.
2. Napisz specyfikację zadania i program realizujący algorytm obliczania kwadratu i sześcianu wprowadzanej z klawiatury liczby rzeczywistej. Zapisz program w pliku pod nazwą *Potegi*.
3. Napisz program obliczający drogę *S* przebytą w czasie *t* przez pojazd poruszający się ze średnią prędkością *v*. Zapisz program w pliku pod nazwą *Droga*.
   Wskazówka: Zadanie wykonaj na podstawie specyfikacji zapisanej w ćwiczeniu 2., punkt 1. z tematu C1.
4. Przepisz program pokazany na rysunku 10. Zapisz go w pliku pod nazwą *Test* i skompiluj. Po wprowadzeniu z klawiatury liczby program powinien wyświetlić na ekranie liczbę o 2 większą od wprowadzonej. Jakie błędy pojawiły się w zapisie programu? Zmodyfikuj program, aby działał poprawnie. Zapisz ponownie plik, skompiluj program, uruchom i sprawdź jego działanie.

[*] TC1_z4.cpp

```
 1  #include <iostream>
 2  using namespace std;
 3
 4  int main()
 5  {
 6      int liczba1;
 7      cin >> liczba1
 8      cout << liczba2
 9      liczba2 = liczba1 + 2;
10      return 0;
11  }
```

Rys. 10. *Błędny program – zadanie 4.*

5. Napisz specyfikację i program do zadania: *Dane są: suma długości przekątnych rombu i długość jednego boku. Oblicz długości przekątnych rombu*. Zapisz program w pliku pod nazwą *Romb*.
6. Napisz specyfikację i program do zadania: *Jaki procent liczby* a *stanowi liczba* b*?* Zapisz program w pliku pod nazwą *Procenty*.
7. Napisz program obliczający wysokość trójkąta, gdy podane są jego pole i podstawa. Zapisz program w pliku pod nazwą *Trojkat*.
8. Napisz program obliczający, ile litrów wody spadło na plac o powierzchni P m^2, jeśli pokryła go warstwa wody o grubości d milimetrów. Zapisz program w pliku pod nazwą *Deszcz*.

Dla zainteresowanych

9. Napisz specyfikację zadania i opisz rozwiązanie: *Dany jest średni wiek trzech osób: S. Najmłodsza osoba ma* x *lat, najstarsza:* y. *Ile lat ma trzecia osoba? S, x, y to liczby naturalne różne od zera*. Napisz program na podstawie zapisanej specyfikacji. Przetestuj program dla następujących wartości zmiennych (S, x, y): (15, 10, 20), (27, 20, 40), (24, 14, 33). Dodaj kilka swoich propozycji danych. Jakie powinny być wartości zmiennych: S oraz x i y, aby otrzymany wynik był sensowny? Uwzględnij te warunki w opisie algorytmu.
10. Dane są suma i iloczyn dwóch liczb całkowitych. Napisz specyfikację zadania i program znajdujący te liczby. Zapisz program w pliku pod nazwą *Liczby*.
11. Opisz wymyślony samodzielnie problem podobnie jak w punkcie 4. tego tematu. Ułóż zadanie, zapisz specyfikację i przygotuj rozwiązanie podobnie jak w przykładzie 5. Na koniec napisz program realizujący to zadanie. Zapisz program w pliku pod nazwą *Problem*.

Temat C3. Tworzenie programów w języku Python

1. Środowisko programistyczne języka Python
   1.1. Tryb interaktywny i skryptowy
   1.2. Pisanie programu w trybie skryptowym
2. Stosowanie zmiennych
   2.1. Nazywanie zmiennych
   2.2. Nadawanie zmiennym wartości
   2.3. Wykonywanie obliczeń
3. Wyprowadzanie komunikatów i wyników na ekran monitora
4. Zapisywanie rozwiązania problemu w języku Python

Warto powtórzyć

1. Jaką funkcję pełni pamięć operacyjna (RAM)?
2. Jakie informacje należy umieścić w specyfikacji zadania (problemu)?
3. Wyjaśnij pojęcia: *język programowania*, *program komputerowy*, *słowo kluczowe*.
4. Dlaczego o języku programowania mówimy, że jest językiem formalnym?
5. Czym jest kod źródłowy programu?
6. Z czego powinna wynikać kolejność instrukcji programu?
7. Na czym polega interpretacja programu?
8. Dlaczego komputer nie wykona błędnie napisanych instrukcji?
9. Czym jest typ zmiennej?

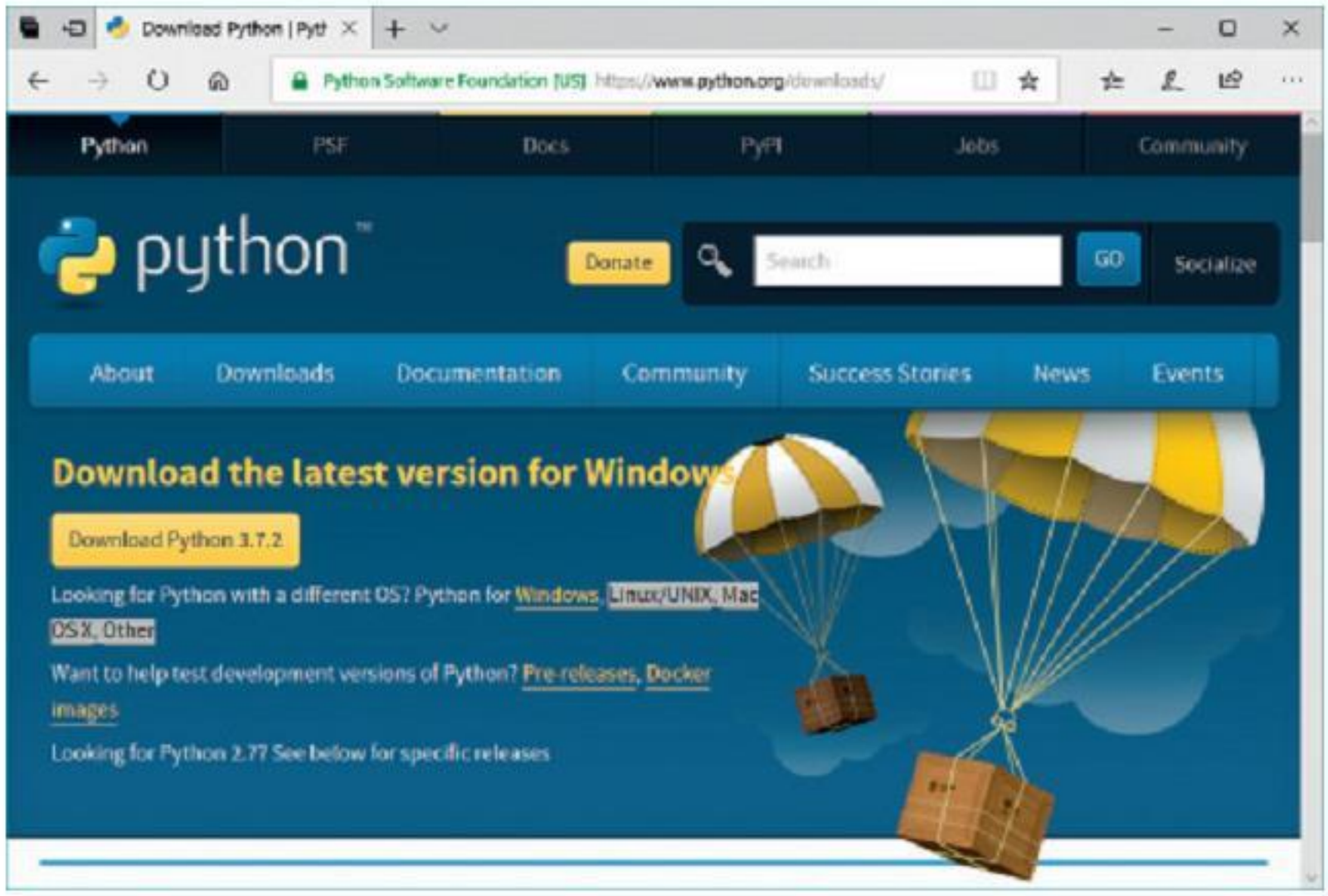
1. Środowisko programistyczne języka Python

Aby pisać programy w języku Python, należy zainstalować darmową aplikację Python, w skład której wchodzi **zintegrowane środowisko programistyczne IDLE**.

IDLE umożliwia tworzenie programów w języku Python, zapisywanie ich i uruchamianie. Zintegrowane środowisko programistyczne IDLE zawiera **powłokę Pythona** (**Python Shell**), **edytor kodu źródłowego** i **interpreter** oraz inne narzędzia wspomagające programowanie.

W tym podręczniku będziemy korzystać z aplikacji Python 3. Po zainstalowaniu aplikacji możemy ją uruchomić i rozpocząć programowanie.

Aby uruchomić IDLE, należy z menu **Start** wybrać program **Python 3.7/IDLE (Python 3.7 32-bit)**. Otworzy się okno programu powłoka Pythona (Python Shell).

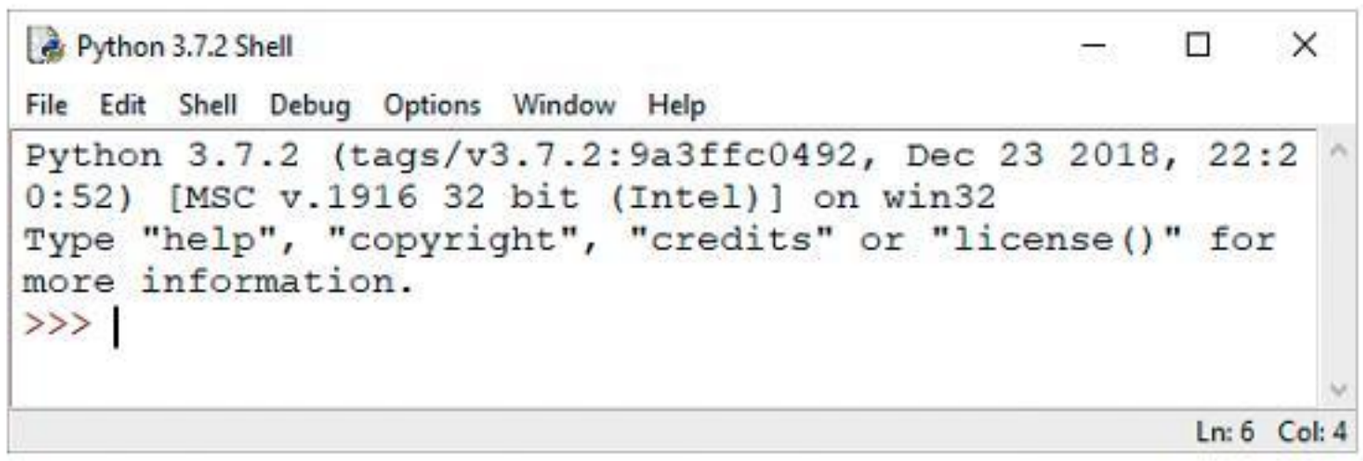


Rys. 1. *Oficjalna strona internetowa służąca m.in. do pobierania programu Python:* https://www.python.org/downloads/

1.1. Tryb interaktywny i skryptowy

Okno powłoki Pythona umożliwia pracę w dwóch trybach: **interaktywnym** i **skryptowym**.

Powłoka Pythona uruchamia się w trybie interaktywnym. Po znaku zachęty >>> widzimy migający kursor. W tym miejscu możemy napisać instrukcję lub wyrażenie. Po naciśnięciu klawisza **Enter** instrukcja zostanie wykonana, a wyrażenie obliczone. Tryb interaktywny widzieliśmy w poznanych w szkole podstawowej językach Logo i Scratch.



Rys. 2. *Okno powłoki Pythona*

Ćwiczenie 1. Sprawdzamy działanie trybu interaktywnego

1. Uruchom środowisko IDLE.
2. W oknie powłoki Pythona sprawdź działanie trybu interaktywnego. Wpisz po znaku zachęty:
   a. `2345 + 564` i naciśnij **Enter**,
   b. `print("Pracuję w trybie interaktywnym")` i naciśnij **Enter**.



W **trybie interaktywnym** możemy sprawdzić działanie jednego polecenia, np. gdy chcemy od razu zobaczyć wynik obliczeń czy sprawdzić działanie konkretnej instrukcji. Natomiast do tworzenia programów w języku Python służy **tryb skryptowy**.

W językach kompilowanych (takich jak C++) musimy napisać cały program, zapisać go w pliku, skompilować i dopiero po poprawnej kompilacji uruchomić. Jeśli kompilator znajdzie w programie błędy, wskaże je. Wtedy program należy poprawić, zapisać, ponownie skompilować, a następnie uruchomić.

W językach interpretowanych, takich jak Python, możemy napisać program składający się z wielu poleceń (tworząc tzw. skrypt), zapisać go w pliku tekstowym, a następnie uruchomić, używając interpretera. Mówimy wówczas, że pracujemy w trybie skryptowym.

1.2. Pisanie programu w trybie skryptowym

W programowaniu w języku Python, jak w każdym języku, bardzo istotna jest prawidłowa kolejność zapisu poleceń.

W języku Python bloki instrukcji wyodrębnia się inaczej niż w innych językach. Nie stosuje się nawiasów ani słów kluczowych, tylko wcięcia, składające się z przynajmniej jednej spacji (przyjęte jest wcięcie składające się z czterech spacji). Bloki instrukcji wykorzystamy w tematach C4 i C5, gdy będziemy omawiać instrukcje warunkowe i iteracyjne. Jeśli nasz program składa się z instrukcji, które nie wchodzą w skład bloku, piszemy je kolejno jedna pod drugą – bez wcięć (przykład takiego programu pokazano na rys. 5. w temacie C1 i na rys. 7a w tym temacie).

Komentarze w języku Python rozpoczyna się znakiem #. Tekst po znaku # do końca wiersza jest ignorowany przez interpreter.

Do trybu skryptowego możemy przejść z trybu interaktywnego z okna powłoki Pythona. Program piszemy w edytorze kodu źródłowego, zapisujemy w pliku na dysku, a następnie uruchamiamy (poprzez wywołanie interpretera Pythona).

Etapy tworzenia programu w języku Python

1. W oknie powłoki Pythona (Python Shell) wybieramy opcję **File/New File** – otworzy się okno **edytora kodu źródłowego**.
2. W oknie edytora piszemy program (rys. 3a).
3. Zapisujemy program w pliku (w języku Python z rozszerzeniem *py*), wybierając w oknie edytora kodu źródłowego opcję **File/Save As**.
4. Uruchamiamy program, wybierając w oknie edytora opcję **Run/Run Module** (lub naciskając klawisz **F5**) – wynik działania programu pojawi się w oknie powłoki Pythona (rys. 3b).
5. Jeżeli wystąpiły błędy, należy je poprawić (interpreter wskazuje prawdopodobne miejsce popełnienia błędu – rys. 4a i 4b) i ponownie przejść do kroku 3.

Instrukcje języka Python, podobnie jak innych języków programowania, realizują takie czynności, jak: wprowadzanie danych, wyprowadzanie wyników, wykonywa-

nie obliczeń, sprawdzanie warunków czy powtarzanie operacji. Poznamy i zastosujemy te instrukcje w ćwiczeniach i zadaniach w tematach C3, C4 i C5.



Rys. 3a. *Okno edytora kodu źródłowego Pythona z programem źródłowym*

Dobra rada

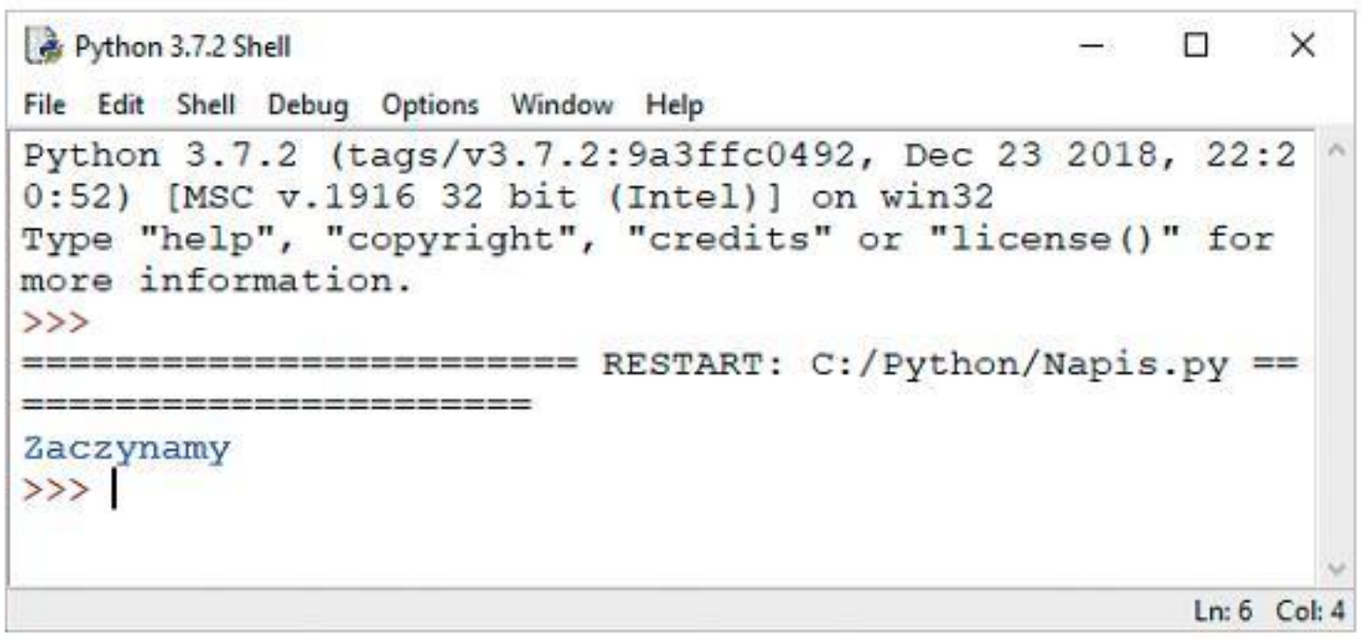
Staraj się samodzielnie poprawiać program do momentu, gdy da się go uruchomić.
W nauce programowania umiejętność wyszukiwania i poprawiania błędów w programie jest bardzo ważna.

Ćwiczenie 2. Poznajemy etapy tworzenia programu w języku Python

1. Utwórz nowy plik źródłowy i napisz program wyświetlający na ekranie napis „Zaczynamy" – przepisz instrukcję pokazaną na rysunku 3a.
2. Zapisz program w pliku pod nazwą *Napis*.
3. Uruchom program.
4. Jeśli interpreter wykrył błędy, popraw je. Zapisz plik pod tą samą nazwą i ponownie uruchom.

Aby zmienić rozmiar czcionki, należy w oknie powłoki Pythona wybrać opcję **Options/Configure IDLE**.

W trakcie wykonywania programu interpreter rozpoznaje polecenia języka Python i zamienia je na instrukcje wewnętrznego języka procesora. Po wykonaniu programu wyniki pojawią się na wybranym urządzeniu zewnętrznym, np. na ekranie monitora (tu w oknie powłoki Pythona – rys. 3b).

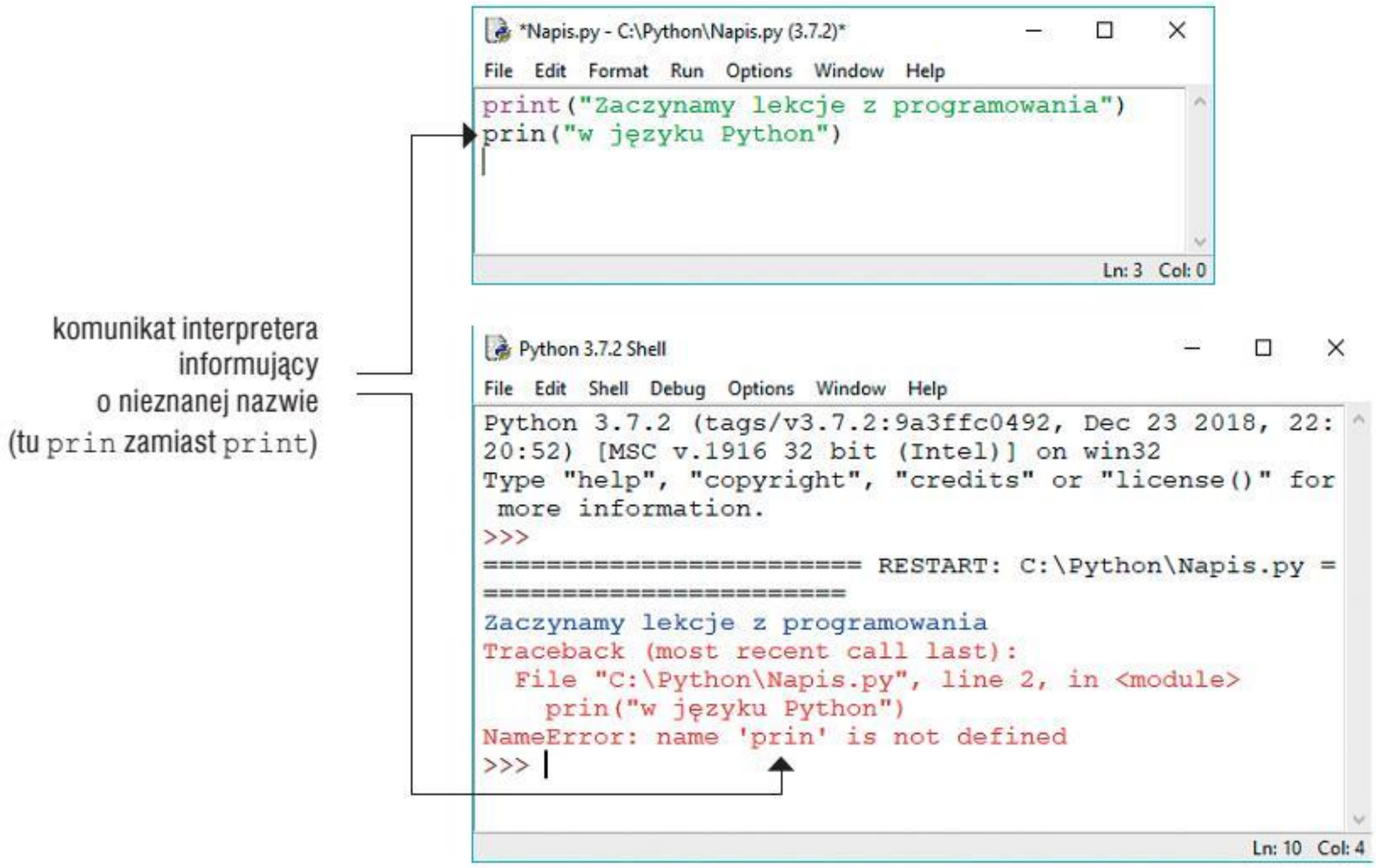


Rys. 3b. *Wynik działania programu – ćwiczenie 2.*

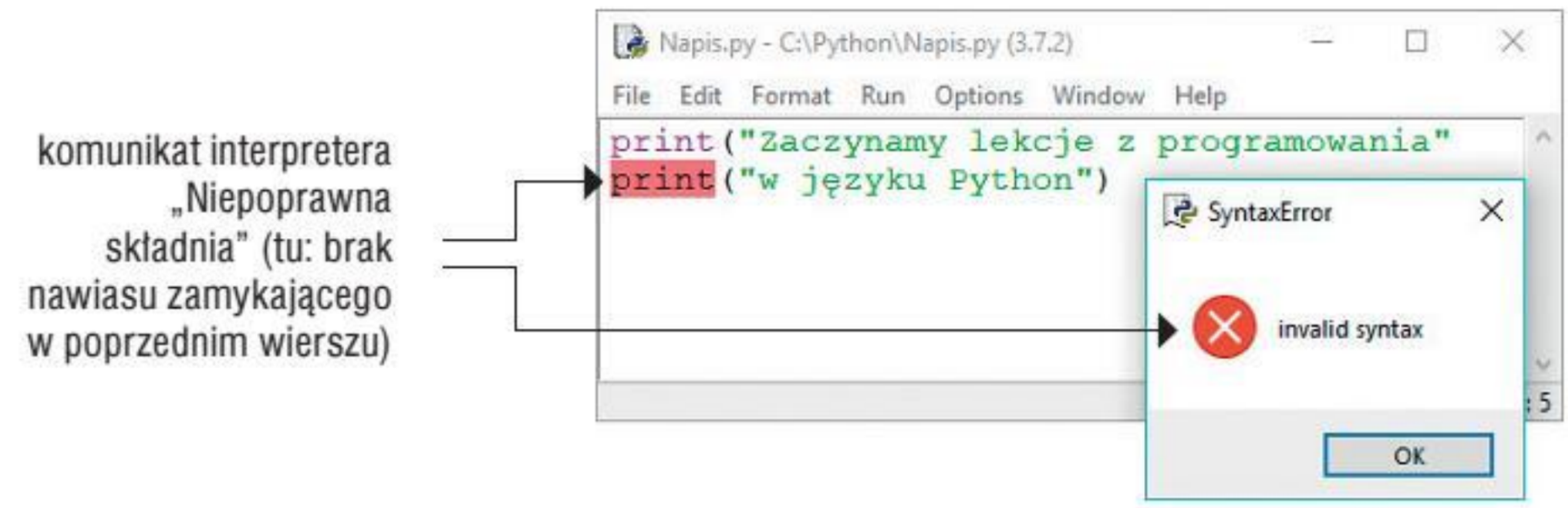
Uwaga

W nazwach plików, w których zapisujemy programy tworzone w języku Python, nie będziemy stosować polskich znaków diakrytycznych i spacji (w nazwach wieloczłonowych użyjemy znaku podkreślenia).

W odróżnieniu od kompilatora, **interpreter** próbuje wykonać program instrukcja po instrukcji. Jeżeli w programie wystąpi błąd, interpreter przerwie wykonanie programu i wyświetli komunikat o błędzie (rys. 4a i 4b). Należy poprawić wskazany błąd, ponownie zapisać program w pliku i uruchomić program.



Rys. 4a. *Okno powłoki Pythona z komunikatem interpretera*



Rys. 4b. *Okno powłoki Pythona z komunikatem interpretera*

Jeśli interpreter wykryje błąd, należy poprawić program, ponownie go zapisać i uruchomić.

Ćwiczenie 3. Modyfikujemy program

1. Zmodyfikuj program utworzony w ćwiczeniu 2., wyświetlając w tym samym wierszu dalszą część napisu: „lekcje z programowania", a w drugim wierszu – napis „w języku Python".
2. Jeśli interpreter wykrył błędy, popraw je. Zapisz plik pod tą samą nazwą. Ponownie uruchom program.

Wskazówki: Na rysunkach 4a i 4b pokazano przykładowe błędy wykryte przez interpreter. Oba błędy są w pierwszym wierszu programu, ale na rysunku 4b interpreter podświetlił drugi wiersz, ponieważ brakuje nawiasu zamykającego na końcu poprzedniego wiersza.

2. Stosowanie zmiennych

Chcemy obliczyć iloczyn dwóch liczb całkowitych. Jak napisać program w języku Python umożliwiający wprowadzenie dwóch liczb całkowitych z klawiatury, obliczający ich iloczyn i wyprowadzający wynik obliczeń na ekran?

Program, który utworzyliśmy w ćwiczeniach 2. i 3., wyświetlał na ekranie napis – nie używaliśmy w nim zmiennych i nie wykonywaliśmy obliczeń. Struktura kolejnych programów będzie bardziej rozbudowana, m.in. dodamy **zmienne**.

2.1. Nazywanie zmiennych

W wielu językach programowania (np. w języku C++) przed użyciem zmiennej należy ją zadeklarować, czyli określić jej nazwę i typ. W języku Python nie deklaruje się zmiennych. Nie musimy wcześniej określać typu zmiennej, czyli rodzaju przechowywanych w niej danych (np. liczby całkowite, rzeczywiste, znaki).

Interpreter określa typ zmiennej na podstawie rodzaju danych, jaki jej przypiszemy. Nie ustanawia jednak typu zmiennej na stałe, ponieważ przy kolejnym przypisaniu może się on zmienić. Na przykład, jeśli na początku programu przypiszemy zmiennej `liczba` wartość całkowitą (34), to jej typ jest całkowity. Jeśli jednak w dalszej części programu przypiszemy jej wartość rzeczywistą (np. 45,8, którą w języku Python zapisujemy z kropką: 45.8), to zmienna `liczba` będzie typu rzeczywistego.

Zasady dotyczące nazw zmiennych w języku Python

1. Wielkie i małe litery w nazwach traktowane są odmiennie (np. `suma` i `Suma` oznaczać będą różne zmienne). Pisząc program, należy zwracać uwagę na poprawne używanie małych i wielkich liter.
2. W nazwach zmiennych powinno się używać liter, znaku podkreślenia i cyfr. Nazwa nie może zaczynać się od cyfry. Przyjęte jest stosowanie małych liter i niestosowanie polskich liter.
3. W nazwach zmiennych nie wolno stosować spacji. W przypadku nazw kilkuczłonowych zamiast spacji stosujemy znak podkreślenia.
4. Należy nadawać nazwy, które określają znaczenie danej zmiennej, np. `suma`, `liczba_elementow`.

W kilku kolejnych ćwiczeniach napiszemy program obliczający iloczyn dwóch liczb całkowitych wprowadzanych z klawiatury. Zaczniemy od uzupełnienia specyfikacji zadania.

Ćwiczenie 4. Zapisujemy dane i wyniki do zadania

Wzorując się na opisie podanym w przykładzie 6. z tematu C1, zapisz w zeszycie przedmiotowym dane i wyniki do zadania: *Napisz w wybranym języku programowania program umożliwiający wprowadzenie z klawiatury dwóch liczb całkowitych, obliczający ich iloczyn i wyprowadzający wynik obliczeń na ekran. Przed wprowadzeniem pierwszej danej wyświetlaj napis „Podaj pierwszą liczbę”, przed wprowadzaniem drugiej – „Podaj drugą liczbę”, a przed wyprowadzeniem wyniku – „Iloczyn wynosi:”.*

2.2. Nadawanie zmiennym wartości

Zmiennej stosowanej w programie możemy nadać konkretną wartość za pomocą **instrukcji przypisania**. W instrukcji przypisania zmiennej podanej po lewej stronie instrukcji zostanie przypisana obliczona przez komputer wartość wyrażenia znajdującego się po prawej stronie instrukcji.

Instrukcja przypisania	`zmienna = wyrażenie`	**Python**

Wyrażeniem mogą być m.in. konkretna wartość, zmienna, wyrażenie algebraiczne.

Przykład 1. Stosowanie instrukcji przypisania

`rok = 2019`
– zmiennej `rok` przypisujemy wartość, która jest liczbą całkowitą, czyli zmienna `rok` będzie typu całkowitego (`int`),
`a = 7.58`
– zmiennej `a` przypisujemy wartość, która jest liczbą rzeczywistą, czyli zmienna `a` będzie typu rzeczywistego (`float`),
`x = y`
– zmiennej `x` przypisujemy wartość zmiennej `y`.
`obwod = 2 * a + 2 * b`
– zmiennej `obwod` przypisujemy wartość wyrażenia `2 * a + 2 * b`; jeśli wynik obliczeń będzie liczbą całkowitą, interpreter określi dla zmiennej `obwod` typ całkowity; jeśli wynik będzie liczbą dziesiętną, interpreter określi typ zmiennej jako rzeczywisty.

Zmiennej stosowanej w programie możemy również nadać wartość za pomocą instrukcji przypisania, wprowadzając wartość z klawiatury w trakcie działania programu. W tym celu stosujemy **instrukcję wejścia** – funkcję `input()`.

Argumentem funkcji `input()` (podanym w nawiasach) jest tekst (ciąg znaków, łańcuch). Tekst ten pojawi się na ekranie jako swego rodzaju „zaproszenie" do wpisania ciągu znaków z klawiatury. Jeśli użyjemy funkcji `input()` w instrukcji przypisania, wpisany przez użytkownika ciąg znaków zostanie zapamiętany w zmiennej podanej po lewej stronie.

Wprowadzanie danych z klawiatury	`zmienna = input(tekst_zachęty)`	**Python**

Aby dana wprowadzona za pomocą instrukcji `input()` została zapamiętana jako liczba, musimy dodać instrukcję, która zamieni ciąg znaków na liczbę: `int()` – w przypadku liczb całkowitych, `float()` – w przypadku liczb rzeczywistych.

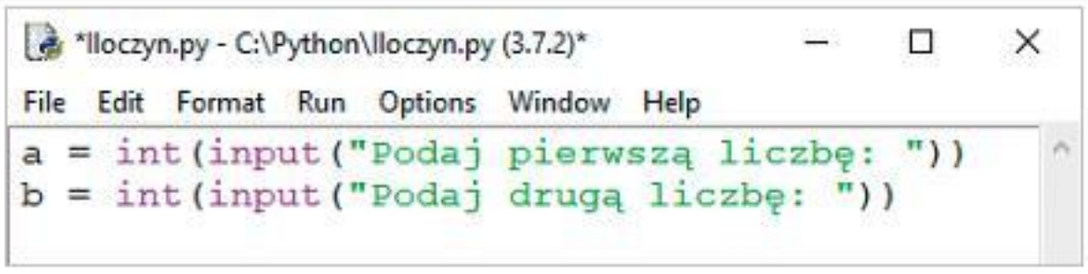
Przykład 2. Wprowadzanie danych z klawiatury – stosowanie funkcji `input()`

`a = input("Wprowadź liczbę: ")`
– jeśli wpiszemy z klawiatury liczbę 346, w zmiennej `a` zostanie zapamiętany ciąg znaków „346”, a nie liczba 346,
`a = int(input("Wprowadź liczbę: "))`
– jeśli wpiszemy z klawiatury liczbę 346, w zmiennej `a` zostanie zapamiętana liczba całkowita 346,
`srednia = float(input("podaj średnią ocen: "))`
– jeśli wpiszemy z klawiatury liczbę 4.3, w zmiennej `srednia` zostanie zapamiętana liczba rzeczywista 4,3,
`miasto = input("Wprowadź nazwę miasta: ")`
– jeśli wpiszemy z klawiatury nazwę miasta „Wrocław”, w zmiennej `miasto` zostanie zapamiętany ciąg znaków „Wrocław”.

W ćwiczeniach 5–7 napiszemy program obliczający iloczyn dwóch liczb na podstawie specyfikacji zadania zapisanego w ćwiczeniu 4.

Ćwiczenie 5. Wprowadzamy dane z klawiatury

1. Otwórz nowy plik źródłowy. Umieść polecenia wprowadzania danych z klawiatury, zgodnie z programem pokazanym na rysunku 5.
2. Zapisz program w pliku pod nazwą *Iloczyn* i uruchom program. Jeśli interpreter wykrył błędy, popraw je, ponownie zapisz plik i uruchom.



Rys. 5. *Program, w którym wprowadza się dane z klawiatury – ćwiczenie 5.*

2.3. Wykonywanie obliczeń

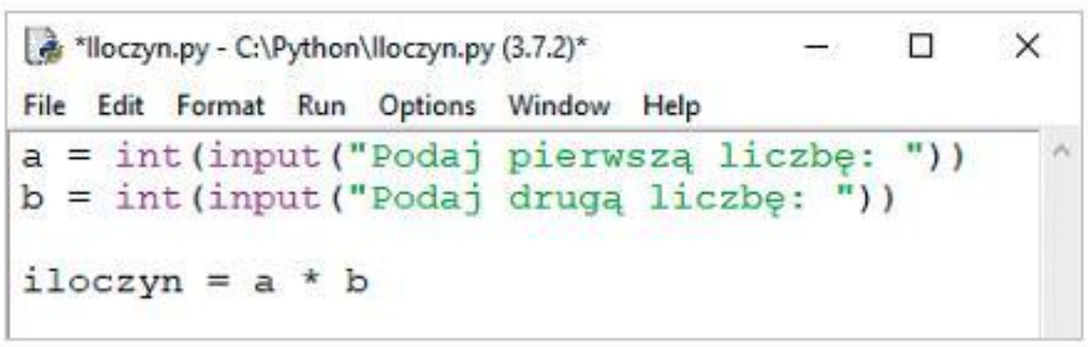
Aby zapisać w programie obliczenia, możemy skorzystać z instrukcji przypisania.

W języku Python stosujemy podstawowe operatory arytmetyczne: + (dodawania), – (odejmowania), * (mnożenia), // (dzielenia całkowitego), / (dzielenia zmiennoprzecinkowego) oraz % (reszty z dzielenia dwóch liczb całkowitych).

Ćwiczenie 6. Stosujemy instrukcję przypisania do zapisania obliczeń

1. Otwórz plik *Iloczyn* zapisany w ćwiczeniu 5.
2. Dodaj obliczanie iloczynu tak jak pokazano na rysunku 6.
3. Zapisz plik pod tą samą nazwą i uruchom program. Dlaczego nie widzisz wyniku obliczeń na ekranie? Uzasadnij odpowiedź.

Wskazówka: Aby otworzyć ponownie plik (jeśli zamkniemy okno edytora z programem), należy w oknie powłoki wybrać opcję **File/Open**.



Rys. 6. *Program obliczający iloczyn dwóch liczb wprowadzanych z klawiatury – ćwiczenie 6.*

Operator	Działanie	Przykład instrukcji przypisania	Wynik działania dla danych: a = 11 i b = 4
+	dodawanie	`suma = a + b`	11 + 4 = 15
–	odejmowanie	`roznica = a - b`	11 – 4 = 7
*	mnożenie	`iloczyn = a * b`	11 * 4 = 44
//	dzielenie całkowite (z zaokrągleniem części ułamkowej „w dół” – do największej liczby całkowitej mniejszej od wyniku dzielenia)	`iloraz = a // b`	11 // 4 = 2
/	dzielenie zmiennoprzecinkowe (z zachowaniem części ułamkowej)	`iloraz = a / b`	11 / 4 = 2.75
%	obliczenie reszty z dzielenia dwóch liczb całkowitych	`reszta = a % b`	11 % 4 = 3

Tabela 1. *Podstawowe operatory arytmetyczne w języku Python*

3. Wyprowadzanie komunikatów i wyników na ekran monitora

W ćwiczeniach 2. i 3. wyprowadzaliśmy komunikaty na ekran monitora, używając polecenia `print()`. Za pomocą tego polecenia można wyprowadzać również wyniki obliczeń.

Wyprowadzanie wyników i komunikatów na ekran monitora	`print(`*wartość*`)`	**Python**

W języku Python `print()` jest **funkcją**, która wyświetla na ekranie wartość zapisaną w nawiasach i umieszcza kursor w nowym wierszu.

Wartością umieszczoną wewnątrz nawiasów może być zmienna, wyrażenie, a także napis. Napis umieszczany w programie, który ma zostać wyświetlony na ekranie monitora, musi być ujęty w górne cudzysłowy podwójne `""` lub pojedyncze `''`. Funkcja `print()` pozwala łączyć wyświetlanie napisów, wyrażeń i zmiennych. Poszczególne wartości, występujące jako parametry funkcji `print()`, należy rozdzielić przecinkami.

Dobra rada

Mimo że interpreter pomija dodatkowe spacje, kod programu powinien być napisany czytelnie, przejrzyście i jednolicie (rys. 7a). Innej osobie łatwiej będzie zrozumieć program napisany w dobrym stylu.

Przykład 3. Wyprowadzanie komunikatów i wyników

```
print("Zaczynamy lekcje z programowania")
print(p)
print(a + b)
print(23 + 89)
print("Obwód =", 2 * a + 2 * b)
print("Obwód wynosi:", obwod)
print("Dla boku o długości", a, "pole kwadratu wynosi:", a * a)
```

Umieszczenie sekwencji znaków `\n` w tekście wyprowadzanym za pomocą instrukcji `print()` wymusza przejście do nowego wiersza.
Instrukcja:

```
print("Pole prostokąta wynosi:", "\nbok1 * bok2 = ", pole)
```

i instrukcje:

```
print("Pole prostokąta wynosi:")
print("bok1 * bok2 =", pole)
```

wyświetlą w taki sam sposób te same wyniki

```
Pole prostokąta wynosi:
bok1 * bok2 = 96
```

.

Iloczyn.py - C:\Python\Iloczyn.py (3.7.2)

File Edit Format Run Options Window Help

```
a = int(input("Podaj pierwszą liczbę: "))
b = int(input("Podaj drugą liczbę: "))

iloczyn = a * b

print("Iloczyn wynosi:", iloczyn)
```

LEPIEJ

Rys. 7a. *Program napisany w uporządkowany sposób – ćwiczenie 7.*

Iloczyn.py - C:\Python\Iloczyn.py (3.7.2)

File Edit Format Run Options Window Help

```
a= int(input("Podaj pierwszą liczbę: ") )
b =int  (input("Podaj drugą liczbę: ") )
iloczyn=a* b
print ("Iloczyn wynosi:",      iloczyn)
```

GORZEJ

Rys. 7b. *Program napisany mniej przejrzyście – ćwiczenie 7.*

Ćwiczenie 7. Wyprowadzamy napisy i wynik na ekran

1. Otwórz plik *Iloczyn* zapisany w ćwiczeniu 6.
2. Dodaj wyświetlanie komunikatu i wyniku podobnie jak pokazano na rysunku 7a.
3. Zapisz plik pod tą samą nazwą i uruchom program.
4. Co sądzisz o programach pokazanych na rysunkach 7a i 7b? Wskaż kilka zasadniczych różnic.

Ćwiczenie 8. Testujemy działanie programu obliczającego średnią dla danych różnego typu

Uwaga

W większości języków programowania, w tym w języku Python, część dziesiętną liczby zapisujemy po kropce, a nie po przecinku, jak na matematyce. W ten sam sposób liczby dziesiętne są również wyświetlane na ekranie.

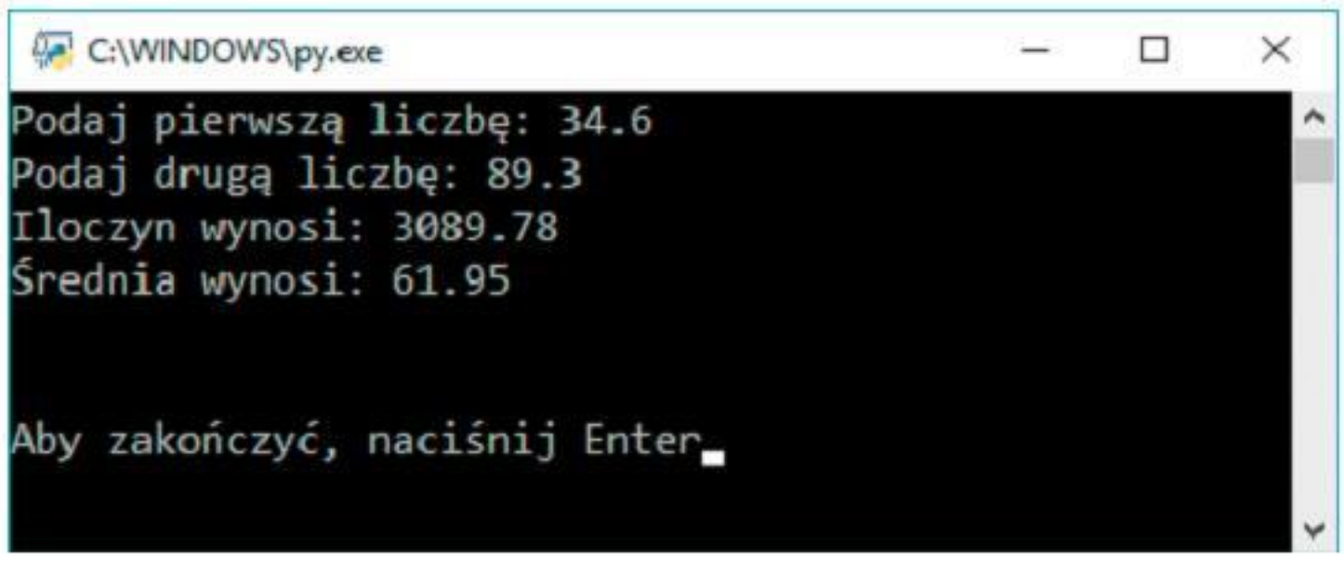
1. Do programu zapisanego w ćwiczeniu 7. dodaj obliczanie średniej arytmetycznej liczb *a* i *b*. Wynik wyprowadź na ekran. Dodaj odpowiednie komunikaty.
2. Zapisz program w pliku pod nazwą *Srednia_c*. Uruchom program. Przetestuj program dla par liczb (wartości zmiennych *a* i *b*): (2; 3), (14; 8); (7; 6). Dodatkowo przetestuj program dla pary liczb dziesiętnych (23,3; 12,9). Dlaczego program nie zadziałał?
3. Zmień w instrukcji przypisania typ wprowadzanych danych *a* i *b* na `float`. Zapisz program w pliku pod nazwą *Srednia_r*. Uruchom program. Przetestuj program dla tych samych danych. Czy teraz program zadziałał za każdym razem?
4. Przetestuj program *Srednia_r* dla liczb: (342,3; 25,7) (3763,82; 109,87).

Wskazówka: Liczby dziesiętne wprowadzaj z kropką, a nie z przecinkiem.

Program możemy również uruchomić, klikając dwukrotnie nazwę pliku z zapisanym programem w Eksploratorze plików – otworzy się okno, w którym zobaczymy wynik działania programu (rys. 8.). Aby program został wykonany, niezbędny jest interpreter Pythona.

Aby okno pokazane na rysunku 8. nie zamknęło się automatycznie po wyświetleniu napisu, możemy napisać na końcu programu instrukcję oczekiwania na naciśnięcie klawisza **Enter**:
`input("\n\nAby zakończyć, naciśnij Enter").`

W tym przypadku wartość zwracana przez funkcję `input()` jest pomijana (nie jest przypisywana żadnej zmiennej). Umieszczenie zapisu `\n\n` oznacza wstawienie dwóch pustych wierszy przed komunikatem.



***Rys. 8.** Wynik działania programu dla zmiennych typu rzeczywistego – ćwiczenia 8. i 9.*

Ćwiczenie 9. Uruchamiamy program, klikając jego nazwę

1. Otwórz plik *Srednia_r* zapisany w ćwiczeniu 8.
2. Dodaj na końcu programu polecenie oczekiwania na naciśnięcie klawisza **Enter**. Zapisz plik pod tą samą nazwą.
3. Zamknij okno edytora i uruchom program, klikając dwukrotnie jego nazwę lub ikonę w oknie Eksploratora plików. Wynik działania programu pokazano na rysunku 8.

Jeśli zamknęliśmy okno powłoki Pythona, a chcemy edytować program, można z menu kontekstowego nazwy pliku w oknie Eksploratora plików wybrać polecenie **Edit with IDLE/ Edit with IDLE 3.7 (32-bit)** – rys. 9.



Rys. 9. *Otwieranie okna edytora kodu źródłowego z programem zapisanym w pliku*

4. Zapisywanie rozwiązania problemu w języku Python

Chcesz przygotować tradycyjną sałatkę jarzynową na imprezę urodzinową. W Internecie są przepisy, ale w proporcjach dla dwóch, pięciu czy dziesięciu osób. Nie możesz znaleźć przepisu dla trzynastu osób – uczestników spotkania. Jak napisać program w języku Python, który policzy, ile potrzeba danego składnika sałatki w zależności od liczby gości, i wyświetli wynik na ekranie?

Przykład 4. Sformułowanie zadania, opis specyfikacji i projekt rozwiązania

Zadanie: Napisz program obliczający, ile potrzeba danego składnika sałatki dla *o2* osób, gdy znasz ilość składnika dla *o1* osób.

Dane: liczba całkowita $o1 > 0$ oznaczająca liczbę osób uwzględnioną w przepisie, liczba rzeczywista $s1 > 0$, określająca ilość składnika dla *o1* osób, liczba całkowita $o2 > 0$, oznaczająca liczbę uczestników spotkania.

Wynik: liczba rzeczywista $s2 > 0$ oznaczająca ilość składnika dla *o2* osób.

W rozwiązaniu skorzystamy z zależności:
jeśli
s1 ⟶ dla *o1* osób
s2 ⟶ dla *o2* osób
to

$$s2 = \frac{s1 \cdot o2}{o1}$$

Aby zapisać te obliczenia w języku Python, zastosujemy instrukcję przypisania:

```
s2 = (s1 * o2) / o1
```

Uwaga: Z matematycznego punktu widzenia nawiasy są zbędne – umieściliśmy je w programie dla zwiększenia czytelności.

Ćwiczenie 10. Realizujemy algorytm obliczania ilości składnika sałatki w języku Python

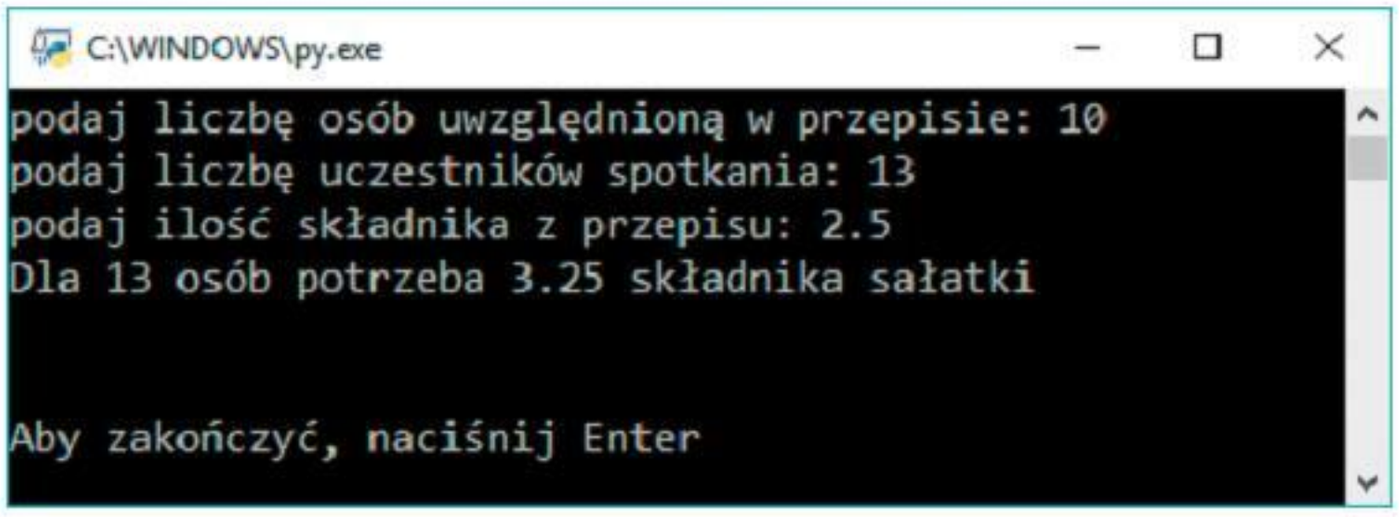
Uwaga

Na razie w programach nie sprawdzamy poprawności danych wprowadzanych przez użytkownika. Zakładamy, że użytkownik podaje poprawne dane. Problem poprawności danych omówimy w temacie C4.

1. Utwórz program realizujący zadanie sformułowane w przykładzie 4. Przed wprowadzaniem zmiennych dodaj komunikaty. Dodaj również komunikat przed wyprowadzeniem wyniku. Komunikaty i wynik wyprowadź na ekran monitora tak jak pokazano na rysunku 10.
2. Zapisz program w pliku pod nazwą *Przepis*. Uruchom program i przetestuj dla następujących trójek liczb (wartości zmiennych *o1*, *o2*, *s1*): (10; 13; 2,5), (3; 12; 0,45), (15; 4; 3,20).
3. Przetestuj program dla trzech innych trójek liczb.

Wskazówka: Aby wyświetlić wyniki, jak pokazano na rysunku 10., należy wpisać instrukcję:

```
print("Dla ", o2, " osób potrzeba ", s2, "składnika sałatki")
```



Rys. 10. *Wynik działania programu – ćwiczenie 10.*

Zauważmy, że program utworzony w ćwiczeniu 10. oblicza ilość jednego składnika sałatki. Jeśli chcemy obliczyć ilość kolejnego składnika, musimy ponownie uruchomić program. W temacie C5 zmodyfikujemy ten program, aby obliczał i wyświetlał na ekranie kolejno ilości wszystkich składników sałatki.

Warto zapamiętać

- Środowisko programistyczne Pythona (IDLE) udostępnia dwa tryby pracy:
  - interaktywny, w którym możemy napisać jedno polecenie i zatwierdzić je klawiszem **Enter**; jeśli polecenie będzie poprawne, zostanie od razu wykonane,
  - skryptowy, w którym możemy napisać program, zapisać go w pliku, a następnie uruchomić; jeśli program będzie poprawny, zostanie wykonany.
- Aby utworzyć program w języku Python, piszemy kod źródłowy w edytorze tekstu, zapisujemy program w pliku i uruchamiamy za pomocą interpretera Pythona.
- W języku Python typ wartości, która jest przypisana zmiennej, określa typ tej zmiennej.
- Zmiennej możemy nadać wartość za pomocą instrukcji przypisania. Po lewej stronie znaku = podajemy nazwę zmiennej, a po prawej:
  - konkretną wartość lub
  - funkcję `input()`, dzięki której użytkownik wprowadzi tę wartość z klawiatury po uruchomieniu programu lub
  - wyrażenie, którego wartość zostanie obliczona w czasie działania programu.
- Aby wyświetlić komunikaty i wyniki na ekranie, stosujemy instrukcję wyjścia – funkcję `print()`.

Pytania i polecenia

1. Wyjaśnij, czym jest tryb interaktywny, a czym tryb skryptowy w środowisku Python.
2. Przedstaw i omów na konkretnym przykładzie etapy tworzenia programu komputerowego w języku Python.
3. Przedstaw na przykładzie sposób wprowadzania danych z klawiatury.
4. Co w języku Python określa typ zmiennej?
5. W jaki sposób możemy nadać wartość zmiennej?
6. Do czego służy funkcja `input()`?
7. Omów na przykładzie sposób wyświetlania napisów i wyników na ekranie monitora.
8. Jaka jest rola zapisu `\n` umieszczonego w programie?
9. Wyjaśnij zapisy:
   a. `bok1 = int(input("podaj długość boku: "))`
   b. `liczba1 = float(input("podaj liczbę: "))`
   c. `iloraz = liczba1 / liczba2`
   d. `print("Iloraz wynosi:", iloraz)`
10. Wyjaśnij, jaka dana zostanie zapamiętana w zmiennej `x` po wykonaniu instrukcji przypisania, gdy z klawiatury wprowadzimy liczbę 258:
    a. `x = int(input("podaj długość"))`
    b. `x = input("podaj długość"))`
11. Wskaż nieprawidłowo zapisane instrukcje. Wyjaśnij, na czym polegają błędy:
    a. `liczba1 = input(int("Podaj liczbę: "))`
    b. `y = 29871`
    c. `print("Suma wynosi: ", a + b)`
    d. `iloraz = liczba1 : liczba2`

Zadania

Uwagi:

- W edytorze języka Python można kopiować (**Edycja/Kopiuj**) i wklejać (**Edycja/Wklej**) fragmenty kodu programu – w tym samym dokumencie i pomiędzy dokumentami (tak jak w edytorze tekstu). Ta możliwość jest przydatna, gdy ponownie wykorzystujemy te same lub podobne fragmenty kodu.
- Staraj się pisać przejrzyste programy oraz dodawać w odpowiednich miejscach komentarze.
- Każdy program należy uruchomić i przetestować dla różnych danych, nawet jeśli w zadaniu nie ma takiego polecenia.

1. Napisz specyfikację zadania i program obliczający pole powierzchni prostokąta dla danych długości boków *a* i *b* wprowadzanych z klawiatury. Po uruchomieniu programu na ekranie powinny pojawiać się w kolejnych wierszach komunikaty: „Podaj długość boku a: ", „Podaj długość boku b: ". Po podaniu danych w trzecim wierszu powinny wyświetlić się: napis „Pole prostokąta wynosi" oraz wartość pola. Zapisz program w pliku pod nazwą *Pole*.
2. Napisz specyfikację zadania i program realizujący algorytm obliczania kwadratu i sześcianu liczby rzeczywistej wprowadzanej z klawiatury. Zapisz program w pliku pod nazwą *Potegi* i uruchom. Przetestuj program dla różnych danych.
3. Napisz specyfikację i program do zadania: *Oblicz drogę* S *przebytą w czasie* t *przez pojazd poruszający się ze średnią prędkością* v. Zapisz program w pliku pod nazwą *Droga*.
   Wskazówka: Zadanie wykonaj na podstawie specyfikacji zapisanej w ćwiczeniu 2., punkt 1. z tematu C1.

4. Przepisz program pokazany na rysunku 11. Zapisz go w pliku pod nazwą *Test* i uruchom. Po wprowadzeniu z klawiatury liczby program powinien wyświetlić na ekranie liczbę o 2 większą. Jakie błędy pojawiły się w zapisie programu? Zmodyfikuj program, aby działał poprawnie. Zapisz plik i uruchom ponownie program.

Rys. 11. *Błędny program – zadanie 4.*

```
liczba1 = int(input("podaj liczbę: "))
print(liczba2)
liczba2 = liczba1 + 2
```

5. Napisz specyfikację i program do zadania: *Dane są: suma długości przekątnych rombu i długość boku. Oblicz długości przekątnych rombu.* Zapisz program w pliku pod nazwą *Romb*.
6. Napisz specyfikację i program do zadania: *Jaki procent liczby* a *stanowi liczba* b? Zapisz program w pliku pod nazwą *Procenty*.
7. Napisz program obliczający wysokość trójkąta, gdy podane są jego pole i podstawa. Zapisz program w pliku pod nazwą *Trojkat*.
8. Napisz program obliczający, ile litrów wody spadło na plac o powierzchni P m^2, jeśli pokryła go warstwa wody o grubości d milimetrów. Zapisz program w pliku pod nazwą *Deszcz*.

Dla zainteresowanych

9. Napisz specyfikację zadania i opisz rozwiązanie: *Dany jest średni wiek trzech osób:* S. *Najmłodsza osoba ma* x *lat, najstarsza:* y. *Ile lat ma trzecia osoba?* S, x, y *to liczby naturalne różne od zera.* Napisz program na podstawie zapisanej specyfikacji. Przetestuj program dla następujących wartości zmiennych (S, x, y): (15; 10; 20), (27; 20; 40), (24; 14; 33). Dodaj kilka swoich propozycji danych. Jakie powinny być wartości zmiennych: S oraz x i y, aby otrzymany wynik był sensowny? Uwzględnij te warunki w opisie algorytmu.
10. Dane są suma i iloczyn dwóch liczb całkowitych. Napisz specyfikację zadania i program znajdujący te liczby. Zapisz program w pliku pod nazwą *Liczby*.
11. Opisz wymyślony samodzielnie problem podobnie jak w punkcie 4. tego tematu. Ułóż zadanie, zapisz specyfikację i przygotuj rozwiązanie podobnie jak w przykładzie 4. Na koniec napisz program realizujący to zadanie. Zapisz program w pliku pod nazwą *Problem*.

Temat C4. Stosowanie instrukcji warunkowych w językach C++ i Python

1. Sytuacje warunkowe
2. Algorytmy z warunkami w językach C++ i Python
   2.1. Proste warunki logiczne
   2.2. Złożone warunki logiczne
3. Sprawdzanie poprawności danych
4. Algorytmy z warunkami zagnieżdżonymi

Warto powtórzyć

1. Wyjaśnij pojęcie *słowo kluczowe*.
2. Na czym polega deklarowanie zmiennych w języku C++?
3. Czy w języku Python deklaruje się zmienne? Uzasadnij odpowiedź.
4. W jaki sposób możemy nadać wartości zmiennym w językach C++ i Python?
5. Do czego stosuje się instrukcję przypisania?
6. Jak wyróżnić blok instrukcji (blok kodu) w językach C++ i Python?

1. Sytuacje warunkowe

Sytuacja warunkowa występuje wtedy, gdy wynik lub dalsze działanie zależą od spełnienia (lub niespełnienia) pewnego warunku. Algorytm zawierający sytuacje warunkowe nazywamy **algorytmem z warunkami** (inaczej: **algorytmem z rozgałęzieniami**).

Sytuacje warunkowe występują w matematyce i fizyce, gdy wykonanie obliczeń zależy od warunku, który muszą spełniać dane, np. powinny być różne od zera lub dodatnie.

Ćwiczenie 1. Podajemy przykłady sytuacji warunkowych

1. Podaj przykład dwóch zadań, w których występują sytuacje warunkowe.
2. Określ, czy w zadaniu „Oblicz liczbę znaków różnych od spacji w dowolnym tekście." (temat C1, ćwiczenie 2., punkt 3.) występuje sytuacja warunkowa. Uzasadnij odpowiedź.

2. Algorytmy z warunkami w językach C++ i Python

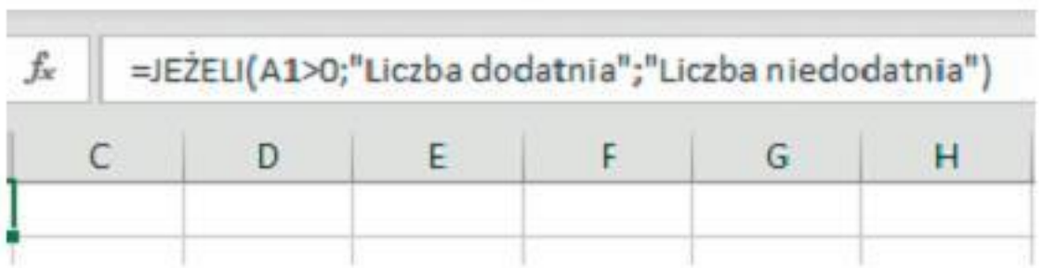
Chcemy napisać program sprawdzający, czy wprowadzona z klawiatury liczba jest dodatnia czy niedodatnia (ujemna albo równa zero). Jak zrealizować ten algorytm w arkuszu kalkulacyjnym? W jaki sposób napisać program realizujący sytuację warunkową w wybranym języku programowania?

2.1. Proste warunki logiczne

Aby w wybranym języku programowania napisać program realizujący algorytm z warunkami, korzystamy z **instrukcji warunkowej**.

Instrukcja warunkowa ma postać i działanie zbliżone do funkcji JEŻELI arkusza kalkulacyjnego:
=JEŻELI(*test_logiczny;wartość_jeżeli_prawda;wartość_jeżeli_fałsz*).

Arkusz kalkulacyjny Excel



Baltie



Scratch



C++

```
if(a > 0)
    cout << "Liczba dodatnia";
else
    cout << "Liczba niedodatnia";
```

Python

```
if a > 0:
    print("Liczba dodatnia")
else:
    print("Liczba niedodatnia")
```

Rys. 1. *Zastosowanie funkcji JEŻELI w arkuszu kalkulacyjnym i instrukcji warunkowej w dydaktycznych środowiskach programowania (Baltie, Scratch) oraz w językach programowania (C++, Python) do realizacji tego samego algorytmu z warunkiem*

Ćwiczenie 2. Analizujemy realizację sytuacji warunkowej w różnych narzędziach

Porównaj zapis sytuacji warunkowej w różnych narzędziach przedstawiony na rysunku 1. Zwróć uwagę na postać instrukcji warunkowej. Wskaż podobieństwa i różnice.



Działanie instrukcji warunkowej jest w większości języków programowania podobne. Sprawdzany jest warunek logiczny (po słowie kluczowym **`if`**) – jeśli jest prawdziwy, wykonywana jest instrukcja (instrukcje) umieszczona (umieszczone) po warunku (*lista_instrukcji1*); jeśli fałszywy – wykonywana jest instrukcja (instrukcje) po słowie **`else`** (*lista_instrukcji2*), a jeśli brak tej części instrukcji – wykonywana jest od razu *kolejna_instrukcja* zapisana po instrukcji warunkowej.

Instrukcja warunkowa	**if**(*warunek*) *lista_instrukcji1;* **else** *lista_instrukcji2;* *kolejna_instrukcja;*	C++
	if *warunek*: *lista_instrukcji1* **else**: *lista_instrukcji2* *kolejna_instrukcja*	Python

Uproszczona postać instrukcji warunkowej	**if**(*warunek*) *lista_instrukcji1;* *kolejna_instrukcja;*	C++
	if *warunek*: *lista_instrukcji1* *kolejna_instrukcja*	Python

Warunek po słowie kluczowym **`if`** może być:

- **prostym warunkiem logicznym**, np.

  `x > 0, a == 9, liczba <= -5, a != 9` C++ i Python

lub

- **złożonym warunkiem logicznym**, np.:

  `x > -5 && x <= 0, x < -5 || x > 0` C++

  `x > -5 and x <= 0, x < -5 or x > 0` Python

Jako *lista_instrukcji1* i *lista_instrukcji2* mogą wystąpić również kolejne instrukcje warunkowe lub **blok instrukcji** (**blok kodu**).

C++ W języku C++ blok instrukcji wyznaczają nawiasy klamrowe {}.

Python W języku Python ważne są wcięcia w programie – *listę_instrukcji1* i *listę_instrukcji2* należy przesunąć w prawo przynajmniej o jedną spację (przyjęte jest wcięcie składające się z czterech spacji). Jeśli *lista_instrukcji1* i *lista_instrukcji2* obejmują więcej instrukcji, wszystkie należy tak samo przesunąć. Jest to sposób wyróżnienia **bloku kodu**. Jeśli nie zastosujemy wcięcia, komputer potraktuje *listę_instrukcji1* i *listę_instrukcji2* jako *kolejną_instrukcję* i wykona ją niezależnie od spełnienia warunku.

Język	Operator	Określenie	Przykład warunku	Interpretacja wyrażenia
C++ i Python	==	równy	`a == b`	*a* równe *b*
	!=	różny	`a != b`	*a* różne od *b*
	<	mniejszy	`a < b`	*a* mniejsze od *b*
	>	większy	`a > b`	*a* większe od *b*
	>=	większy lub równy	`a >= b`	*a* większe lub równe *b*
	<=	mniejszy lub równy	`a <= b`	*a* mniejsze lub równe *b*

Tabela 1. *Podstawowe operatory porównania w językach C++ i Python*

Uwaga: Do porównywania wartości używamy operatora porównania ==. Nie stosujemy w tym celu operatora przypisania =. W języku C++ zapisanie warunku za pomocą operatora przypisania, np. `if(x = 0)` nie spowoduje błędu kompilacji, ale program będzie działał niepoprawnie.

Przykład 1. Zapisywanie algorytmu z warunkiem logicznym prostym w językach C++ i Python

Chcemy zrealizować w językach C++ i Python algorytm obliczania wartości bezwzględnej liczby całkowitej na podstawie specyfikacji zadania 1. podanej w przykładzie 3. z tematu C1.

C++

[*] Wartosc_bezw.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int x, w;
    cout << "Podaj liczbe x: ";
    cin >> x;
    if(x >=0)
        w = x;
    else
        w = -x;
    cout << "Wartosc bezwzgledna liczby x = " << w;

    return 0;
}
```

C++ Uwagi:

- Programiści używający języka C++ lubią korzystać z konstrukcji języka skracających zapis. Instrukcję: `if (x >= 0) w = x; else w = -x;` można zastąpić instrukcją: `w = x >= 0 ? x : -x;`
- Wartość równą 0 uważa się za logiczny fałsz, a wartość różną od zera – za logiczną prawdę.

Python

Wartosc_bezwgl.py - C:\Python\Wartosc_bezwgl.py (3.7.2)

File Edit Format Run Options Window Help

```
x = int(input( "Podaj liczbe x: "))

if x >= 0:
    w = x
else:
    w = -x
print("Wartość bezwzględna liczby x =", w)


input("\n\nAby zakończyć, naciśnij Enter")
```

Python Uwagi:

- Programiści używający języka Python lubią korzystać z konstrukcji języka skracających zapis. Instrukcję warunkową z powyższego programu można zastąpić instrukcją: `w = x if x >= 0 else -x`
- Wartość równą 0 uważa się za logiczny fałsz, a wartość różną od zera – za logiczną prawdę (tak jak w języku C++).

Ćwiczenie 3. Realizujemy algorytm obliczania wartości bezwzględnej liczby w wybranym języku programowania

1. Przepisz wybrany program podany w przykładzie 1. (zależnie od używanego języka programowania). Sprawdź, czy program jest zgodny ze specyfikacją zadania 1. podaną w przykładzie 3. z tematu C1.
2. Zapisz program w pliku pod nazwą *Wartosc_bezwzgl* i wykonaj program dla kilku różnych wartości *x*.

Uwaga

W języku C++ musimy skompilować program przed uruchomieniem. Nie będziemy o tym przypominać w każdym ćwiczeniu i zadaniu.

Ćwiczenie 4. Realizujemy algorytm z warunkiem logicznym prostym w wybranym języku programowania

1. Napisz specyfikację zadania i program w wybranym języku programowania realizujący algorytm sprawdzania, czy wprowadzona liczba jest dodatnia czy niedodatnia. Wykorzystaj zapis instrukcji warunkowej z rysunku 1.
2. Zapisz program w pliku pod nazwą *Liczby* i wykonaj program dla kilku różnych wartości zmiennych.

2.2. Złożone warunki logiczne

Wszystkie klasy twojej szkoły rywalizują w tzw. lidze klas. Zdobycie największej liczby punktów daje wygraną. Liczba punktów zależy m.in. od frekwencji (np. większa od 94%) i średniej ocen (np. większa od 4,0 lub równa 4,0) w ostatnim półroczu. W jaki sposób napisać program, w którym będziemy mogli obliczać punkty zależnie od spełnienia obydwu warunków?

Jest to przykład sytuacji warunkowej z warunkiem logicznym złożonym. Jeśli frekwencja i średnia ocen spełniają podane warunki, to klasa uzyskuje punkty. Mamy tu do czynienia z logiczną koniunkcją. W wybranym języku programowania zastosujemy zatem instrukcję warunkową z warunkiem złożonym. W instrukcjach warunkowych warunki złożone zapisujemy, stosując odpowiednie operatory logiczne (tabela 2.).

Język	Operator	Określenie	Przykład warunku logicznego	Interpretacja wyrażenia
C++	`||`	alternatywa logiczna (lub)	`a < -5 || a > 5`	*a* mniejsze od –5 lub *a* większe od 5
Python	**`or`**		`a < -5 or a > 5`	
C++	`&&`	koniunkcja logiczna (i)	`a > 0 && a < 10`	*a* większe od 0 i *a* mniejsze od 10
Python	**`and`**		`a > 0 and a < 10`	

Tabela 2. *Operatory logiczne w językach C++ i Python*

Przykład 2. Zapisywanie algorytmu z warunkiem logicznym złożonym w językach C++ i Python

Zadanie: Pewna klasa uczestniczy w lidze klas. Napisz program obliczający, ile wynosi liczba punktów (*pkt*) danej klasy zależnie od uzyskanej frekwencji (*f*) i średniej ocen (*so*) w ostatnim półroczu. Klasa otrzymuje dodatkowe 20 punktów, jeśli frekwencja jest powyżej 94% ($f > 94\%$) i średnia ocen nie jest mniejsza niż 4,0 ($so \geq 4{,}0$). Zdobytą wcześniej liczbę punktów i frekwencję wprowadzaj z klawiatury, a aktualną liczbę punktów wyświetlaj na ekranie.

Dane: liczba całkowita $pkt > 0$ oznaczająca punkty wcześniej zdobyte przez klasę, liczba rzeczywista $f > 0$ oznaczająca procent frekwencji, liczba rzeczywista `so` oznaczająca średnią ocen.

Wyniki: liczba całkowita *pkt* oznaczająca aktualną liczbę punktów danej klasy.

Opis rozwiązania: W rozwiązaniu zastosujemy instrukcję warunkową (w wersji uproszczonej) z warunkiem złożonym zawierającym operator koniunkcji (`&&` lub **`and`**). Zastosujemy wersję uproszczoną, ponieważ niezależnie od spełnienia czy niespełnienia warunku, na ekranie ma być wyświetlana aktualna liczba punktów.

Uwaga: Jeśli podczas pisania programu zwrócimy uwagę na sposób formatowania poszczególnych elementów kodu programu, możemy uniknąć niepotrzebnych błędów. Warto zauważyć, że tekst kodu programu ma różne kolory w zależności od pełnionej roli.

C++ Na przykład: słowa kluczowe i nazwy zmiennych są czarne, operatory, nawiasy – czerwone, napisy – niebieskie.

Python Na przykład: nazwy funkcji `print()`, `input()` są fioletowe, napisy – zielone, słowa kluczowe – pomarańczowe, a nazwy zmiennych – czarne.
Dzięki temu już na etapie pisania programu możemy zauważać błędy, ponieważ nazwa danej funkcji zostanie zaznaczona kolorem fioletowym dopiero wtedy, gdy napiszemy ją w całości poprawnie. Gdy zrobimy błąd, nazwa pozostanie czarna.

C++

[*] Liga.cpp

```cpp
#include <iostream>
using namespace std;
int main()
{
    int pkt;
    float f, so;

    cout << "Podaj liczbe punktow zdobytych przez klase: ";
    cin >> pkt;
    cout << "Podaj frekwencje klasy: ";
    cin >> f;
    cout << "Podaj srednia ocen klasy: ";
    cin >> so;
    if(f > 94 && so >= 4.0)
        pkt = pkt + 20;
    cout << "Aktualna liczba punktow wynosi: " << pkt;

    return 0;
}
```

Python

Liga.py - C:\Python\Liga.py (3.7.2)

File Edit Format Run Options Window Help

```python
pkt = int(input("Podaj liczbę punktów zdobytych przez klasę: "))
f = float(input("Podaj frekwencję klasy: "))
so = float(input("Podaj średnią ocen klasy: "))

if f > 94 and so >= 4.0:
    pkt = pkt + 20
print("Aktualna liczba punktów wynosi: " , pkt)


input("\n\nAby zakończyć, naciśnij Enter")
```

Ćwiczenie 5. Realizujemy algorytm z warunkiem logicznym złożonym w wybranym języku programowania

1. Utwórz nowy plik źródłowy wybranego języka programowania. Przepisz kod programu podany w przykładzie 2.
2. Zapisz program w pliku pod nazwą *Liga* i wykonaj program dla kilku różnych wartości zmiennych. Wyjaśnij, co się dzieje w poszczególnych wierszach programu. Dlaczego zastosowano instrukcję warunkową w wersji uproszczonej (bez części z **`else`**)?

Ćwiczenie 6. Realizujemy algorytm z warunkiem logicznym złożonym w wybranym języku programowania

1. Klasa uczestniczy w lidze klas. Jeśli klasa w ostatnim semestrze uzyskała frekwencję powyżej 93% lub średnią ocen powyżej 4,2, to w nagrodę pojedzie na wycieczkę. Napisz program sprawdzający, czy klasa zakwalifikuje się na wycieczkę. Frekwencję i średnią ocen wprowadzaj z klawiatury, a komunikat „nagroda” lub „brak nagrody” wyświetlaj na ekranie.
2. Zapisz program w pliku pod nazwą *Wycieczka*.

W językach C++ i Python stosuje się **operatory przypisania**, np. instrukcję przypisania: `pkt = pkt + 20` można zapisać krócej: `pkt += 20` (tabela 3.).

Język	Przykład zastosowania	Odpowiednik	Znaczenie
Python, C++ i —	`s += a`	`s = s + a`	przypisanie wartości zmiennej `s` poprzedniej wartości tej zmiennej, powiększonej o `a`
	`s -= a`	`s = s - a`	przypisanie wartości zmiennej `s` poprzedniej wartości tej zmiennej, pomniejszonej o `a`
	`s *= a`	`s = s * a`	przypisanie wartości zmiennej `s` poprzedniej wartości tej zmiennej, powiększonej `a` razy
	`s /= a`	`s = s / a`	przypisanie wartości zmiennej `s` poprzedniej wartości tej zmiennej, pomniejszonej `a` razy

Tabela 3. *Operatory przypisania w językach C++ i Python*

3. Sprawdzanie poprawności danych

Chcemy napisać program obliczający pierwiastek kwadratowy danej liczby. Pierwiastki kwadratowe w zbiorze liczb rzeczywistych można jednak obliczyć tylko dla liczb nieujemnych. Jak sprawdzić w programie, czy wprowadzona przez użytkownika liczba jest nieujemna?

Podczas pisania programów ważne jest sprawdzanie poprawności wprowadzanych danych. Dzięki temu jesteśmy zabezpieczeni przed niespodziewanym przerwaniem pracy programu oraz zapewniamy zgodność danych wejściowych ze specyfikacją zadania.



Poprawny program powinien spełniać kilka kryteriów:

- rozwiązywać problem, dla którego został utworzony;
- rozwiązywać problem dla wszystkich danych określonych w specyfikacji i odpowiednio reagować na wprowadzanie niepoprawnych danych;
- rozwiązywać problem w sposób jak najbardziej efektywny, optymalnie wykorzystując zasoby komputera.

Z punktu widzenia wygody obsługi programu wskazane byłoby, aby użytkownik był natychmiast powiadamiany o wprowadzeniu niepoprawnych danych i mógł je od razu poprawić.

Przykład 3. Sprawdzanie poprawności danych

Zadanie: Napisz program obliczający pierwiastek kwadratowy *p* z liczby rzeczywistej dodatniej *x*, wprowadzanej z klawiatury. Program ma sprawdzać, czy wprowadzona przez użytkownika liczba jest poprawna (nieujemna) i wyprowadzać na ekran wynik obliczenia lub komunikat o błędnych danych.

Dane: liczba rzeczywista *x*.

Wyniki: pierwiastek kwadratowy *p* z liczby *x* lub komunikat „*x* mniejsze od zera".

Opis rozwiązania: Aby sprawdzić, czy wprowadzona przez użytkownika liczba spełnia określone w zadaniu warunki, zastosujemy instrukcję warunkową. Pierwiastki kwadratowe w zbiorze liczb rzeczywistych można obliczyć tylko dla liczb nieujemnych. W programie sprawdzamy, czy wprowadzona liczba *x* jest ujemna. Jeśli tak, wyprowadzamy komunikat o błędnych danych, w przeciwnym wypadku wykonujemy obliczenia i wyprowadzamy wynik na ekran.
Wykorzystamy standardową funkcję matematyczną `sqrt()`, służącą do obliczania pierwiastka kwadratowego.

M Moduł

Plik zawierający kod przeznaczony do wykorzystania w innych programach.

C++

[*] Pierwistek.cpp

```
1  #include <iostream>
2  #include <cmath>          /* sqrt */
3  using namespace std;
4  int main()
5  {
6      float x, p;
7
8      cout << "Podaj x: ";
9      cin >> x;
10     if (x < 0)
11         cout << "x mniejsze od zera";
12     else
13     {
14         p = sqrt(x);
15         cout << "Pierwiastek z " << x << " = " << p;
16     }
17     return 0;
18 }
```

dołączenie biblioteki standardowej `cmath` zawierającej funkcje matematyczne, m.in. funkcję `sqrt()` zwracającą pierwiastek kwadratowy z nieujemnej liczby rzeczywistej

wywołanie funkcji `sqrt()`

Python

Pierwiastek.py - C:\Python\Pierwiastek.py (3.7.2)

File Edit Format Run Options Window Help

```
from math import sqrt                    #sqrt

x = float(input("Podaj x: "))

if x < 0:
    print("x mniejsze od zera")
else:
    p = sqrt(x)
    print( "Pierwiastek z ", x,  " = ", p)

input("\n\nAby zakończyć, naciśnij Enter")
```

importowanie z **modułu** `math` (zawierajacego funkcje matematyczne) funkcji `sqrt()` zwracającej pierwiastek kwadratowy z nieujemnej liczby rzeczywistej

wywołanie funkcji `sqrt()`

Uwaga

W przykładach i ćwiczeniach dotyczących obliczeń nie uwzględniamy przypadków wprowadzenia danych innych niż liczba. Takie przypadki omówimy na informatyce realizowanej w zakresie rozszerzonym.

Ćwiczenie 7. Sprawdzamy poprawność danych liczbowych

1. Utwórz nowy plik źródłowy wybranego języka programowania. Przepisz odpowiedni kod programu podany w przykładzie 3.
2. Zapisz program w pliku pod nazwą *Pierwiastek* i wykonaj dla kilku różnych wartości zmiennych. Wyjaśnij, co się dzieje w poszczególnych wierszach programu. Dlaczego zastosowano instrukcję warunkową w pełnej wersji (z `else`)?

Ćwiczenie 8. Stosujemy instrukcję warunkową do sprawdzania poprawności wprowadzanych danych liczbowych

1. Napisz program obliczający pole trójkąta. W programie uwzględnij sprawdzanie poprawności wprowadzanych danych (podstawy i wysokości trójkąta).
2. Zapisz program w pliku pod nazwą *Trojkat*.

4. Algorytmy z warunkami zagnieżdżonymi

Chcemy napisać program realizujący algorytm wyboru najmniejszej z trzech różnych liczb całkowitych wprowadzanych z klawiatury. W jaki sposób po sprawdzeniu, czy pierwsza liczba jest mniejsza od drugiej, sprawdzać, czy pierwsza jest też mniejsza od trzeciej oraz druga od trzeciej?

Instrukcje warunkowe, podobnie jak funkcja JEŻELI w arkuszu kalkulacyjnym, mogą się **zagnieżdżać**, czyli po warunku oraz po słowie `else` może wystąpić kolejna instrukcja warunkowa.

Ćwiczenie 9. Piszemy specyfikację problemu

Sformułuj zadanie i napisz specyfikację zadania do sytuacji problemowej podanej w nawiasach klamrowych w punkcie 4. tego tematu. Przyjmij następujące nazwy zmiennych *a*, *b*, *c* (dla sprawdzanych liczb), *najmniejsza* (dla zapamiętania najmniejszej liczby).

Przykład 4. Zapisywanie algorytmu z warunkami zagnieżdżonymi w językach C++ i Python

Napiszemy program realizujący algorytm wyboru najmniejszej z trzech różnych liczb całkowitych. W rozwiązaniu zastosujemy zagnieżdżone instrukcje warunkowe.

C++

[*] Najmniejsza_z_trzech.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    int a, b, c, najmniejsza;

    cout << "Podaj trzy liczby: " << endl;
    cin >> a >> b >> c;
    if(a < b)
    {
        if(a < c)
            najmniejsza = a;
        else
            najmniejsza = c;
    }
    else
    {
        if(b < c)
            najmniejsza = b;
        else
            najmniejsza = c;
    }
```

C++ Uwagi:

- W przypadku instrukcji zagnieżdżonych zaleca się pisać słowo kluczowe **`else`** pod odpowiadającym mu słowem **`if`** (dla zachowania czytelności).
- Wszystkie instrukcje **`if`** można zastąpić zapisem:
  `najmniejsza = a < b ? (a < c ? a : c) : (b < c ? b : c);`

Python

```python
a = int(input("Podaj pierwszą liczbę: "))
b = int(input("Podaj drugą liczbę: "))
c = int(input("Podaj trzecią liczbę: "))

if a < b:
    if a < c:
        najmniesza = a
    else:
        najmniesza = c
else:
    if b < c:
        najmniesza = b
    else:
        najmniesza = c

print("Najmniesza liczba wynosi:", najmniesza)

input("\n\nAby zakończyć, naciśnij Enter")
```

Ćwiczenie 10. Stosujemy zagnieżdżone instrukcje warunkowe

1. Przepisz program podany w przykładzie 4. (zależnie od używanego języka programowania).
2. Zapisz program w pliku pod nazwą *Najmniejsza_z_trzech*. Zwróć uwagę na zapis zagnieżdżonych instrukcji warunkowych.
3. Uruchom program i przetestuj dla następujących trójek liczb (wartości zmiennych *a*, *b*, *c*): (3; -56; 0), (76; 123; -2), (44; 44; 44), (-70; -40; -40), (56; 68; 88), (100; 0; 2).
4. Wyjaśnij, co dzieje się w poszczególnych wierszach programu, zwracając uwagę na działanie instrukcji warunkowej. W zeszycie przedmiotowym dokończ opis: *Sprawdzamy, czy pierwsza liczba jest mniejsza od drugiej. Jeśli tak, to…*
5. Odpowiedz na pytanie: *Kiedy będzie wykonane przypisanie* `najmniejsza = b`*?*

Ćwiczenie 11. Zapisujemy problem sprzedawcy komputera (z tematu C1) w języku programowania

1. Przypomnij sobie problem sprzedawcy komputerów (przykład 1. z tematu C1).
2. Korzystając ze specyfikacji zadania (przykład 4. z tematu C1) i zaprojektowanego rozwiązania (przykład 5. z tematu C1), napisz program realizujący przedstawione rozwiązanie. Zadbaj o sprawdzanie poprawności wprowadzanych danych (zgodnie ze specyfikacją).
3. Zapisz program w pliku pod nazwą *Cena_komputera*, uruchom i przetestuj dla różnych wartości zmiennych (*w*, *cenak*, *cenap*).

Warto zapamiętać

- W algorytmie z warunkami występują sytuacje warunkowe – wynik lub dalsze działanie algorytmu zależą od spełnienia lub niespełnienia warunku.
- W językach programowania do zapisania algorytmów z warunkami stosujemy instrukcje warunkowe. Działanie tych instrukcji jest podobne w większości języków programowania.
- Warunek w instrukcji warunkowej może być prosty lub złożony. W warunkach złożonych stosujemy operatory alternatywy i koniunkcji logicznej: `||`, `&&` (w języku C++) i **`or`**, **`and`** (w języku Python).
- Program powinien efektywnie rozwiązywać problem dla wszystkich danych określonych w specyfikacji.
- Należy dbać o zgodność danych wejściowych ze specyfikacją i sprawdzać w programie, czy użytkownik wprowadził właściwe dane.
- W programach komputerowych można stosować zagnieżdżone instrukcje warunkowe.

Pytania i polecenia

1. Wyjaśnij, na czym polega sytuacja warunkowa. Podaj przykład sytuacji warunkowej.
2. W jaki sposób zapisuje się w językach programowania algorytmy zawierające sytuacje warunkowe?
3. Omów działanie instrukcji warunkowej w wybranym języku programowania. Porównaj działanie tej instrukcji do działania funkcji JEŻELI w arkuszu kalkulacyjnym.
4. Podaj dwa przykłady zapisania złożonych warunków logicznych.
5. Jakie kryteria powinien spełniać poprawny program?
6. Podaj przykład sytuacji, w której należy sprawdzić poprawność danych.
7. Omów sposób zapisywania zagnieżdżonej instrukcji warunkowej.

Zadania

Uwagi:

- Staraj się dodawać do programów odpowiednie komunikaty dla użytkownika.
- Pisz przejrzyste programy oraz dodawaj w odpowiednich miejscach komentarze.
- Każdy program uruchom i przetestuj dla kilku różnych danych, nawet jeśli w zadaniu nie ma takiego polecenia (w przypadku języka C++ najpierw skompiluj program).

1. Kiedy zostanie wykonana instrukcja `print(x - y)` (zapisana w języku Python), a kiedy instrukcja `cout << x - y` (w języku C++) w programach z podpunktów a i b? Rozważ przypadki: `x > y`, `x = y`, `x < y`. Czym różni się efekt działania instrukcji **`if`** w punktach a i b?

a.

C++

```
cin >> x >> y;
if(x > y)
    x = x - y;
cout << x - y;
```

Python

```
x = int(input("Podaj pierwszą liczbę: "))
y = int(input("Podaj drugą liczbę: "))
if x > y:
    x = x - y
print(x - y)
```

b.

C++

```
cin >> x >> y;
if(x > y)
    x = x - y;
else
    cout << x - y;
```

Python

```
x = int(input("Podaj pierwszą liczbę: "))
y = int(input("Podaj drugą liczbę: "))
if x > y:
    x = x - y
else:
    print(x - y)
```

2. Napisz specyfikację zadania i program realizujący algorytm sprawdzania, czy wprowadzona z klawiatury liczba jest parzysta czy nieparzysta. Zapisz plik pod nazwą *Parzyste_nieparzyste*.
   Wskazówka: Oblicz resztę z dzielenia wprowadzonej liczby przez 2.
3. Napisz specyfikację zadania i program realizujący algorytm sprawdzania, czy liczba jest podzielna przez 3. Zapisz plik pod nazwą *Podzielna_przez_3*.
   Wskazówka: Oblicz resztę z dzielenia wprowadzonej liczby przez 3.
4. Napisz specyfikację zadania i program realizujący algorytm sprawdzania, która z dwóch liczb rzeczywistych jest większa. Uwzględniaj przypadek, gdy liczby są równe. Zapisz plik pod nazwą *Wieksza_z_dwoch*.
5. Napisz specyfikację zadania: *Wprowadź dowolną liczbę rzeczywistą. Gdy liczba jest dodatnia, obliczaj pole koła o promieniu równym tej liczbie. Dla liczby ujemnej podawaj jej wartość bezwzględną, a dla równej zero – wyprowadź napis „równa zero”*. Napisz program na podstawie zapisanej specyfikacji. Zapisz plik pod nazwą *Zadanie5*.
6. Napisz specyfikację zadania i program obliczający prędkość samochodu *v*, gdy dane są droga *s* i czas *t*. Jeśli prędkość samochodu przekracza 90 km/h, wyświetlaj komunikat „za szybko”. Jeśli prędkość będzie równa 90/km, wyświetlaj komunikat „w sam raz”, a jeśli mniejsza, wyświetlaj komunikat „za wolno”. Program napisz na podstawie przygotowanej wcześniej specyfikacji. Zapisz plik pod nazwą *Predkosc*.
7. Klasa ma być podzielona na dwie grupy, w zależności od wyniku testu kompetencji językowych z wybranego języka obcego lub oceny na świadectwie ukończenia szkoły podstawowej. Jeśli dany uczeń uzyskał z testu wynik powyżej 90% punktów lub jeśli na świadectwie ukończenia szkoły podstawowej miał z danego języka ocenę nie niższą niż 5, to kwalifikuje się do grupy zaawansowanej. W przeciwnym wypadku kwalifikuje się do grupy podstawowej. Napisz program sprawdzający, do jakiej grupy zakwalifikuje się dany uczeń. Liczbę punktów uzyskanych z testu i ocenę ze świadectwa wprowadzaj z klawiatury, a komunikat: „grupa zaawansowana” lub „grupa podstawowa” wyświetlaj na ekranie. Zapisz plik pod nazwą *Grupy*.

8. Napisz specyfikację zadania i program realizujący algorytm obliczania objętości walca o wysokości *h* i promieniu podstawy *r*, gdzie *h* i *r* są dodatnimi liczbami rzeczywistymi. Zadbaj o poprawność wprowadzanych danych, zgodnie z zapisaną wcześniej specyfikacją. Dla liczb niedodatnich powinien być wyprowadzany napis błędne dane. Zapisz plik pod nazwą *Objetosc_walca*.
9. Otwórz plik *Przepis* zapisany w ćwiczeniu 9. z tematu C2 lub 10. z tematu C3. Dodaj do programu sprawdzanie poprawności wprowadzanych danych, zgodnie ze specyfikacją. Zapisz plik pod tą samą nazwą.

Dla zainteresowanych

10. Napisz specyfikację zadania i program sprawdzający, czy z odcinków o długościach wprowadzonych przez użytkownika z klawiatury można zbudować trójkąt (warunek istnienia trójkąta przypomnij sobie z lekcji matematyki). Program ma sprawdzać dziesięć trójek liczb i zależnie od rezultatu wyprowadzać komunikaty: „można zbudować trójkąt" lub „nie można zbudować trójkąta". Dodaj do programu sprawdzanie poprawności wprowadzanych danych. Zapisz plik pod nazwą *Trojkat*.
11. Napisz specyfikację zadania i program sprawdzający, czy wprowadzone trzy liczby naturalne są bokami trójkąta prostokątnego. Zależnie od rezultatu program ma wyprowadzać komunikaty „tak" lub „nie". Zapisz plik pod nazwą *Trojkat_prostokat*.
12. Otwórz plik *Najmniejsza_z_trzech* zapisany w ćwiczeniu 10. Zmodyfikuj program tak, aby był uwzględniony przypadek, gdy liczby są równe. Zapisz plik pod tą samą nazwą.
13. Zadanie dla dwóch osób: Każda z osób projektuje zadanie według własnego pomysłu (można wzorować się na zadaniach z tego tematu) – formułuje treść zadania, pisze specyfikację i przekazuje zadanie drugiej osobie, aby napisała program. Potem program wraca do autora zadania, który testuje go dla różnych danych.

Temat C5. Stosowanie instrukcji iteracyjnych w językach C++ i Python

1. Iteracja – podstawowa technika algorytmiczna
2. Stosowanie iteracji do sumowania liczb
3. Algorytmy iteracyjne w językach C++ i Python
   - 3.1. Instrukcja iteracyjna `for` w języku C++
   - 3.2. Instrukcja iteracyjna `for` w języku Python
   - 3.3. Zapisywanie rozwiązania problemu iteracyjnego w językach C++ i Python
4. Instrukcje iteracyjne zagnieżdżone



Warto powtórzyć

1. Czym charakteryzuje się algorytm z warunkami?
2. Jak działa instrukcja warunkowa w wybranym języku programowania?
3. Kiedy program jest poprawny?
4. Czym jest blok instrukcji (blok kodu) i jak go wyróżniamy?

1. Iteracja – podstawowa technika algorytmiczna

Iteracja, czyli technika powtarzania tych samych operacji, pojawia się w wielu zadaniach. Często trzeba wykonać takie same działania (ciągi operacji) na wielu różnych danych, co jest pracochłonne i wymaga sporo czasu.

Jednym z pierwszych zadań komputera było zastępowanie człowieka w wykonywaniu powtarzających się obliczeń na wielu liczbach. W tym celu powstały też pierwsze maszyny liczące. Rozwiązywanie problemów iteracyjnych stało się ważnym zastosowaniem komputera, zwłaszcza że tworzenie programów komputerowych ma sens przede wszystkim w przypadku czynności powtarzalnych.

Prawie wszystkie wcześniej omawiane problemy można rozwiązać, korzystając z takich narzędzi, jak arkusz kalkulacyjny czy kalkulator. Jednak, w celu wielokrotnego wykonania (tysiące, a nawet miliony razy) tych samych operacji, o wiele szybciej i efektywniej jest skorzystać z programu komputerowego utworzonego specjalnie w tym celu.



Iteracja polega na powtarzaniu tej samej operacji (ciągu operacji).

Iterację **implementujemy**, stosując tzw. **pętlę**. Z pętlą mamy do czynienia, gdy w pewnym kroku algorytmu wracamy do jednego z wcześniejszych kroków, co powoduje, że kroki te zostaną wykonane wiele razy.

2. Stosowanie iteracji do sumowania liczb

Zastanówmy się, jak dodajemy w pamięci lub na kalkulatorze (np. siedem liczb). Zadanie wydaje się banalne i zazwyczaj odpowiadamy: „po prostu dodajemy do siebie wszystkie liczby i podajemy wynik". Ale co to znaczy „dodać do siebie wszystkie liczby"?

Wykonując to zadanie w pamięci lub na kalkulatorze, realizujemy pewien algorytm: dodajemy dwie pierwsze liczby do siebie, zapamiętujemy ich sumę, następnie do tej sumy dodajemy trzecią liczbę, znów zapamiętujemy wynik itd. Opis tego algorytmu może wyglądać jak w przykładzie 1.

Przykład 1. Obliczenie sumy siedmiu liczb

1. Zacznij algorytm.
2. Suma jest równa zero.
3. Wprowadź pierwszą liczbę.
4. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
5. Wprowadź drugą liczbę.
6. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
7. Wprowadź trzecią liczbę.
8. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
9. Wprowadź czwartą liczbę.
10. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
11. Wprowadź piątą liczbę.
12. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
13. Wprowadź szóstą liczbę.
14. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
15. Wprowadź siódmą liczbę.
16. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
17. Wyprowadź wynik sumowania.
18. Zakończ algorytm.

Zauważmy, że operacje zapisane w krokach 3. i 4. z przykładu 1. powtarzają się w kolejnych parach kroków (5-6, 7-8, 9-10, 11-12, 13-14 i 15-16). Gdybyśmy mieli zsumować np. kilka tysięcy liczb, opis algorytmu byłby bardzo długi. W przedstawionej sytuacji możemy powtarzające się ciągi operacji (tu kroki 3. i 4.) zapisać tylko raz i zastosować pętlę, czyli po wykonaniu czwartego kroku wrócić do trzeciego, to znaczy wykonać kroki 3. i 4. siedem razy.

W przykładzie 1. dodaliśmy siedem liczb. Zazwyczaj jednak zapisujemy algorytm w uogólnionej postaci – dla dowolnej liczby danych. Taki algorytm jest uniwersalny.

Przykład 2. Zapisujemy algorytm sumowania *n* liczb w postaci listy kroków

Zadanie: Oblicz sumę *n* liczb całkowitych.

Dane: liczba naturalna *n* większa od zera, oznaczająca, ile liczb ma być sumowanych, *n* dowolnych liczb całkowitych.

Wynik: wartość sumy: *suma*.

Lista kroków:

1. Zacznij algorytm.
2. Zmiennej `suma` przypisz wartość 0.
3. Zmiennej `i` przypisz wartość 0.
4. Podaj, ile liczb ma być zsumowanych. Zapamiętaj wprowadzoną liczbę w zmiennej `n`.
5. Wprowadź liczbę do sumowania – zapamiętaj ją w zmiennej `a`.
6. Zmiennej `suma` przypisz wartość zmiennej `suma` powiększoną o wartość zmiennej `a`: `suma = suma + a`.
7. Zmiennej `i` przypisz wartość zmiennej `i` powiększoną o `1`: `i = i + 1`.
8. Jeśli `i < n`, wróć do kroku 5.
9. Wyprowadź wynik: `suma`.
10. Zakończ algorytm.

W przykładzie 2. wartość sumy liczb dla kolejno wprowadzanych danych jest zapamiętywana w zmiennej `suma`, której na początku algorytmu przypisujemy wartość zero.

Każdą z kolejno wprowadzanych liczb zapamiętujemy w zmiennej `a` (w przypadku algorytmu iteracyjnego zwykle nie stosujemy różnych nazw zmiennych dla kolejnych liczb).

Przypisanie `suma = suma + a` oznacza „pod zmienną `suma` podstaw poprzednią wartość `suma` zwiększoną o wartość kolejnej liczby `a`". Jeśli w tym samym programie zapiszemy w tej samej zmiennej nową wartość (wykonamy instrukcję przypisania), to poprzednia wartość zostanie usunięta. Na przykład po wykonaniu kolejno następujących instrukcji podstawienia:

C++

```
suma = 7;
suma = suma + 20;
```

Python

```
suma = 7
suma = suma + 20
```

w zmiennej `suma` będzie pamiętana wartość 27.



Przykład 3. Analiza działania algorytmu dodawania *n* liczb

Prześledzimy działanie algorytmu dla `n` = 7 i dla kolejnych wartości zmiennej `a`: 7, 20, 13, 15, 8, 11, 33.
Zmienna `i` oznacza w tym przykładzie numer wprowadzonej liczby. W analizie przedstawiamy wartości zmiennych po wykonaniu każdego cyklu iteracji.
Na początku przyjmujemy: `suma` = 0, `i` = 0, `n` = 7.

cykl 1.	`a` = 7	`suma` = 0 + 7 = 7	`i` = 0 + 1 = 1	czy `i < n`,	`i` < 7? TAK
cykl 2.	`a` = 20	`suma` = 7 + 20 =27	`i` = 1 + 1 = 2	czy `i < n`,	`i` < 7? TAK
cykl 3.	`a` = 13	`suma` = 27 + 13 = 40	`i` = 2 + 1 = 3	czy `i < n`,	`i` < 7? TAK
cykl 4.	`a` = 15	`suma` = 40 + 15 = 55	`i` = 3 + 1 = 4	czy `i < n`,	`i` < 7? TAK
cykl 5.	`a` = 8	`suma` = 55 + 8 = 63	`i` = 4 + 1 = 5	czy `i < n`,	`i` < 7? TAK
cykl 6.	`a` = 11	`suma` = 63 + 11 = 74	`i` = 5 + 1 = 6	czy `i < n`,	`i` < 7? TAK
cykl 7.	`a` = 33	`suma` = 74 + 33 =107	`i` = 6 + 1 = 7	czy `i < n`,	`i` < 7? NIE

Wyprowadzenie wartości zmiennej `suma`: 107.

Ćwiczenie 1. Analizujemy działanie algorytmu dodawania liczb

1. Korzystając z przykładów 2. i 3., prześledź działanie algorytmu dodawania n liczb dla $n = 5$ i dla kolejnych wartości zmiennej a: 23, -35, 40, -7, 25.
2. Wykonaj to zadanie, korzystając z programu Kalkulator. Zwróć uwagę, że zadanie zawsze wykonywane jest według tego samego algorytmu.

3. Algorytmy iteracyjne w językach C++ i Python

Chcemy napisać program obliczający sumę n liczb. W algorytmie mamy do czynienia z powtarzaniem poleceń wprowadzania liczby i dodawania. W jaki sposób napisać program realizujący iterację w wybranym języku programowania?

Aby w języku programowania napisać program realizujący algorytm iteracyjny, stosujemy instrukcje iteracyjne (zwane też instrukcjami pętli).

3.1. Instrukcja iteracyjna `for` w języku C++

W języku C++ są dostępne trzy instrukcje iteracyjne. Pokażemy zastosowanie jednej z nich – instrukcji **`for`**.

Instrukcja iteracyjna **`for`**	**`for`** *(wyrażenie_początkowe; warunek; wyrażenie_pętli)* *lista_instrukcji;*	C++

Najpierw wykonywane jest *wyrażenie_początkowe* (np. $i = 0$), zwykle ustalające wartość początkową zmiennej sterującej. Następnie, dopóki spełniony jest *warunek*, wykonywana jest *lista_instrukcji*, a potem obliczana wartość *wyrażenia_pętli*, służącego zazwyczaj do zwiększenia (lub zmniejszenia) wartości zmiennej sterującej. Jako *lista_instrukcji* może wystąpić pojedyncza instrukcja (w tym kolejna instrukcja pętli) lub więcej instrukcji ujętych w blok `{}`.

W języku C++ do określania kroków iteracji (powtórzeń) w instrukcji **`for`** stosujemy zwykle tzw. **zmienną sterującą**.

Przykład 4. Stosowanie instrukcji for w języku C++

W języku C++ zwyczajowo początkową wartość zmiennej sterującej przyjmuje się jako równą 0, a warunek zapisuje jako np. `i < n` (jeśli chcemy, aby liczba iteracji wynosiła `n`). Wyrażenie `i++` oznacza „zmiennej `i` przypisz wartość `i` zwiększoną o 1”. Jest to skrót instrukcji przypisania: `i = i + 1`.

```
for (i = 0; i < 7; i++)
      cout << "*";
```

Zmienna sterująca `i` na początku ma wartość 0. Instrukcja `cout << "*"` zostanie wykonana siedem razy dla zmiennej `i` mniejszej od 7 (czyli równej kolejno: 0, 1, 2, 3, 4, 5, 6). Na ekranie zostanie wyświetlonych siedem gwiazdek w jednym wierszu.

```
for (i = 0; i <= n; i++)
      cout << "Witaj" << endl;
```

Instrukcja `cout "Witaj" << endl` zostanie wykonana `n + 1` razy dla zmiennej `i` mniejszej od `n` lub równej `n` (czyli równej kolejno: 0, 1, 2, 3,… `n`). Program wypisze `n + 1` razy w kolejnych wierszach napis „Witaj".

```
for (i = 0; i < k; i++)
{
      cout << "Podaj liczbe: ";
      cin >> liczba;
}
```

Instrukcje ujęte w blok `{}` zostaną wykonane `k` razy dla zmiennej `i` równej kolejno: 0, 1, 2, 3, … `k`-1). Program wyświetli napis „Podaj liczbe: ", po czym będzie czekał na wprowadzenie wartości zmiennej `liczba`. Te czynności zostaną powtórzone `k` razy.

```
for (i = 20; i >= 1; i--)
      cout << i << endl;
```

Instrukcja `cout << i << endl` zostanie wykonana 20 razy dla zmiennej `i` równej kolejno: 20, 19, 18, … 1). Program wypisze w kolejnych wierszach wartości zmiennej `i`.



Ćwiczenie 2. Stosujemy instrukcję **`for`** w języku C++

Korzystając z przykładu 4., napisz cztery programy:

a. Program wyświetlający piętnaście razy znak „@" w tym samym wierszu.
b. Program wyświetlający na ekranie w kolejnych wierszach *n* napisów „Lubię informatykę" (wartość zmiennej *n* wprowadzaj z klawiatury).
c. Program wprowadzający *k* liczb rzeczywistych z klawiatury i zapamiętujący je kolejno w zmiennej *liczba*.
d. Program wyświetlający w kolumnie liczby całkowite od 10 do -10.

Zmienną sterującą *i* zadeklaruj jako **`int`**. Zadeklaruj również inne używane zmienne, dobierając odpowiednio ich typ. Każdy program zapisz w pliku, skompiluj i uruchom.

3.2. Instrukcja iteracyjna `for` w języku Python

W języku Python są dostępne dwie instrukcje iteracyjne. Pokażemy zastosowanie jednej z nich – instrukcji **`for`**.

Instrukcja iteracyjna **`for`**	**`for`** *`zmienna`* **`in`** *`lista_wartości:`* *`lista_instrukcji`*	Python

Jako *lista_instrukcji* może wystąpić pojedyncza instrukcja (w tym instrukcja pętli) lub więcej instrukcji (blok instrukcji). *Lista_instrukcji* musi być przesunięta w prawo przynajmniej o jedną spację (przyjęte jest wcięcie składające się z czterech spacji).



W języku Python liczbę iteracji w instrukcji **`for`** określa długość *listy_wartości* po słowie **`in`**. *Lista_instrukcji* zostanie wykonana dla wszystkich wartości z *listy_wartości*. *Listę_wartości* możemy zapisać w różny sposób.

Przykład 5. Stosowanie instrukcji `for` w języku Python

Podajemy przykłady określania *listy_wartości* w instrukcji `for`:

```
for i in [0, 1, 2, 3, 4, 5]:
    print(i)
```

Instrukcja `print(i)` zostanie wykonana sześć razy dla listy wartości [0, 1, 2, 3, 4, 5]; zmienna `i` będzie przyjmować kolejne wartości z tej listy, czyli: 0, 1, 2, 3, 4, 5.

```
for i in [2, 4, 6, 8, 10]:
    print(i)
```

Instrukcja `print(i)` zostanie wykonana pięć razy dla listy [2, 4, 6, 8, 10]; zmienna `i` będzie przyjmować kolejne wartości z tej listy, czyli: 2, 4, 6, 8, 10.

Nie zawsze chcemy wpisywać wszystkie wartości listy, zwłaszcza dla dużej liczby iteracji. Do utworzenia *listy_wartości* można użyć funkcji `range()`, która tworzy sekwencję wartości całkowitych.

Przykład 6. Stosowanie funkcji `range()` do określania *listy_wartości* w instrukcji `for`

Argumentami funkcji `range()` mogą być konkretne wartości lub zmienne, np. `range(10)`, `range(n)`. Wartość zmiennej `n` możemy wprowadzać z klawiatury.

- Dla jednego argumentu: `range(`*koniec*`)`

  ```
  for i in range(7):
      a = int(input("Podaj liczbę: "))
  ```

  Instrukcja `a = int(input("Podaj liczbę: "))` zostanie wykonana siedem razy. Funkcja `range()` wygeneruje kolejne liczby całkowite z przedziału ⟨0, *koniec*), czyli zmienna `i` będzie przyjmować kolejno wartości: 0, 1, 2, 3, 4, 5, 6.

  ```
  for i in range(10):
      print("Witaj")
  ```

  Instrukcja `print("Witaj")` zostanie wykonana dziesięć razy. Funkcja `range()` wygeneruje kolejne liczby całkowite z przedziału ⟨0, *koniec*), czyli zmienna `i` będzie przyjmować kolejno wartości: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

- Dla dwóch argumentów: `range(`*początek*`, `*koniec*`)`

  ```
  for i in range(18, 95):
      print(i)
  ```

  Instrukcja `print(i)` zostanie wykonana siedemdziesiąt siedem (95 – 18 = 77) razy. Funkcja `range()` wygeneruje kolejne liczby całkowite z przedziału ⟨*początek*, *koniec*), czyli zmienna `i` będzie przyjmować kolejno wartości: 18, 19, 20, …, 94.

- Dla trzech argumentów: `range(`*początek*`, `*koniec*`, `*krok*`)`

  ```
  for i in range(2, 102, 2):
      print(i)
  ```

  Instrukcja `print(i)` zostanie wykonana pięćdziesiąt razy. Funkcja `range()` wygeneruje kolejne liczby całkowite z przedziału ⟨*początek*, *koniec*), zmieniające się o *krok*, czyli zmienna `i` będzie przyjmować kolejno wartości: 2, 4, 6, …, 100. Trzeci argument (*krok*) określa tym samym, o jaką wartość zmienia się zmienna `i`.

```
for i in range(10, 0, -1):
    print(i)
```

Instrukcja `print(i)` zostanie wykonana dziesięć razy. *Krok* wynosi -1, czyli zmienna `i` będzie przyjmować kolejno wartości: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.



Ćwiczenie 3. Stosujemy instrukcję `for` w języku Python

Korzystając z przykładów 5. i 6., napisz pięć programów:

a. Program wprowadzający 10 liczb rzeczywistych z klawiatury i zapamiętujący je kolejno w zmiennej *liczba*.
b. Program wyświetlający na ekranie w kolejnych wierszach *n* napisów „Lubię informatykę" (wartość zmiennej *n* wprowadzaj z klawiatury).
c. Program wyświetlający w kolumnie liczby całkowite od 1 do *k* (wartość zmiennej *k* wprowadzaj z klawiatury).
d. Program wyświetlający w kolumnie liczby całkowite nieparzyste od 1 do 33.
e. Program wyświetlający w kolumnie liczby całkowite od 10 do -10.

Każdy program zapisz w pliku i uruchom.

3.3. Zapisywanie rozwiązania problemu iteracyjnego w językach C++ i Python



Przykład 7. Zapisywanie algorytmu iteracyjnego w językach C++ i Python

Algorytm sumowania *n* liczb całkowitych wprowadzanych z klawiatury na podstawie specyfikacji i listy kroków podanej w przykładzie 2.

Uwaga: W językach C++ i Python instrukcję przypisania `suma = suma + a` można zapisać krócej: `suma += a` (tabela 3., temat C4).

C++

[*] Suma_n.cpp

```
#include <iostream>
using namespace std;

int main()
{
    int i, a, suma, n;

    cout << "Ile liczb: ";
    cin >> n;
    suma = 0;
    for(i = 0; i < n; i++)
    {
        cout << "Podaj liczbe: ";
        cin >> a;
        suma = suma + a;
    }
    cout << "Suma wynosi: " << suma;

    return 0;
}
```

Python

Suma_n.py - C:/Python/Suma_n.py (3.7.2)

File Edit Format Run Options Window Help

```
n = int(input("Ile liczb: "))

suma = 0
for i in range(n):
    a = int(input("Wprowadź liczbę: "))
    suma = suma + a
print("Suma wynosi:", suma)


input("\n\nAby zakończyć, naciśnij Enter")
```

Język	Rodzaj operatora	Przykład zastosowania	Odpowiednik	Znaczenie
C++	**operator inkrementacji**	`i++` lub `++i`	`i = i + 1`	przypisanie wartości zmiennej `i` poprzedniej wartości tej zmiennej, powiększonej o 1
	operator dekrementacji	`i--` lub `--i`	`i = i - 1`	przypisanie wartości zmiennej `i` poprzedniej wartości tej zmiennej, pomniejszonej o 1

Tabela 1. *Operatory inkrementacji i dekrementacji w języku C++*

Ćwiczenie 4. Zapisujemy algorytm sumowania *n* liczb w języku programowania.

1. Przepisz wybrany program z przykładu 7. Zapisz program w pliku pod nazwą *Suma_n*.
2. Uruchom i przetestuj program dla kilku różnych wartości zmiennych. Wyjaśnij, co się dzieje w poszczególnych wierszach programu.
3. Dodaj do programu sprawdzanie poprawności wprowadzanej wartości zmiennej *n*, zgodnie ze specyfikacją zadania (podaną w przykładzie 2.). Sumowanie wprowadzanych liczb powinno być wykonywane dla poprawnej wartości *n*. Jeśli użytkownik wprowadzi błędne dane, powinien wyświetlić się komunikat „Niepoprawne dane". Zapisz plik pod tą samą nazwą i uruchom program dla różnych wartości zmiennej *n*.

Ćwiczenie 5. Stosujemy instrukcję iteracyjną `for`

1. Napisz specyfikację zadania i program obliczający sumę *n* kolejnych liczb naturalnych (począwszy od 1). Sprawdzaj poprawność wprowadzania wartości zmiennej *n*.
2. Zapisz program w pliku pod nazwą *Naturalne_suma_n*, uruchom i sprawdź jego działanie dla różnych wartości zmiennej *n*.

4. Instrukcje iteracyjne zagnieżdżone

Instrukcje iteracyjne **for** mogą być zagnieżdżone, czyli instrukcją powtarzaną w pętli może być kolejna instrukcja pętli (przykład 8.). Liczba kroków tej iteracji jest określona przez iloczyn: $n \cdot m$, gdzie n i m to liczby powtórzeń odpowiednio w każdej pętli.

Przykład 8. Pętla w pętli w językach C++ i Python

Zadanie: Zapisz w wybranym języku programowania algorytm, który umożliwi wyświetlenie na ekranie monitora „prostokąta" utworzonego ze znaków „x" o bokach *n*, *m* (*m* – liczba znaków „x" w poziomie, *n* – liczba znaków „x" w pionie). Wnętrze prostokąta ma być również wypełnione znakami „x".

Python Uwaga: Domyślnie po wykonaniu funkcji `print()` kursor przechodzi do następnego wiersza, czyli ostatnim znakiem wypisywanym przez funkcję `print()` jest znak nowego wiersza. Jeśli tego nie chcemy, możemy użyć parametru `end` i określić, co ma być ostatnim znakiem, np. pusty ciąg znaków (`end = ""`).

C++

[*] Prostokat.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    int i, j, m, n;
    cout << "Podaj wymiary prostokata: ";
    cin >> m >> n;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            cout << "x";
        cout << endl;
    }
    return 0;
}
```

Python

Prastokat.py - C:\Python\Prastokat.py (3.7.2)

File Edit Format Run Options Window Help

```python
m = int(input("Podaj pierwszy wymiar prostokata: "))
n = int(input("Podaj pierwszy drugi prostokata: "))

for i in range(n):
    for j in range(m):
        print("x", end = "")
    print()
```

Ćwiczenie 6. Stosujemy zagnieżdżone instrukcje iteracyjne `for`

1. Napisz specyfikację zadania z przykładu 8.
2. Przepisz wybrany program z przykładu 8. Zapisz program w pliku pod nazwą *Prostokat*, uruchom i przetestuj go dla kilku różnych wartości zmiennych *m* i *n*. Objaśnij, co się dzieje w poszczególnych wierszach programu, m.in. wskaż w programie miejsce (instrukcje), gdzie powstaje „wiersz prostokąta”.

W języku C++ należy stosować odrębną nazwę zmiennej sterującej określającą liczbę iteracji dla pętli zewnętrznej i wewnętrznej.

Ćwiczenie 7. Sprawdzamy działanie pętli zagnieżdżonych dla takich samych zmiennych określających liczbę iteracji

1. Otwórz plik *Prostokat* zapisany w ćwiczeniu 6.
2. Sprawdź, co się stanie, jeśli zamiast zmiennej *j* w wewnętrznej pętli użyjesz również zmiennej *i*. Zmodyfikuj program i uruchom go. Co zauważasz?

Warto zapamiętać

- Jeśli w programie w instrukcji przypisania nadamy zmiennej nową wartość, to poprzednia zostanie usunięta.
- Technika iteracji ma zastosowanie do czynności powtarzalnych. Iteracja oznacza powtarzanie tej samej operacji lub ciągu operacji.
- Iterację implementujemy, stosując tzw. pętlę. Z pętlą mamy do czynienia, gdy w pewnym kroku algorytmu wracamy do jednego z wcześniejszych kroków, co powoduje, że pewne kroki algorytmu zostaną wykonane wiele razy.
- W językach programowania dostępne są instrukcje iteracyjne (instrukcje pętli), z których korzystamy, aby zapisać algorytm iteracyjny w postaci programu.
- Jeśli instrukcje wewnątrz pętli mają zostać wykonane zadaną liczbę razy, stosujemy pętlę **`for`**.
- W języku C++ zmienna sterująca określa liczbę kroków iteracji (powtórzeń).
- W języku Python liczbę iteracji określa długość *listy_wartości* po słowie **`in`** w instrukcji **`for`**.
- Zapętlenie programu nastąpi wtedy, gdy nie określimy prawidłowo warunku zakończenia algorytmu.
- W programach komputerowych można stosować zagnieżdżone instrukcje iteracyjne. W przypadku dwóch zagnieżdżonych instrukcji **`for`** liczba kroków iteracji to iloczyn liczby iteracji zewnętrznej pętli i liczby iteracji wewnętrznej pętli.

Pytania i polecenia

1. Na czym polega iteracja?
2. Kiedy mamy do czynienia z pętlą?
3. Omów zastosowanie iteracji na przykładzie sumowania ośmiu liczb.
4. Wyjaśnij na przykładach stosowanie instrukcji iteracyjnej **`for`** w wybranym języku programowania.
5. Do czego służy zmienna sterująca w instrukcji **`for`** w języku C++?
6. Co określa liczbę iteracji w instrukcji **`for`** w języku Python?
7. Przedstaw, na czym polega zagnieżdżanie się instrukcji pętli.
8. Na podstawie następujących fragmentów programów napisz, jakie są wyniki ich działania:

a. C++

```
for(i = 0; i <= 9; i++)
    cout << "$";
```

Python

```
for i in range(10):
    print("$", end = "")
```

b. C++

```
for(i = 0; i <= 9; i++)
    cout << "$" << endl;
```

Python

```
for i in range(10):
    print("$")
```

9. C++: Ile razy będzie wykona instrukcja `cout << i`? Dla jakich wartości zmiennej `i` zostanie wykonana instrukcja `cout << i`?

```
for (i = 0; i <= 20; i++)
    cout << i;
```

Python: Ile razy będzie wykona instrukcja `print(i)`? Dla jakich wartości zmiennej `i` zostanie wykonana instrukcja `print(i)`?

```
for i in range(9, 51):
    print(i)
```

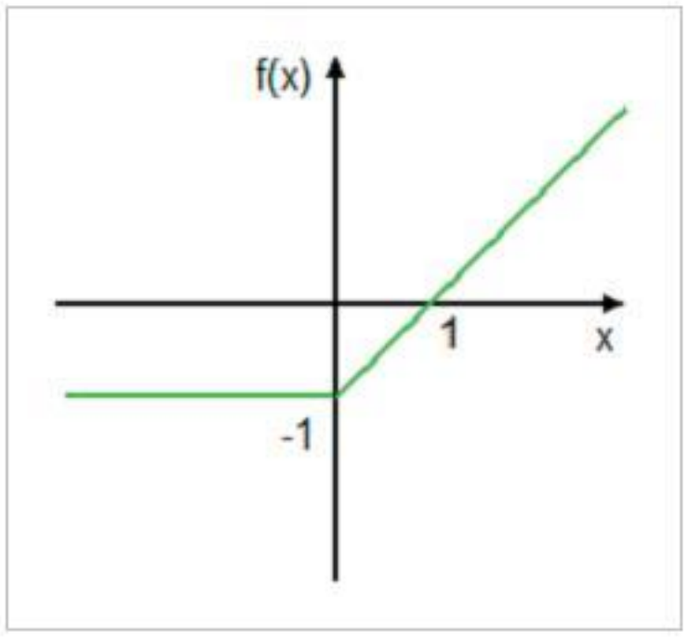
Zadania

Uwagi:

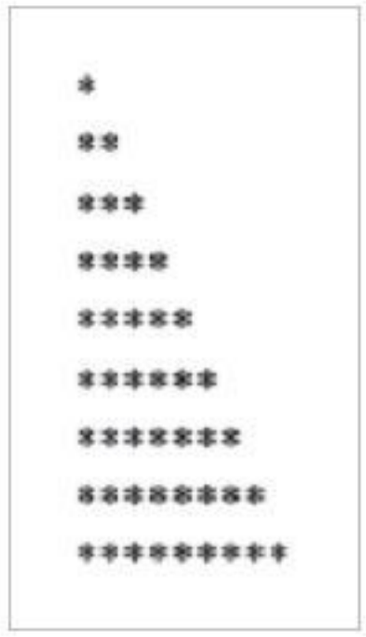
- Staraj się dodawać do programów odpowiednie komunikaty dla użytkownika.
- Pisz przejrzyste programy oraz dodawaj w odpowiednich miejscach komentarze.
- Każdy program uruchom i przetestuj dla różnych danych, nawet jeśli w zadaniu nie ma takiego polecenia (w przypadku języka C++ najpierw skompiluj program).

1. Zmodyfikuj program *Przepis* zapisany w zadaniu 9. z tematu C4 tak, aby obliczenia wykonywane były dla każdego z *n* składników sałatki, gdzie *n* jest wprowadzane z klawiatury. Zapisz plik pod tą samą nazwą.
2. Zmodyfikuj program utworzony w zadaniu 2. z tematu C4. Program powinien dodatkowo zliczać, ile wprowadzono liczb parzystych i na koniec wyprowadzać ich liczbę. Zapisz plik pod tą samą nazwą.
3. Napisz specyfikację zadania, utwórz listę kroków oraz napisz w wybranym języku programowania program obliczania sumy liczb podzielnych przez 7 dla *n* liczb wprowadzanych z klawiatury. Zapisz program w pliku pod nazwą *Podzielne_7_suma*.
4. Napisz program realizujący algorytm obliczania iloczynu *n* dowolnych liczb całkowitych. Wynik iloczynu zapamiętuj w zmiennej *iloczyn* i wyprowadź na ekran. Zapisz program w pliku pod nazwą *Iloczyn_n*.
   Wskazówka: Zmiennej *iloczyn* na początku programu przypisz wartość 1.
5. Napisz specyfikację i program do obliczania osobno sumy liczb dodatnich i liczb ujemnych dla *n* liczb całkowitych wprowadzanych z klawiatury. Zapisz program w pliku pod nazwą *Dodatnie_i_ujemne*.
6. Zmodyfikuj program zapisany w zadaniu 5. Jeśli zostanie wprowadzone z klawiatury zero, wyświetlaj komunikat „Niepoprawne dane". Zapisz plik pod tą samą nazwą.
7. Na podstawie specyfikacji utworzonej w zadaniu 1b z tematu C1 napisz program sprawdzający, czy wśród *n* znaków wprowadzonych z klawiatury (pamiętanych w zmiennej *znak*) dany znak jest samogłoską („a", „e", „i", „o", „u", „y") czy spółgłoską (pozostałe). Zależnie od wyniku, wyprowadzaj napisy: „samogłoska" lub „spółgłoska". Zapisz program w pliku pod nazwą *Litery*.
   Wskazówki:
   - W instrukcji warunkowej należy zastosować złożony warunek z alternatywą.
   - W języku C++ zmienną *znak* zadeklaruj jako typ znakowy: `char znak;`.
     W języku Python wystarczy użyć przypisania
     `znak = input("Wprowadź znak: ")`.
8. W firmie X miesięczna płaca podstawowa jest zwiększana m.in. o kwotę za przepracowane nadgodziny. Jeśli liczba nadgodzin przekroczy 30, to stawka za każdą kolejną nadgodzinę jest zwiększana o 50%. Wprowadzaj liczbę nadgodzin przepracowanych przez jednego pracownika oraz stawkę za jedną nadgodzinę. Oblicz i wyprowadź płacę za przepracowane nadgodziny dla *n* pracowników. Program napisz na podstawie specyfikacji podanej we wskazówce. Zapisz program w pliku pod nazwą *Place*.
   Wskazówka:
   Dane: liczba całkowita *n* – liczba pracowników, liczba całkowita nieujemna *lng* – liczba nadgodzin, liczba rzeczywista dodatnia *s* – stawka za jedną nadgodzinę.
   Wynik: wartość płacy za nadgodziny: *placa*.
9. Napisz program obliczający wartości funkcji przedstawionej na wykresie (rys. 1.). Zapisz program w pliku pod nazwą *Wykres_funkcji*.

10. Napisz program, który umożliwi wyprowadzenie na ekran monitora „choinki” (rys. 2.) składającej się z gwiazdek „*”. Wymiary choinki, czyli liczba gwiazdek składających się na podstawę i wysokość choinki, wprowadzane są z klawiatury. Zapisz program w pliku pod nazwą *Choinka*.



Rys. 1. *Wykres funkcji – zadanie 9.*



Rys. 2. *Układ gwiazdek – wynik działania programu dla podstawy 9 i wysokości 9 gwiazdek – zadanie 10.*

11. Napisz program realizujący algorytm, który umożliwi wyprowadzenie na ekran monitora prostokąta o bokach *n* x *m*, składającego się z liter „A” i „B”, występujących co drugi wiersz (*m* – liczba znaków „A” lub „ B” w poziomie, *n* – liczba znaków „A” lub „B” w pionie). Zapisz program w pliku pod nazwą *AB*.

Dla zainteresowanych

12. Iloczyn *n* kolejnych liczb naturalnych nazywamy silnią: $n! = 1 \cdot 2 \cdot \ldots \cdot n$. Napisz program obliczający silnię liczby *n* wprowadzanej z klawiatury, gdzie $n \leq 20$.
13. Zmodyfikuj program utworzony w zadaniu 10. z tematu C4 tak, aby warunek trójkąta był sprawdzany dla *n* trójek liczb wprowadzanych z klawiatury. Zapisz plik pod tą samą nazwą.
14. Zmodyfikuj program utworzony w zadaniu 11. z tematu C4 tak, aby warunek dla trójkąta prostokątnego był sprawdzany dla *n* trójek liczb wprowadzanych z klawiatury. Zapisz plik pod tą samą nazwą.
15. Napisz program obliczający pierwiastki równania kwadratowego $ax^2 + bx + c = 0$. Zapisz program w pliku pod nazwą *Rownanie_kwadratowe*.
16. Zmodyfikuj program z zadania 15., aby poprawnie reagował na błędne dane ($a = 0$), oraz rozróżniał sytuacje: $\Delta = 0$ i $\Delta > 0$. Zapisz plik pod tą samą nazwą.
17. Napisz program wypisujący na ekranie tabliczkę mnożenia liczb naturalnych od 1 do 10. Zapisz program w pliku pod nazwą *Tabliczka*.
18. Napisz program obliczający sumę: 1 + 11 + 111 + 1111 + … + 1…1 (*n* składników), gdzie *n* wprowadzane jest z klawiatury. Zapisz program w pliku pod nazwą *Jedynki*.
19. Napisz program realizujący algorytm, który umożliwi wyprowadzenie na ekran monitora „szachownicy” składającej się z zer i jedynek. Zapisz program w pliku pod nazwą *Szachownica*.
20. Napisz program tworzący na ekranie obraz składający się z gwiazdek lub innych znaków – według własnego pomysłu.
21. Zaproponuj własne zadanie, w którym zastosujesz poznane techniki: warunki i iteracje. Zapisz rozwiązania w wybranym języku programowania.

Wokół Internetu i projektów

D

Moduł

Temat D1. Internet jako ocean informacji

Temat D2. Korzystanie z wybranych e-usług

Temat D3. Zadania projektowe



Temat D1. Internet jako ocean informacji

1. Piramida rozwoju Internetu
2. Organizacja informacji w WWW
3. Przykłady wyszukiwania informacji
   - 3.1. Korzystamy z encyklopedii i słowników
   - 3.2. Wyszukujemy informacje zapisane w innych językach
   - 3.3. Szukamy szybciej i sprawniej
   - 3.4. Oglądamy mapy i podróżujemy z Internetem
4. Praktyczny poradnik

Warto powtórzyć

1. Jaki typ programu komputerowego umożliwia przeglądanie stron internetowych? Podaj nazwę przykładowego programu.
2. Dlaczego Internet nazywany jest „światową siecią sieci”?
3. Jakie znasz usługi internetowe?

1. Piramida rozwoju Internetu

Na początku lat 60. XX wieku, na zlecenie Departamentu Obrony USA, grupa amerykańskich naukowców podjęła prace badawcze nad koncepcją sieci komputerowej o dużej odporności na uszkodzenia. Pierwotna nazwa tej sieci brzmiała **ARPANET** (z ang. *Advanced Research Projects Agency Network*). Początkowo dostęp do sieci był zastrzeżony na użytek armii. Z czasem do Arpanetu dołączono uczelnie, jednostki naukowo-badawcze, urzędy administracji publicznej.



Internet *– jak ocean – jest rozległy i ciągle się zmienia. Nikt nie jest tu prawdziwym mistrzem, nikt nie wie o nim wszystkiego. Ale, podobnie jak mistrz nawigacji, możesz sporządzić mapę znanych mórz i wyliczyć, jak bezpiecznie żeglować po tych nieznanych.*

G.H. Cady, P. McGregor, *Internet*, Komputerowa Oficyna Wydawnicza „HELP”, Warszawa 1995, s. 17.

Podstawowym elementem składowym Arpanetu był protokół transmisji danych TCP/IP – „język", dzięki któremu mogły wymieniać informacje komputery wyposażone w różne systemy operacyjne i podłączone do sieci łączami różnych typów: telefonicznymi, satelitarnymi, światłowodowymi.

W latach 80. XX wieku zbudowano sieć nowoczesnych komunikacyjnych **linii szkieletowych**, łączących duże centra komputerowe. Z czasem dołączyły do nich regionalne sieci komputerowe. W USA i w innych krajach powstały pierwsze firmy komercyjne oferujące odpłatny dostęp do sieci.

Pod koniec lat 80. XX wieku w sieci pracowało już około miliona komputerów. Aby podkreślić globalny zasięg sieci, zaczęto ją nazywać **Internetem**.

Pierwotnym zastosowaniem Internetu było umożliwienie użytkownikom przesyłania informacji, a także dostępu do programów i danych umieszczonych na specjalnych komputerach nazywanych **serwerami**. W 1971 roku wynaleziono pocztę elektroniczną (**e-mail**). W 1973 roku powstał protokół **FTP** używany do przesyłania plików.

Wśród przeciętnych użytkowników Internet upowszechnił się dzięki pojawieniu się systemu **stron WWW**, który umożliwił im dostęp do różnych serwisów informacyjnych. Niektórzy nawet sądzą, że Internet to właśnie WWW. Chociaż tak nie jest, to usługi świadczone przez WWW tak bardzo zdominowały Internet, że zacierają się granice między tymi pojęciami.

Połączenie komputerów w światową sieć umożliwiło zdalną pracę poprzez protokół komunikacyjny **Telnet** na komputerach oddalonych od siebie czasem o tysiące kilometrów. Powstały grupy dyskusyjne (**USENET**). Pojawiła się również możliwość prowadzenia rozmów w czasie rzeczywistym (**IRC**) oraz internetowych pogaduszek z wykorzystaniem **komunikatorów**.

W latach 90. XX wieku dzięki Internetowi można już było robić zakupy w sklepie internetowym, skorzystać z usług e-banku i prowadzić internetowy pamiętnik (**blog**). Tematycznym uporządkowaniem informacji w Internecie zajęły się **portale**.

W drugiej połowie lat 90. wynaleziono technologię cyfrową **VoIP**, wykorzystującą sieć internetową jako środek transmisji głosu, a następnie także obrazu. Internet stał się również dostępny z poziomu urządzeń przenośnych (najpierw telefonów komórkowych, potem smartfonów i tabletów) – w tym kontekście mówimy o **Internecie mobilnym.**

I **Internet**

Rozległa sieć komputerowa łącząca inne sieci z całego świata. Jest wszechstronnym systemem informacyjnym i komunikacyjnym. Udostępnia informacje ze wszystkich dziedzin życia, zgromadzone na komputerach całego globu oraz umożliwia komunikację między ludźmi nawet z najbardziej odległych miejsc.

Internet – z łac. *inter* (między), z ang. *net* (sieć); w swobodnym tłumaczeniu „międzysieć".

TCP/IP – z ang. *Transmission Control Protocol / Internet Protocol* (protokół kontroli transmisji / protokół internetowy).

FTP – z ang. *File Transfer Protocol* (protokół transmisji plików).

e-mail – z ang. *e – electronic* (elektroniczna), *mail* (poczta).

WWW – z ang. *World Wide Web* (ogólnoświatowa sieć).

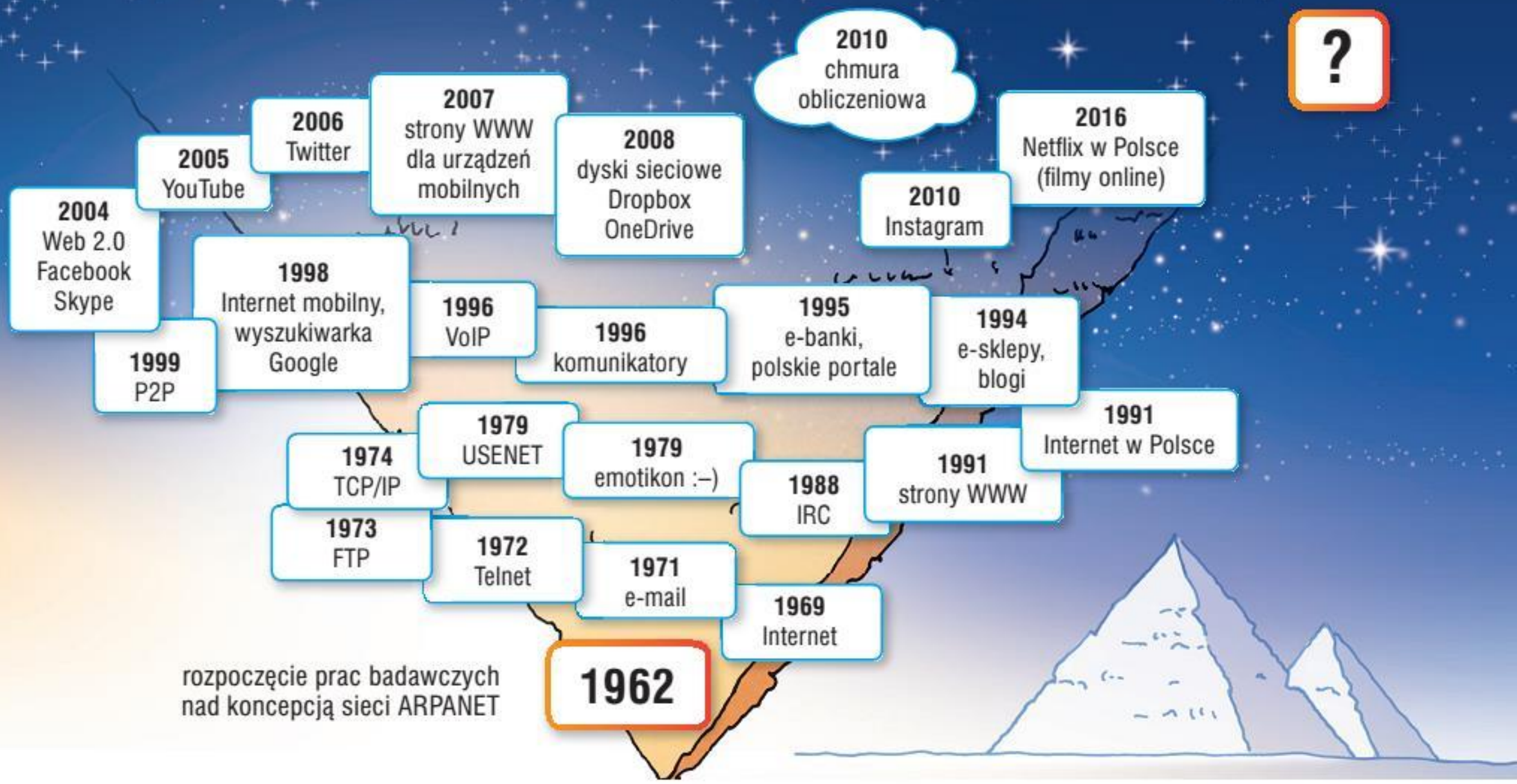
USENET – z ang. *USEr NETwork* (sieć użytkowników).

IRC – z ang. *Internet Relay Chat* (rozmowa w czasie rzeczywistym).

Blog – z ang. *web log* (dziennik sieciowy).

VoIP – z ang. *Voice over Internet Protocol* (telefonia internetowa wykorzystująca przesyłanie głosu za pomocą protokołu IP).

P2P – z ang. *peer-to-peer* (każdy z każdym).



Rys. 1. *Piramida rozwoju Internetu*

Hipertekst H

Sposób organizacji danych (w szczególności tekstowych) polegający na połączeniu niezależnych fragmentów tekstu hiperłączami.

Hiperłącze H
(ang. *hyperlink* – inaczej: odsyłacz, łącze, link)

Umieszczone w dokumencie komputerowym odwołanie do innego dokumentu (hipertekstowego, tekstowego, graficznego, animacji, wideo) lub innego miejsca w tym samym dokumencie.

Adres IP A

Unikatowy numer przyporządkowany urządzeniom sieci komputerowych (pracującym zarówno w Internecie, jak i w sieciach lokalnych) funkcjonujących na podstawie protokołu IP.

Pod koniec XX wieku powstała koncepcja, aby każdy komputer pełnił jednocześnie funkcję **serwera** (udostępniającego zasoby) i **klienta** (korzystającego z zasobów serwera). W ten sposób każdy plik mógłby znajdować się na wielu komputerach i być pobierany z wielu źródeł. Tak narodził się model komunikacji **P2P**, który gwarantuje równorzędne prawa wszystkim komputerom wymieniającym dane (w przeciwieństwie do modelu klient-serwer).

Na początku XXI wieku Internet zdominowały serwisy **Web 2.0**. Ich działanie opiera się na interakcji użytkowników polegającej na współtworzeniu treści serwisu. Wokół serwisów tworzą się rozbudowane społeczności z coraz większymi możliwościami nawiązywania kontaktów.

W 2004 roku powstał Facebook – serwis społecznościowy, którego liczba użytkowników przekroczyła już 2 miliardy. Serwis ten zyskał wielu zwolenników również w Polsce – od roku 2008 roku funkcjonuje polska wersja językowa serwisu. Poza Facebookiem powstały również inne serwisy społecznościowe, m.in.: Twitter (2006 r.), Instagram (2010 r.).

Piramida rozwoju Internetu, na której pokazano wybrane usługi i wydarzenia związane z Internetem (rys. 1.) sięgnęła… chmur. Możliwe stało się **przetwarzanie danych w chmurze obliczeniowej** (ang. *cloud computing*), czyli w systemie połączonych komputerów i pamięci masowych, do których dostęp odbywa się przez Internet. W tym przypadku programy nie są uruchamiane na komputerze użyt-

kownika, ale przechowywane na serwerze internetowym i dostępne przez sieć. Także dane są przechowywane nie na komputerze użytkownika, tylko w chmurze, dzięki czemu dostęp do nich można uzyskać z dowolnego komputera w każdym miejscu na świecie.

2. Organizacja informacji w WWW

WWW to system prezentacji informacji oparty na **hipertekście** i **hiperłączach**, dający możliwość korzystania z informacji umieszczonych na komputerach (zwanych **serwerami internetowymi**) znajdujących się w sieci Internet.



Strona WWW jest zbiorem powiązanych ze sobą dokumentów tekstowych, hipertekstowych, graficznych, dźwiękowych lub zawierających animacje, dostępnych poprzez przeglądarkę internetową.

Dostęp do stron WWW umożliwia protokół komunikacyjny **HTTP**.



Pojęcie „serwer internetowy" oznacza:

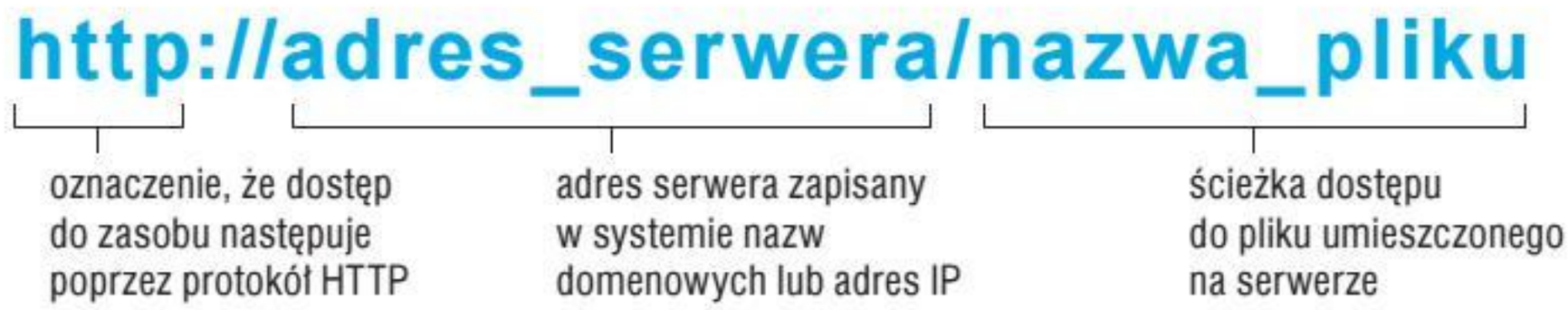
- każde urządzenie (zazwyczaj komputer) pracujące w sieci, udostępniające określone usługi,

lub

- specjalne oprogramowanie uruchomione na tym sprzęcie, np. **serwer HTTP**, **FTP** i inne.

Każda strona WWW ma swój **adres internetowy**, który jednoznacznie ją identyfikuje. Do oznaczania adresów stron stosuje się standard **URL**.

Dla ułatwienia identyfikacji poszczególnych urządzeń w Internecie przyjęto jednolity system nazewnictwa, zwany **systemem nazw domenowych** (**DNS** – z ang. *Domain Name System*). Nazwa domenowa składa się z kilku członów literowych, oddzielonych kropkami. System serwerów DNS pozwala na zamianę nazw domenowych na **adresy IP**.



Rys. 2. *Standardowa postać adresu internetowego strony WWW*

S **Serwer HTTP**

Program działający na serwerze internetowym, udostępniający informacje poprzez protokół HTTP.

U **URL**

(z ang. *Uniform Resource Locator* – uniwersalny lokalizator zasobów)

Standard tworzenia nazw zasobów Internetu, jednoznacznie określający adres zasobu i sposób dostępu do niego.

! **Uwaga**

Na początku adresu strony WWW zamiast *http:* może pojawić się *https:* oznaczające połączenie bezpieczne (szyfrowane).

Witryna internetowa (serwis internetowy) jest zbiorem powiązanych ze sobą (przez hiperłącza) stron WWW, umieszczonych pod tym samym adresem internetowym.

Witryna (serwis) zawiera jedną stronę główną, tzw. stronę startową. Na niej umieszczone są hiperłącza, które umożliwiają przejście do innych części (innych dokumentów) tej samej witryny (serwisu).

Ćwiczenie 1. Korzystamy z odsyłaczy na stronie internetowej

Uruchom przeglądarkę internetową. Wejdź na stronę *http://www.sejm.gov.pl/*. Korzystając z odsyłaczy, zapoznaj się z organami Sejmu RP.

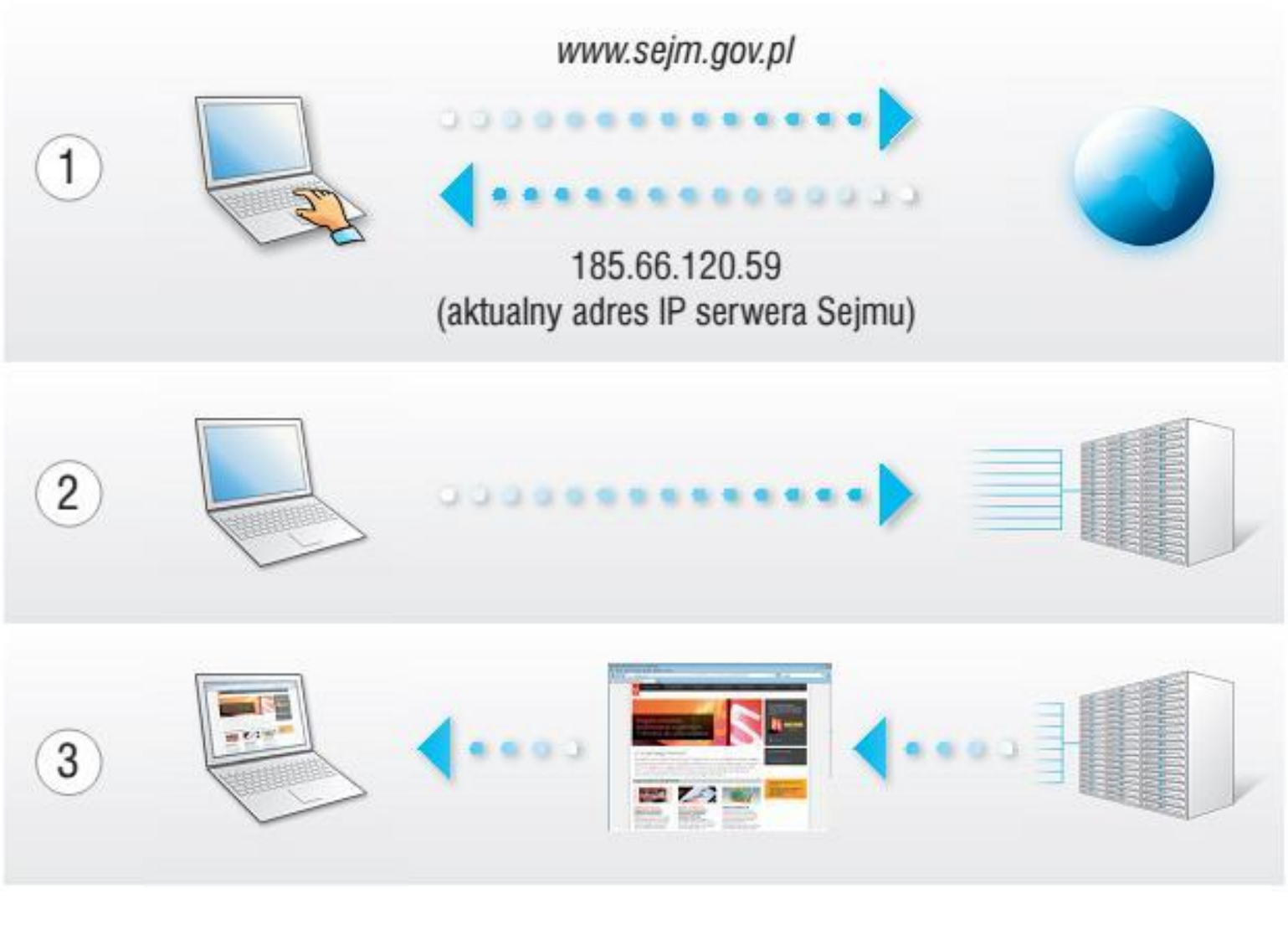
Aby załadować wskazaną przez użytkownika stronę WWW, przeglądarka internetowa łączy się z serwerem WWW.

Uwaga

Komunikat „nie można wyświetlić strony" może być spowodowany różnymi przyczynami, np. chwilowym brakiem dostępu do serwera lub błędem we wpisanym adresie.

Przykład 1. Przeglądanie strony internetowej

1. Po wpisaniu adresu strony internetowej w pasku adresu przeglądarka sprawdza, czy w sieci WWW istnieje żądany serwer i jaki jest jego adres IP.
2. Gdy serwer zostanie odnaleziony, przeglądarka wysyła żądanie pobrania danej strony.
3. Jeśli odpowiednia strona zostanie znaleziona, serwer wysyła ją do naszego komputera. W przeciwnym wypadku wysyłany jest odpowiedni komunikat.



Rys. 3. *Operacje wykonywane przez komputer podczas przeglądania strony internetowej*

3. Przykłady wyszukiwania informacji

Chcemy znaleźć biografię Stefana Żeromskiego, pliki arkusza kalkulacyjnego z rozszerzeniem *xlsx* z wykresami funkcji liniowej, informacje o smartfonach z ostatnich trzech miesięcy, a także dowiedzieć się, ile będzie trwała podróż samochodem z Wrocławia do Suwałk. Jak można sprawnie i szybko znaleźć odpowiednie informacje?

3.1. Korzystamy z encyklopedii i słowników

Jeśli chcemy poznać definicję jakiegoś określenia, biografię znanej osoby, znaleźć informacje dotyczące obiektu geograficznego itp., najlepiej skorzystać z **encyklopedii internetowej**, np. Wikipedii (*pl.wikipedia.org).* Wikipedia jest encyklopedią otwartą, redagowaną przez użytkowników Internetu z całego świata. Każdy może zostać redaktorem Wikipedii – wystarczy do tego przeglądarka internetowa, w której edytuje się treść haseł. Aby nie wprowadzać do internetowej encyklopedii niepełnych lub nieprawdziwych informacji, konieczne jest jednak posiadanie odpowiedniej wiedzy z jakiejś dziedziny. Należy pamiętać, że Wikipedia, jako encyklopedia redagowana nie przez specjalistów, lecz zwykłych użytkowników, może zawierać informacje nie zawsze poprawne i prawdziwe. Jak jednak pokazują niezależne badania, jakość merytoryczna Wikipedii nie jest niższa niż jakość tradycyjnych encyklopedii.

W Internecie dostępnych jest wiele słowników, np. ortograficzny, języka polskiego, antonimów, synonimów, wyrazów obcych, języków obcych. Aby wyjaśnić znaczenie konkretnego słowa, można skorzystać np. ze *Słownika języka polskiego* (*sjp.pwn.pl*).

Ćwiczenie 2. Wyszukujemy informacje w encyklopedii i słownikach internetowych

1. Poszukaj w encyklopedii internetowej biografii Stefana Żeromskiego oraz informacji na temat instrumentu muzycznego o nazwie „bajan".
2. Poszukaj w *Słowniku języka polskiego* znaczenia słowa „bifurkacja". Sprawdź również znaczenie tego słowa w Wikipedii.
3. Odszukaj w wybranym słowniku komputerowym znaczenie określenia „adres IP".
4. Znajdź w *Słowniku języka polskiego* znaczenie słów „synonim" i „antonim". Odszukaj w wybranym słowniku (lub słownikach) synonim i antonim słowa „aseptyczny".
5. Znajdź w *Słowniku języka polskiego* znaczenie słów: „pleonazm", „oksymoron".

3.2. Wyszukujemy informacje zapisane w innych językach

Przed skorzystaniem z konkretnej wyszukiwarki (np. *www.google.pl*, *www.bing.com*) warto zapoznać się z właściwymi dla niej zasadami formułowania zapytania i innymi możliwościami (można je znaleźć w **Pomocy** dołączonej do wyszukiwarki).

Przykłady wyszukiwania informacji pokażemy z wykorzystaniem jednej z popularniejszych wyszukiwarek, czyli Google (*www.google.pl*).

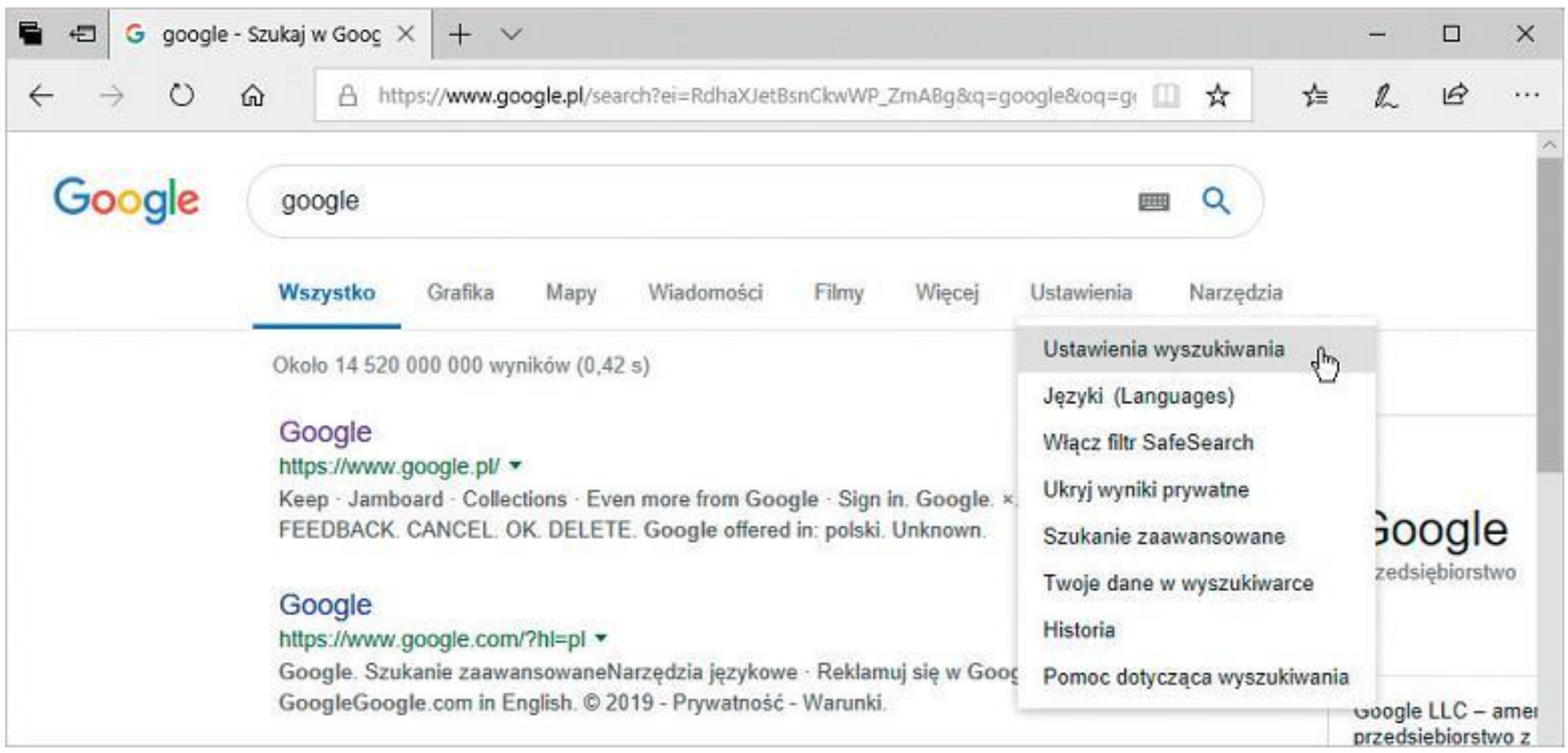
Za pomocą wyszukiwarki Google można m.in.:

- szukać stron zapisanych w innych językach,
- przetłumaczyć stronę zapisaną w języku obcym na język polski lub odwrotnie,
- odwiedzić stronę wyszukiwarki Google w innym kraju.

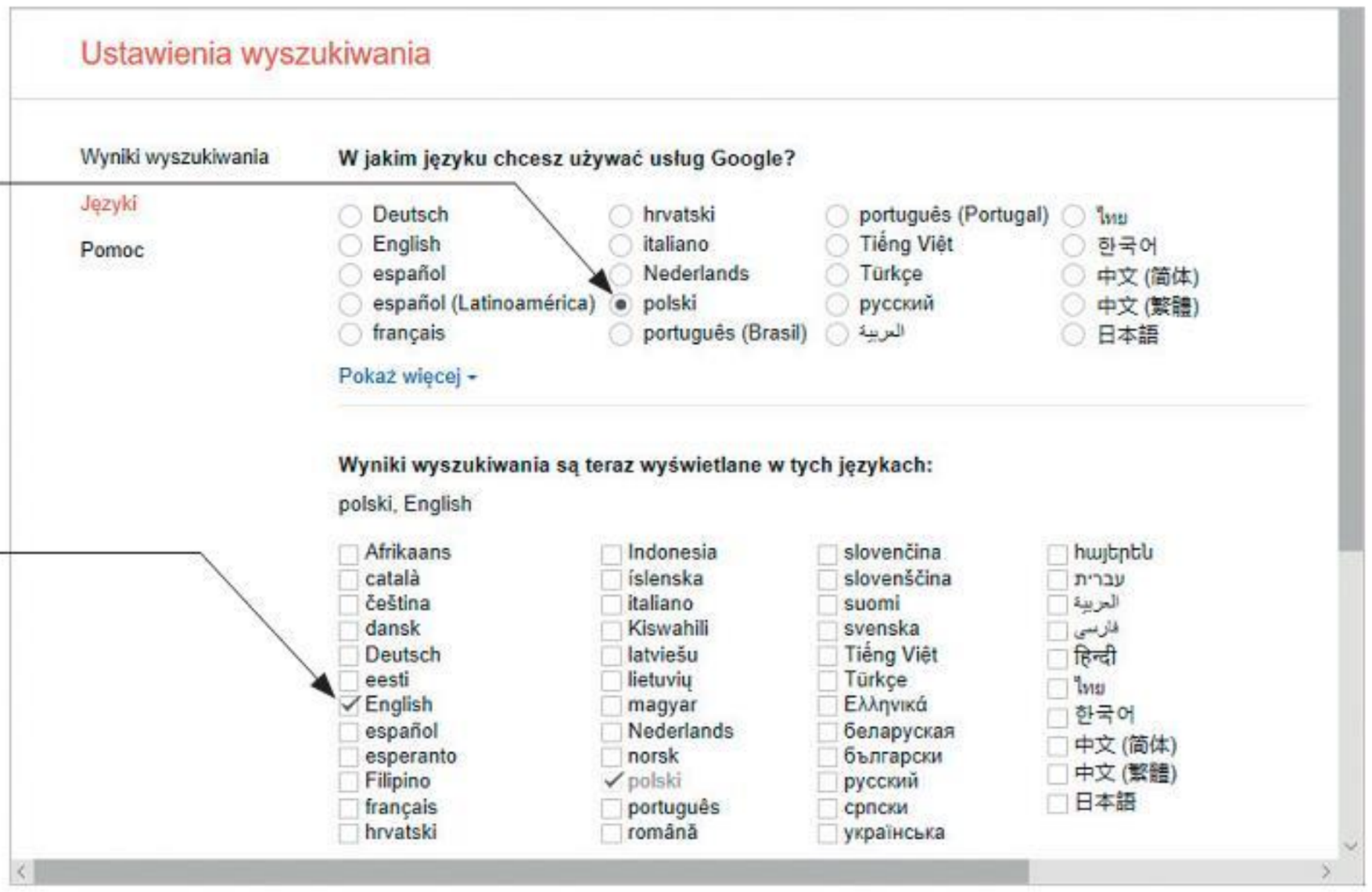
Aby nazwy opcji i innych usług w wyszukiwarce wyświetlały się w wybranym języku, należy w oknie wyszukiwarki odszukać i wybrać opcję **Ustawienia wyszukiwania** (rys. 4a.), a następnie kliknąć link **Języki** i wybrać język, np. angielski (rys. 4b). W tym przypadku, po zapisaniu zmian ustawień wyszukiwarki, nazwy usług i opcji będą wyświetlały się w języku angielskim.

Przykład 2. Szukanie stron zapisanych w innym języku

Jeśli wpiszemy hasło w obcym języku (bez wcześniejszego wybierania języków, w których mają być pokazywane wyniki wyszukiwania), przeglądarka domyślnie wyszuka też strony w języku, w którym hasło zostało wpisane.

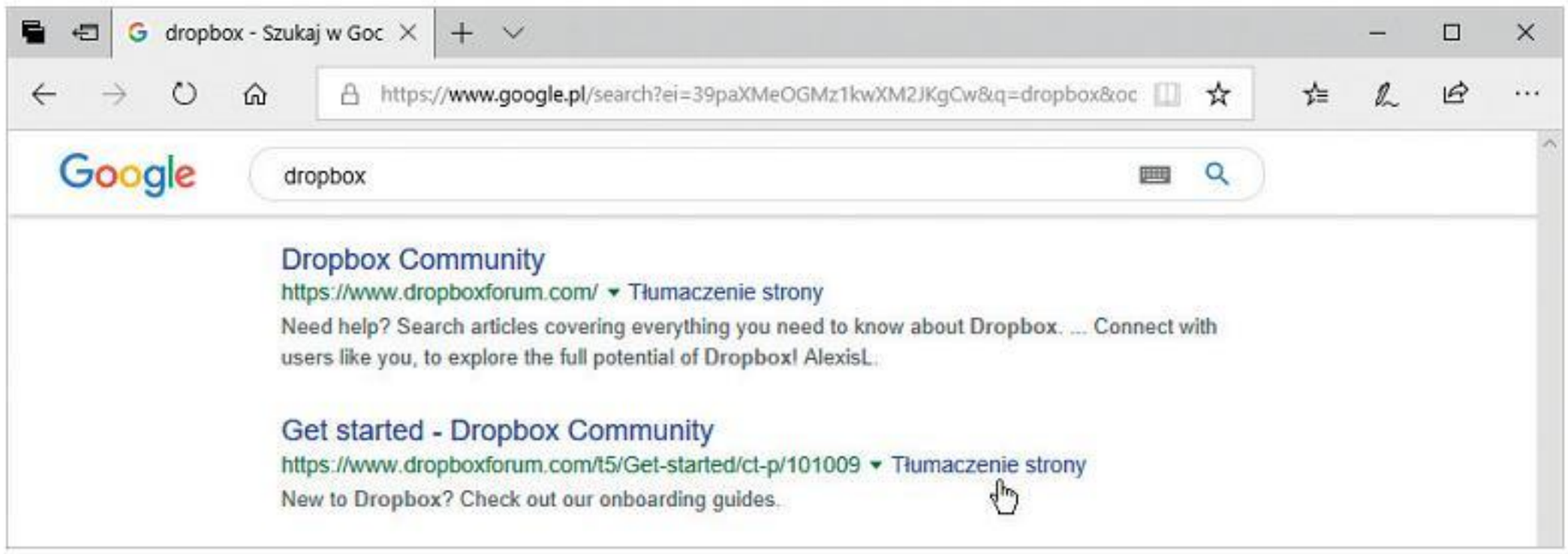


Rys. 4a. *Wybieranie opcji wyszukiwarki Google –* ***Ustawienia wyszukiwania***



Rys. 4b. *Wybieranie w* ***Ustawieniach wyszukiwania*** *języka używania usług Google i dodatkowych języków wyników wyszukiwania*

Stronę, która została wyświetlona w innym języku, można przetłumaczyć na używany język, klikając link **Tłumaczenie strony** obok adresu wyszukanej strony (rys. 5.).



Rys. 5. *Przykładowy wynik wyszukiwania stron internetowych w języku angielskim*

W wyszukiwarce Google tłumaczeniem tekstów zajmuje się **Tłumacz Google** (*translate.google.pl*) – serwis internetowy, umożliwiający tłumaczenie tekstów i całych stron internetowych na różne języki.

Ćwiczenie 3. Tłumaczymy stronę internetową z języka angielskiego na język polski

1. Korzystając z wyszukiwarki Google, wyszukaj adresy stron internetowych z informacjami w języku angielskim na temat usługi Dropbox.
2. Wyświetl tłumaczenie wybranej strony na język polski.

Ćwiczenie 4. Korzystamy z Tłumacza Google

1. Wykorzystaj możliwość tłumaczenia zdań i przetłumacz na język polski zdania zapisane w językach:
   a. angielskim (*The Internet is a vast computer network*),
   b. niemieckim (*Das Internet ist ein riesiges Computer-Netzwerk*),
   c. norweskim (*Internett er et enormt datanettverk*).
2. Przetłumacz zdanie „Być albo nie być – oto jest pytanie" na języki: angielski, duński, hiszpański, niemiecki i chiński. Odsłuchaj tłumaczenie. Jeśli nie wiesz, z jakiego utworu pochodzi ten cytat, znajdź odpowiedź w Internecie.

Ćwiczenie 5. Szukamy informacji na stronach zapisanych w różnych językach

1. Poszukaj informacji na temat czynnych wulkanów w Europie. Przeszukaj strony w języku angielskim i włoskim. Wyświetl (jeśli potrzebujesz) tłumaczenie wybranych stron na język polski.
2. Podaj nazwy trzech wulkanów, które spełniają kryteria wyszukiwania. Wymień państwa, w których znajdują się te wulkany.
3. Odszukaj, gdzie i kiedy miała miejsce ostatnia erupcja wulkanu w Europie. Podaj nazwę tego wulkanu i kraju, na którego terenie się znajduje.

Wskazówka: Najpierw skorzystaj z Tłumacza Google do przetłumaczenia frazy „czynne wulkany w Europie" na wybrany język.

3.3. Szukamy szybciej i sprawniej

W celu usprawnienia i przyspieszenia wyszukiwania można skorzystać z wyszukiwania zaawansowanego.

Wybierając opcję **Szukanie zaawansowane** (rys. 4a), pokazaną na przykładzie wyszukiwarki Google, możemy zmieniać różne opcje wyszukiwania. Dzięki zastosowaniu zaawansowanego wyszukiwania możliwe jest m.in. wyszukanie stron zmodyfikowanych w ostatnim miesiącu, ograniczenie wyszukiwania tylko do tytułów stron lub wybranie plików tylko określonego formatu, m.in. dokumentów PDF czy dokumentów Microsoft Office (np. z rozszerzeniami *rtf*, *doc*, *xls*, *ppt*).

Przykład 3. Wyszukiwanie informacji zapisanych w plikach określonego formatu

Poszukujemy dokumentów tekstowych (zapisanych z rozszerzeniem *doc*) na temat historii komputerów.

Jeśli chcemy zawęzić wyszukiwanie do plików określonego typu, należy:

- na stronie **Szukanie zaawansowane** wpisać hasło (tu frazę: *historia komputerów*), a w polu **Typ pliku** wybrać odpowiedni format pliku – *Microsoft Word (.doc)*

lub

- na stronie głównej wyszukiwarki Google w polu wyszukiwania wpisać słowo *filetype*, a po dwukropku – rozszerzenie i wyszukiwane hasło, np.: *filetype:doc „historia komputerów”*.

W wynikach wyszukiwania otrzymamy dokumenty programu Microsoft Word zawierające podaną frazę.

Uwaga: Aby wyszukać pliki określonego typu zapisane w nowszych wersjach programów Microsoft Office (wersja 2007 lub kolejne), w polu wyszukiwania można wpisać np.: *filetype:xlsx „Funkcja kwadratowa”*, *filetype:docx „historia komputerów”*, *filetype:pptx „Internet”*.

Ćwiczenie 6. Szukamy plików arkusza kalkulacyjnego

Poszukaj w Internecie plików arkusza kalkulacyjnego Microsoft Excel zawierających przykłady funkcji liniowej.

Ćwiczenie 7. Szukamy plików z określonego przedziału czasowego

Poszukaj w Internecie plików z rozszerzeniem *pdf* z ostatniego miesiąca z zadaniami matematycznymi dla liceum.

Przykład 4. Szukanie grafiki, muzyki i filmów

Aby znaleźć obraz, zdjęcie czy inną grafikę, należy po wpisaniu hasła, np. *„Jan Matejko” + Stańczyk*, kliknąć w oknie wyników wyszukiwarki link **Grafika**.

W wynikach wyszukiwania pokażą się miniatury obrazów. Pod każdą miniaturą podane jest źródło grafiki (adres strony, na której grafika jest zamieszczona). Znaleziony obraz można powiększyć, klikając jego miniaturę.

Aby posłuchać różnych piosenek i obejrzeć krótki film, klikamy odpowiednio link **Filmy**.

Ćwiczenie 8. Szukamy grafiki

Znajdź obraz Jacka Malczewskiego przedstawiający portret Władysława Reymonta. Podaj jeden z uzyskanych adresów.

3.4. Oglądamy mapy i podróżujemy z Internetem

Aby obejrzeć mapy i zdjęcia satelitarne różnych miejsc na świecie, można skorzystać z **serwisu mapowego**.

Po wejściu na stronę takiego serwisu, np. *mapy.google.pl*, należy w polu wyszukiwania wpisać nazwę miejsca (np. nazwę miejscowości), które chcemy zobaczyć. W polu tym można także wpisać: dokładny adres miejsca, współrzędne geograficzne miejsca lub kategorię obiektów (np. *muzeum*, *pizzeria*).

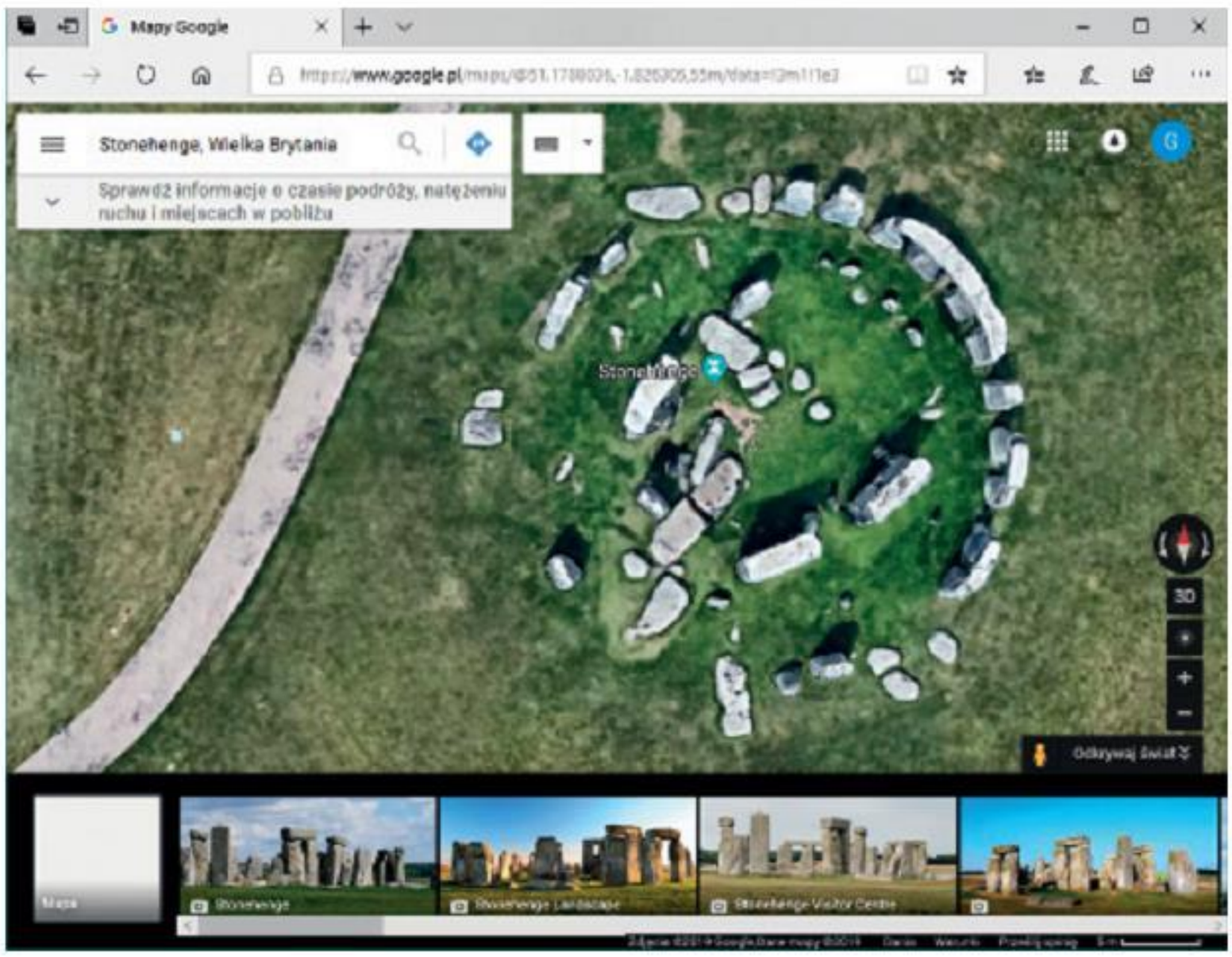
Przykład 5. Oglądanie map i zdjęć satelitarnych

Chcemy obejrzeć zdjęcie satelitarne słynnej budowli znajdującej się w Wielkiej Brytanii – Stonehenge.

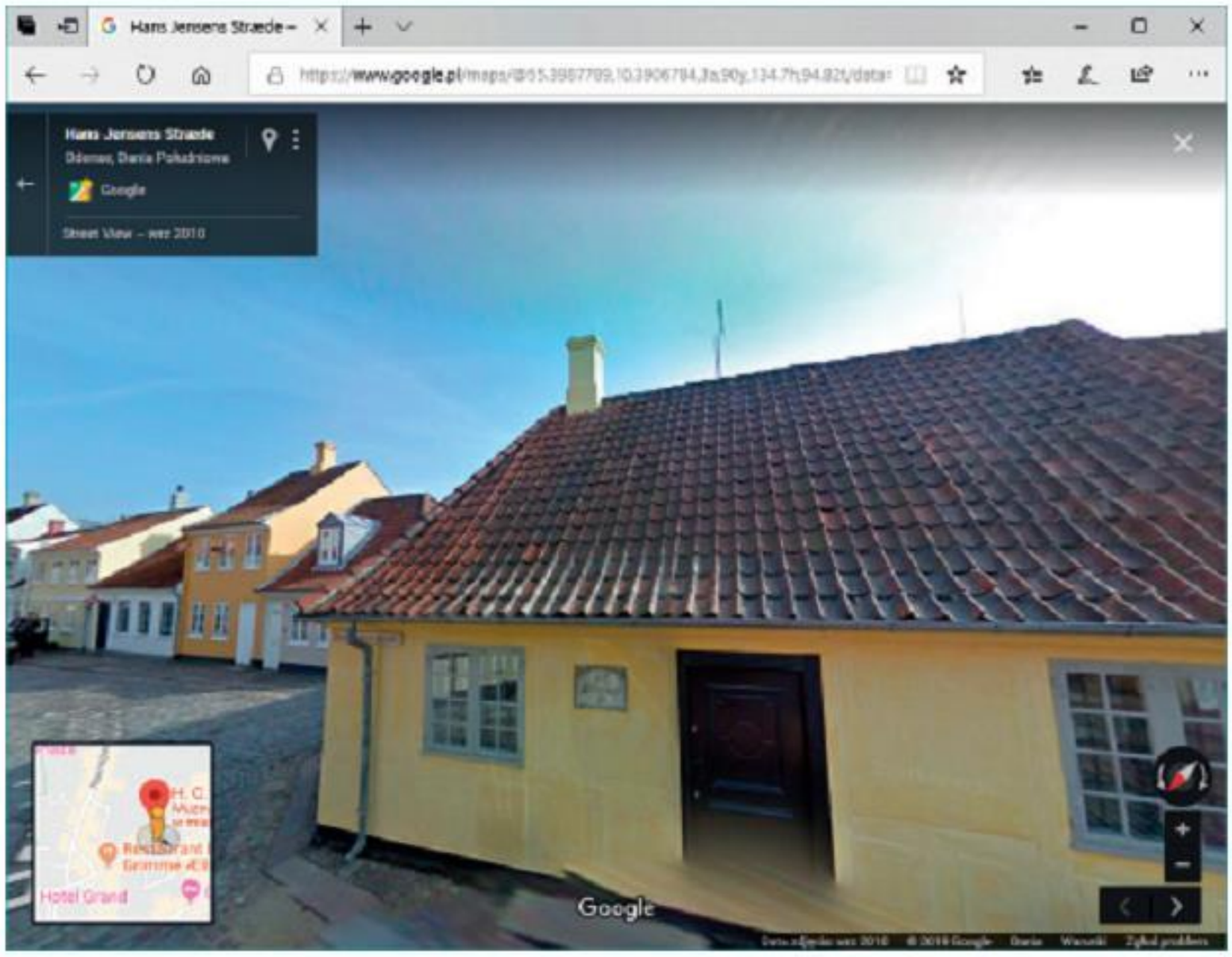
- W polu wyszukiwania należy wpisać: *Stonehenge, Wielka Brytania*.
- Jeśli widoczna jest mapa, należy kliknąć przycisk **Satelita**, aby zobaczyć zdjęcia satelitarne danego obszaru.

Ćwiczenie 9. Korzystamy z serwisu mapowego

1. Odszukaj Stonehenge na mapie satelitarnej (rys. 6.). Znajdź w Wikipedii informacje na temat tej ciekawej budowli. Jest ona wpisana na listę światowego dziedzictwa UNESCO.
2. Odszukaj na zdjęciach satelitarnych dom Hansa Christiana Andersena w Odense (Dania), w którym Andersen mieszkał od 2 do 14 roku życia (rys. 7.). Powiększ obraz i „przespaceruj się" malowniczymi uliczkami Odense.



Rys. 6. *Stonehenge na zdjęciu satelitarnym. Jaką figurę przypomina swoim kształtem ta budowla?*



Rys. 7. *Dom Hansa Christiana Andersena w Odense widoczny na mapie Google w widoku 3D*

Przykład 6. Przeglądanie trasy podróży

Korzystając ze strony *mapy.google.pl*, można również zobaczyć trasę pieszego przejścia lub przejazdu samochodem z wybranego miejsca do miejsca docelowego.
Po wybraniu opcji **Pokaż trasę** i wpisaniu do pola **A** miejsca początkowego trasy, a do pola **B** – miejsca docelowego i zaakceptowaniu danych trasa zostanie pokazana na mapie.
Podany zostanie również przewidywany czas przebycia (przejścia lub przejazdu) i opis (krok po kroku) całej trasy.

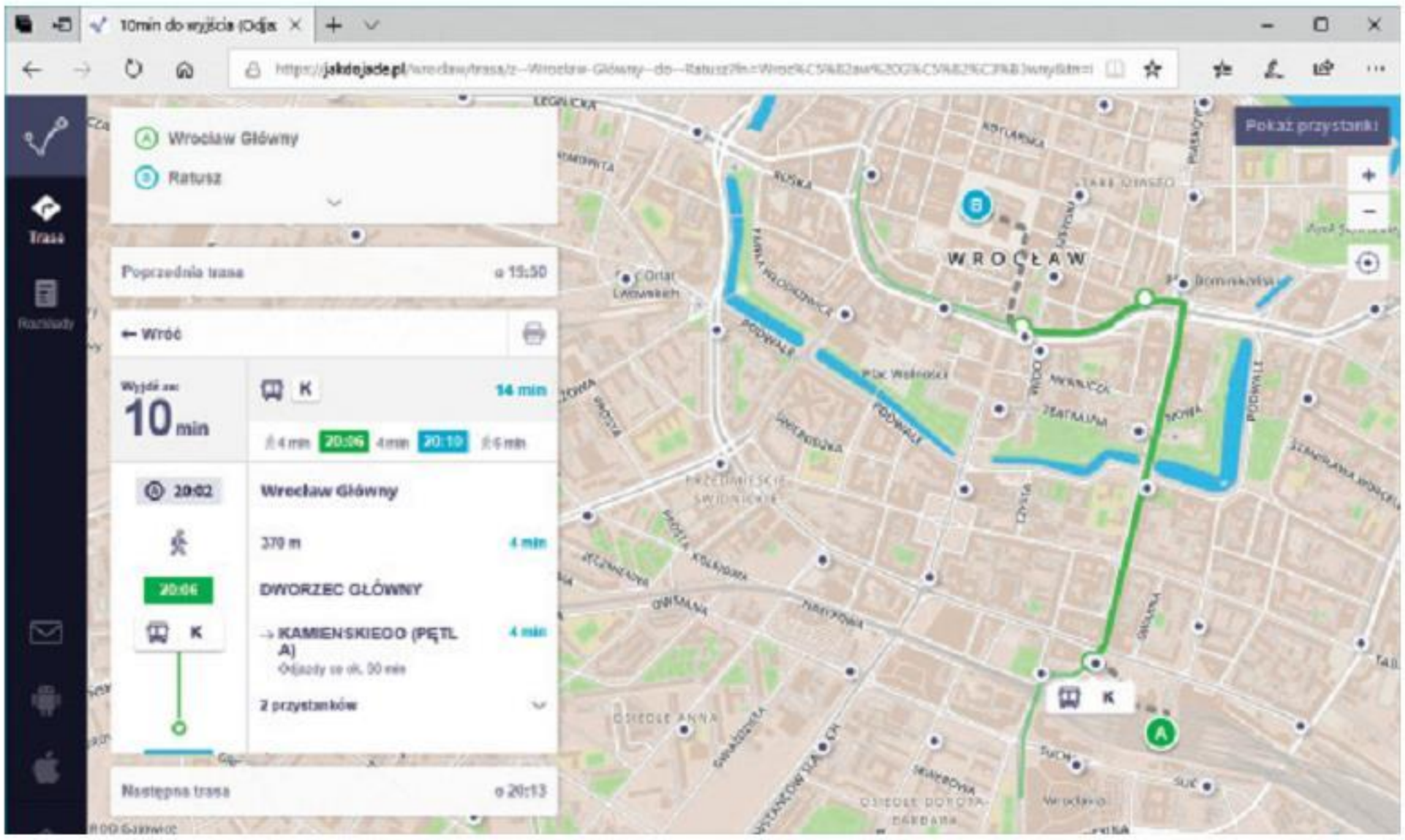
Przykład 7. Szukanie połączeń komunikacji miejskiej

Korzystając z wyszukiwarki połączeń komunikacji miejskiej (*jakdojade.pl*), można znaleźć połączenia w wybranym mieście (rys. 8.).
Wyszukiwarka pokazuje m.in.: linie komunikacji miejskiej (autobusy, tramwaje, metro), ile przystanków trzeba przejechać daną linią, godziny przesiadek.

Uwaga: W Internecie można sprawdzić również połączenia kolejowe, autobusowe, lotnicze i inne.

Ćwiczenie 10. Sprawdzamy czas podróży

1. Na stronie głównej wyszukiwarki *jakdojade.pl* sprawdź, które miasta zostały uwzględnione w serwisie.
2. Sprawdź, ile czasu może zająć ci piesza wędrówka z dworca kolejowego Wrocław Główny do Ratusza na Rynku we Wrocławiu. Sprawdź również, ile zajmie ci pokonanie tej trasy środkami komunikacji miejskiej.



Rys. 8. *Przykładowy opis połączenia komunikacji miejskiej pokazany na stronie* jakdojade.pl

Ćwiczenie 11. Szukamy połączeń komunikacji miejskiej

Planujesz wycieczkę po Krakowie. Sprawdź, jakimi środkami komunikacji miejskiej można dojechać z Wawelu na kopiec Kościuszki.

4. Praktyczny poradnik

Porada 1. Porządkowanie informacji o stronach WWW

Przeglądarki internetowe zapamiętują adresy stron, które ostatnio odwiedzaliśmy. Są one zapisywane w specjalnym folderze *Historia*, otwieranym przez kliknięcie odpowiedniej opcji lub przycisku na pasku narzędzi albo przez rozwinięcie listy wyboru znajdującej się poniżej pola adresowego przeglądarki.

Adresy stron, które często odwiedzamy lub tych, które wydały nam się szczególnie interesujące, możemy zapisać. Adresy są zapisywane w folderze *Ulubione* lub *Zakładki* (zależnie od przeglądarki). Zapisane adresy wyświetlane są w menu przeglądarki, dzięki czemu dotarcie do nich wymaga jedynie wybrania odpowiedniej pozycji z listy. Zakładki można organizować, tworząc w folderze *Ulubione* lub *Zakładki* dodatkowe podfoldery.

Porada 2. Umieszczanie skrótu do strony WWW na pulpicie

Aby umieścić skrót do strony WWW na pulpicie, należy przeciągnąć i upuścić na pulpit ikonę znajdującą się po lewej stronie paska adresu strony internetowej. Okno przeglądarki należy wcześniej zmniejszyć tak, aby pulpit był widoczny.

Aby po zamknięciu okna przeglądarki powrócić do strony, wystarczy kliknąć utworzoną na pulpicie ikonę.

Porada 3. Zapisywanie strony WWW w pliku

Aby zapisać stronę internetową, należy w oknie przeglądarki wybrać opcję menu **Plik/Zapisz stronę jako** – strona zostanie zapisana w pliku z rozszerzeniem *htm* lub *html*.

Porada 4. Sposoby wpisywania adresu internetowego

Zwykle nie ma potrzeby wpisywania pełnego adresu URL witryny (np. *http://www.sejm.gov.pl/*). Można pominąć znajdujący się na początku człon „http://" oraz końcowy znak „/" (czyli wpisać *www.sejm.gov.pl*). Brakujące znaki zostaną dodane automatycznie przez przeglądarkę.

Przeglądarka może być również wyposażona w tzw. automatyczne uzupełnianie adresu. Po wpisaniu fragmentu adresu przeglądarka „domyśla się", co chcemy napisać i próbuje uzupełnić brakującą część. W tym celu wykorzystuje adresy stron, które zostały kiedyś przez nas odwiedzone lub zapamiętane jako ulubione. Jeśli adres zostanie uzupełniony właściwie, wystarczy go zaakceptować.

Jeśli znamy jedynie np. nazwę miasta, szkoły czy zespołu muzycznego, możemy postarać się samodzielnie odgadnąć adres ich strony WWW, próbując różnych podstawowych kombinacji, np. *www.nazwa-zespołu.com*, *www.nazwa-miasta.pl*, *www.nazwa-szkoły.edu.pl*, *www.nazwa-firmy.pl*.

Porada 5. Odświeżanie zawartości strony WWW

Zawartość niektórych stron (np. zawierających prognozę pogody, notowania giełdowe, wyniki zawodów sportowych, egzaminów) może zmieniać się bardzo szybko. Aby zobaczyć aktualną zawartość strony, należy ją odświeżyć.

Z funkcji odświeżania strony możemy również skorzystać, gdy z powodu błędów transmisji danych strona nie załadowała się w całości lub nie została poprawnie wyświetlona.

Jeżeli strona jest bardzo długa i nie chcemy czekać, aby pokazała się w całości, możemy zatrzymać proces ładowania.

Do odświeżania strony i zatrzymywania procesu ładowania służą odpowiednie przyciski na pasku narzędzi przeglądarki internetowej.



Warto zapamiętać

- Historia Internetu zaczęła się w 1969 roku, kiedy zainstalowano pierwsze węzły sieci Arpanet – bezpośredniego przodka dzisiejszego Internetu. Prace nad koncepcją sieci rozpoczęły się na początku lat 60.
- Internet to rozległa sieć komputerowa, łącząca inne sieci z całego świata. Udostępnia informacje zgromadzone na komputerach całego globu oraz umożliwia komunikację między ludźmi.
- Internet udostępnia wiele usług, jednak wyszukiwanie informacji (strony WWW) i ich wymiana (poczta elektroniczna, komunikatory, a przede wszystkim portale społecznościowe) to te najczęściej wykorzystywane.

- Informacji w Internecie możemy szukać, korzystając z internetowych baz danych, np. encyklopedii multimedialnych, słowników, rozkładów jazdy komunikacji miejskiej i zamiejscowej.
- Wyszukiwarki, w tym Google, umożliwiają m.in. szukanie stron zapisanych w innych językach oraz tłumaczenie stron zapisanych w języku obcym na język polski.
- Jeśli wynik wyszukiwania zawiera zbyt długą listę pozycji, możemy ją zawęzić przez wybranie zaawansowanych opcji wyszukiwania, np. wyszukiwania danego słowa tylko w adresie strony czy w jej tytule, określenie typu pliku lub daty ostatniej aktualizacji.
- Korzystając z Internetu, możemy przeglądać mapy i zdjęcia satelitarne różnych miejsc w Polsce i na świecie.

Pytania i polecenia

1. Czym jest Internet?
2. Dlaczego Internet określany jest jako ocean informacji?
3. Jaka jest geneza powstania Internetu?
4. Wyjaśnij opisy umieszczone w prostokątach przedstawiających bloki piramidy (rys. 1.). Jakie inne usługi i wydarzenia związane z Internetem można na niej jeszcze umieścić? Podaj trzy przykłady.
5. Na czym polega organizacja informacji w WWW?
6. Wyjaśnij pojęcie „serwer internetowy".
7. Przedstaw na wybranym przykładzie znaczenie poszczególnych składników adresu internetowego.
8. Czym są hiperłącza?
9. Na czym polega przeglądanie strony WWW?
10. Wyjaśnij na konkretnych przykładach, w jaki sposób można:
    a. znaleźć strony zapisane w innych językach,
    b. wyszukać grafikę,
    c. odszukać informacje umieszczone w plikach z rozszerzeniem *pdf*,
    d. szybko znaleźć biografię znanej osoby,
    e. obejrzeć zdjęcie satelitarne wybranego miejsca na Ziemi.
11. W jaki sposób można uporządkować adresy stron internetowych, które często odwiedzamy? Omów sposób organizacji adresów stron stosowany w przeglądarce internetowej, z której korzystasz.

Zadania

1. Przedstaw według własnego pomysłu rozwój usług internetowych i innych wydarzeń związanych z Internetem.
2. Poszukaj stron internetowych, na których można posłuchać odgłosów wydawanych przez leśne zwierzęta.
3. Korzystając z wyszukiwarki internetowej, odpowiedz na pytania:
    a. Ile powieści napisał Henryk Sienkiewicz?
    b. Ile ścian ma ikosaedr i jakiego są one kształtu?
    c. Jaki jest wzór chemiczny glukozy?
    d. W jaki sposób parzy się kawę po turecku?
    e. W jakich miesiącach kwitną szafirki?
    f. Jakie miasto jest stolicą Zimbabwe?
    g. Gdzie urodził się Cyprian Kamil Norwid?
    h. W jakim stylu malował Alfred Sisley?

i. Jakimi dziedzinami nauki zajmował się Michael Faraday?
j. Ilu graczy liczy drużyna curlingu?
k. Jakie są wymiary najmniejszego komputera osobistego?
l. Ile Nagród Nobla przyznano w dziedzinie literatury? Wymień Polaków, którzy otrzymali tę nagrodę i podaj rok jej przyznania.
m. W jaki sposób przygotowuje się węgierski placek lángos?
n. Gdzie znajduje się najdłuższy tunel drogowy w Europie?
o. W jakim państwie rozgrywa się akcja „Hamleta"?
p. Do jakiego gatunku należy najstarsze drzewo rosnące w Polsce?

4. Odszukaj informacje na temat kursów języka angielskiego prowadzonych w województwie, w którym mieszkasz. Możesz zawęzić obszar poszukiwań do miasta, na terenie którego mieszkasz lub znajdującego się najbliżej twojego miejsca zamieszkania.
5. Odszukaj połączenie kolejowe z Wrocławia do Suwałk z najmniejszą liczbą przesiadek. Podaj godzinę odjazdu pociągu i jego przyjazdu do stacji docelowej. Podaj miejscowości, w których planowane są ewentualne przesiadki, oraz godziny przesiadek. Podaj łączny czas podróży. Pokaż sposób szybkiego, powtórnego wejścia na tę stronę PKP.
6. Poszukaj obrazów Rembrandta, które znajdują się w muzeum w Luwrze. Ile otrzymałeś adresów stron?
7. Poszukaj zdjęć twojego ulubionego sportowca, aktora i piosenkarza.
8. Poszukaj plików arkusza kalkulacyjnego Microsoft Excel zawierających przykłady funkcji kwadratowej.
9. Wpisz w wyszukiwarce w cudzysłowie hasło *wieża Eiffla*, przetłumacz je na język francuski i wyszukaj strony internetowe w języku francuskim. Przetłumacz wybrane strony na język polski. Podaj wysokość wieży Eiffla.
10. Poszukaj w Internecie innej, od podanej w tym temacie, możliwości tłumaczenia zdań i przetłumacz zdanie „WWW to system prezentacji informacji oparty na hipertekście i hiperłączach" na języki: angielski, czeski, niemiecki i francuski.
11. Poszukaj w Internecie prezentacji na temat Internetu wykonanej w programie PowerPoint (plik z rozszerzeniem *ppt* lub *pptx*).
12. Odszukaj w Internecie strony z czasopismami z dziedziny informatyki w wersji elektronicznej. Podaj przykładowe adresy takich stron i tytuły czasopism. Zapoznaj się z zasadami korzystania z nich, sprawdź m.in., czy artykuły z danego czasopisma można czytać bezpłatnie, czy należy uiścić opłatę.
13. Poszukaj w Internecie, jakie polskie obiekty, miejsca, zabytki zostały wpisane na listę dziedzictwa światowego UNESCO.
14. Postaraj się znaleźć na mapie satelitarnej szkołę, do której uczęszczasz.
Wskazówka: W polu wyszukiwania obok nazwy miejscowości można wpisać również nazwę ulicy i numer domu.
15. Odszukaj stronę internetową, na której można znaleźć kody pocztowe poszczególnych miejscowości w Polsce

Dla zainteresowanych

16. Zaczynasz pracować w firmie sprzedającej sprzęt radiowy. Okazuje się, że firma nie ma swojej strony internetowej. Jakich argumentów należałoby użyć, aby przekonać pracodawcę, że obecność w Internecie jest nieodzownym elementem działalności firmy?
17. W pierwotnym zamierzeniu piramida rozwoju Internetu (rys. 1.) miała stać normalnie. Dlaczego ostatecznie piramida została odwrócona? Uzasadnij odpowiedź.

18. Piramidy przetrwały tysiące lat. Jaka jest szansa przetrwania Internetu? Zaprezentuj własny pogląd na ten temat.
19. Poszukaj w Internecie więcej informacji na temat chmury obliczeniowej.
20. Znajdź w Internecie więcej informacji i miejsc związanych z postacią Chansa Christiana Andersena.
21. Poszukaj na stronach w języku angielskim informacji na temat zabytków Londynu. Korzystając ze znalezionych wiadomości, przygotuj w edytorze tekstu krótki opis w języku angielskim trzech wybranych zabytków wraz z ilustracjami pokazującymi ich położenie na mapie miasta.
22. Korzystając z Internetu, odpowiedz na pytania:
    a. Jaki procent osób w Polsce korzysta z Internetu?
    b. Ile stron WWW jest w Internecie?
23. Niektóre wyszukiwarki uwzględniają odmianę wyrazów. Sprawdź, czy używana przez ciebie wyszukiwarka po wpisaniu hasła *książkami* znajdzie również strony zawierające słowa *książki*, *książkę*, *książką* itp.



Przeczytaj, jeśli chcesz wiedzieć więcej…

Na stronie *http://www.internetworldstats.com/stats.htm* można sprawdzić, jaka jest obecnie liczba użytkowników Internetu na świecie.

Wikipedia stała się inspiracją do stworzenia w 2004 roku projektu OpenStreetMap (OSM) – Swobodnej Wiki-Mapy Świata (*www.openstreetmap.org*). Celem tego projektu jest stworzenie darmowej, swobodnie dostępnej mapy całej kuli ziemskiej. Podobnie, jak w Wikipedii, mapę mogą edytować użytkownicy Internetu z całego świata, m.in. na podstawie danych zapisanych przy użyciu odbiorników GPS, zdjęć satelitarnych, map, do których wygasły prawa autorskie, oraz własnej wiedzy na temat edytowanego obszaru.

Symbol @ (ang. *at* – na, przy) znalazł się w adresie e-mail przez przypadek. W 1971 roku twórca poczty elektronicznej, Ray Tomlinson, szukał znaku, który mógłby oddzielić nazwę osoby czy firmy od nazwy komputera (domeny). Miał to być symbol, który nie występuje w nazwiskach, imionach, nazwach instytucji, nie jest literą ani cyfrą. Tomlinson wybrał przypadkowy znak @, myśląc, że z czasem zmieni go na inny, jednak znak się przyjął i jest używany w tej funkcji do dziś.

Znak @ był umieszczony na klawiaturze komputera wcześniej niż pojawiły się pierwsze e-maile. W systemach operacyjnych lat 70. ubiegłego wieku był używany do usuwania znaków. Nazywano go wówczas *line killing* – usuwał bowiem cały wiersz. Potem znak ten stał się bezużyteczny. Dużo wcześniej, bo już w XIX wieku, znak @ pojawił się na pierwszych amerykańskich maszynach do pisania.

Przykłady potocznych określeń symbolu @: w Grecji – kaczuszka, w Chinach – myszka, w Rosji – pieseczek, w Niemczech – świńskie ucho, na Węgrzech – keks, w Czechach – rolmops, w Turcji – róża, w Anglii – ślimak, a w Polsce – małpa, małpka.

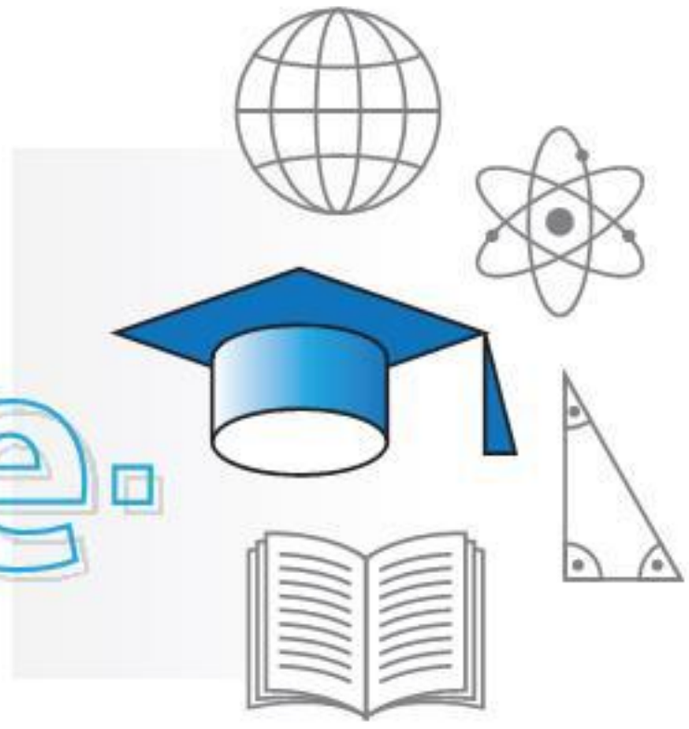
Temat D2. Korzystanie z wybranych e-usług

1. E-nauczanie
2. E-dziennik
3. E-praca
4. E-bank
5. E-zakupy
6. E-aukcje
7. E-podpis

Warto powtórzyć

1. Jakie są najistotniejsze fakty dotyczące rozwoju Internetu?
2. Kiedy powstały pierwsze e-banki?
3. Jakie usługi pojawiły się dzięki rozwojowi technologii informacyjno-komunikacyjnej?

1. E-nauczanie



E-learning polega na prowadzeniu nauczania w dowolnej odległości od szkoły, uczelni czy innej instytucji szkoleniowej z wykorzystaniem sieci komputerowej i Internetu. Jeśli do nauczania na odległość wykorzystuje się urządzenia mobilne, używa się również pojęcia **m-learning** (ang. *mobile learning*).

Korzystając z e-learningu, nie słuchamy wykładu w klasie czy na sali wykładowej, ale zapoznajemy się z nim za pośrednictwem Internetu. Można to zrobić w dowolnym czasie oraz odtworzyć nagranie dowolną liczbę razy. Ćwiczenia i inne materiały dydaktyczne otrzymujemy na nośniku elektronicznym lub pobieramy z Internetu. Swoje rozwiązania zadań, prace zaliczeniowe i projekty wysyłamy prowadzącemu zajęcia, np. za pomocą poczty elektronicznej czy chmury. Dzięki komunikacji internetowej słuchacze mogą kontaktować się z nauczycielami (wykładowcami) również w czasie rzeczywistym, np. prowadząc wideokonferencje.

E-learning może uzupełniać (wspomagać) tradycyjny proces nauczania lub stanowić samodzielną formę szkoleniową (np. kurs języka obcego realizowany tylko w Internecie). Połączenie co najmniej dwóch różnych metod nauczania, np. e-learningu z tradycyjnymi metodami nauki, nazywane jest z języka angielskiego **blended learningiem**. Wybrane szkoły i wyższe uczelnie oferują poza tradycyjnym nauczaniem wiele kursów prowadzonych wyłącznie przez Internet.

Zdalne nauczanie wymaga przygotowania odpowiedniego **środowiska e-learningowego**, które umożliwi uczestnikom dostęp do materiałów dydaktycznych i różnych narzędzi. Można do tego celu wykorzystać na przykład platformę e-learningową **Moodle** (z ang. ***M**odular **O**bject-**O**riented **D**ynamic **L**earning **E**nvironment* – środowisko nauczania zdalnego za pomocą sieci teleinformatycznych) – darmowy program dostępny w Internecie na podstawie licencji GNU GPL.

Zdalne nauczanie umożliwia:

- dowolny wybór miejsca i czasu szkolenia;
- dostosowywanie tempa nauki do własnych możliwości (np. materiały audiowizualne można w dowolnym momencie zatrzymać i przewinąć w celu ponownego obejrzenia);
- indywidualne nadzorowanie postępów każdego ucznia (uczestnika szkolenia, studenta);
- oszczędność kosztów działalności firmy szkoleniowej (np. nie trzeba ponosić kosztu wynajmu sali, transportu, zakwaterowania, wyżywienia);
- oszczędność czasu na dojazdy do miejsca szkolenia;
- ukończenie szkolenia bez konieczności fizycznej obecności w klasie czy na sali wykładowej.

Wątpliwości dotyczące e-nauki

Sceptycy mają jednak wątpliwości, czy zdalne metody nauczania mogą być jedyną formą nauki. Uważają, że uczniowie (studenci) powinni także spotykać się w sali wykładowej (ćwiczeniowej) z kolegami oraz wykładowcami, aby w bezpośrednich kontaktach wymieniać myśli, doświadczenia, dzielić się opiniami.

Często mówi się o wrażeniu odosobnienia, którego doświadczają osoby uczące się na odległość. Może to skutkować trudnościami związanymi z zabieraniem głosu w dyskusji, np. na seminarium naukowym, szkolnym czy zebraniu firmowym. Nie zawsze komunikator, portal społecznościowy, chmura czy wideokonferencja mogą zastąpić bezpośredni kontakt.

Ćwiczenie 1. Szukamy stron internetowych oferujących zdalne nauczanie

Odszukaj w Internecie strony oferujące zdalne nauczanie języka angielskiego. Podaj przynajmniej trzy adresy stron.

2. E-dziennik

E-dziennik to system informatyczny umożliwiający prowadzenie elektronicznych dzienników szkolnych. Oceny zapisywane są w systemie informatycznym szkoły, a następnie udostępnianie przez odpowiednią stronę WWW zainteresowanym osobom – uczniom oraz rodzicom.



Dziennik elektroniczny umożliwia:

- kompleksowe prowadzenie dokumentacji procesu nauczania i innej szkolnej dokumentacji;
- łatwiejsze tworzenie zestawień i statystyk obrazujących jakość pracy ucznia i szkoły;
- znaczne skrócenie czasu przygotowania dokumentów, np. świadectw;
- stały dostęp do informacji o ocenach, frekwencji, zachowaniu oraz tematach zajęć lekcyjnych;
- powiadamianie SMS-em rodziców (opiekunów), np. o otrzymanych ocenach, nieobecności w szkole, dzięki czemu mogą na bieżąco śledzić postępy dziecka, sprawdzać jego frekwencję czy otrzymywać informacje o zebraniach.

Wątpliwości dotyczące e-dziennika:

- ograniczenie bezpośrednich kontaktów nauczycieli z rodzicami (opiekunami) uczniów;
- ograniczenie doraźnych kontaktów nauczycieli, którzy nie wymieniają się już dziennikami papierowymi podczas przerw.

Ćwiczenie 2. Poznajemy dziennik elektroniczny

Dowiedz się więcej na temat wyglądu i funkcjonowania dziennika elektronicznego. Skorzystaj w tym celu z Internetu lub zbierz informacje w swojej szkole (jeśli taki dziennik w niej funkcjonuje).

3. E-praca

E-praca (inaczej **praca na odległość**, **zdalna praca**, telepraca, ang. *teleworking*) polega na wykonywaniu pracy w dowolnej odległości od miejsca zatrudnienia (siedziby firmy).



E-pracownicy kontaktują się ze współpracownikami i przełożonymi za pomocą poczty elektronicznej i innych form komunikacji internetowej, w tym chmury. Praca na odległość była możliwa również przed rozpowszechnieniem Internetu. Wówczas kontaktowano się z pracodawcą przez tradycyjne urządzenia telekomunikacyjne.

Takie zajęcia, jak tworzenie programów komputerowych czy opracowywanie komputerowych dokumentów (tj. raportów, sprawozdań, bilansów, projektów) i ich redagowanie (np. korekty, tłumaczenia), mogą być wykonywane przez pracowników w domu, przy komputerze. Dzięki zastosowaniu technologii informacyjno-komunikacyjnej przepływ materiałów i dokumentów pomiędzy współpracownikami jest płynny.

Telepraca umożliwia:

- wyeliminowanie czasu na dojazdy do pracy, brak spóźnień pracownika;
- oszczędność wydatków firmy na przygotowanie stanowiska pracy (powierzchni, sprzętu, materiałów eksploatacyjnych itp.), choć niektóre firmy wyposażają stanowisko telepracownika;
- zatrudnienie osób niepełnosprawnych, osób z terenów wysokiego bezrobocia, kobiet zajmujących się dziećmi;
- swobodne organizowanie czasu i miejsca pracy przez pracownika, który może realizować obowiązki zawodowe w domu, a także w telecentrum, telechatce, telewiosce; są to miejsca, w których gromadzą się ludzie z lokalnej społeczności, mieszkający blisko siebie, ale daleko od siedziby firmy; w Polsce jeszcze nie stały się popularne.

Wątpliwości dotyczące telepracy:

- pracodawca nie może obserwować stopnia zaangażowania pracowników;
- telepracownik może stać się tzw. pracoholikiem; ważną kwestią staje się samodyscyplina, trzeba umieć samodzielnie organizować sobie czas pracy;
- brak bezpośrednich kontaktów z innymi pracownikami może prowadzić do wyobcowania.

Ćwiczenie 3. Szukamy w Internecie ofert telepracy

1. Odszukaj w Internecie adresy stron internetowych z ofertami e-pracy.
2. Znajdź zdalną pracę dla telemarketera. Sprawdź, czy wśród znalezionych ofert jest również praca dla osoby niepełnosprawnej. Jeśli nie wiesz, czym zajmuje się telemarketer, sprawdź w Internecie.

4. E-bank

W banku elektronicznym wszystkie formalności, począwszy od założenia konta do wykonywania operacji bankowych, możemy załatwić, korzystając z komputera i Internetu lub urządzenia mobilnego, np. smartfonu czy tabletu. Dostęp do e-banku odbywa się przez specjalną stronę WWW lub aplikację w smartfonie bądź tablecie.



Większość tradycyjnych banków ma w swojej ofercie konta z obsługą przez Internet.

W początkowym okresie rozwoju bankowości elektronicznej w Polsce istniały również banki wyłącznie internetowe.

Oferta e-banku jest podobna do tej, którą proponuje bank tradycyjny.

E-banki umożliwiają m.in.:

- sprawdzenie stanu konta;
- płacenie rachunków za e-zakupy, w tym za przedmioty kupione na aukcjach internetowych;
- dokonywanie płatności (np. za energię elektryczną, mieszkanie, Internet);
- zlecanie przelewów;
- zakładanie lokat;
- składanie wniosków o przyznanie kredytu lub karty płatniczej;
- kupowanie jednostek funduszy inwestycyjnych i polis ubezpieczeniowych.

Wymienione operacje można wykonywać przez Internet. Do placówki banku musimy pójść tylko wtedy, gdy chcemy wpłacić na konto gotówkę. Konto można także zasilić, dokonując przelewu z innego konta lub korzystając z **wpłatomatu** (bankomatu służącego do wpłat na konto bankowe).

Banki zachęcają klientów do korzystania z kont internetowych, oferując np. niższe opłaty za wykonanie operacji. Wiele osób obawia się jednak, że w wirtualnej rzeczywistości pieniądze rozpłyną się w nieznany sposób i nie będzie można ich odzyskać. Z tego powodu bardzo ważne jest zadbanie o bezpieczeństwo konta internetowego.



Podstawowym elementem zabezpieczenia konta jest **szyfrowanie danych** przesyłanych pomiędzy naszym komputerem a serwerem banku.
Adresy URL połączeń szyfrowanych posiadają na początku oznaczenie *https* (z ang. *HTTP Secure* – bezpieczne HTTP), zamiast tradycyjnego *http*.

Informacje dotyczące szyfrowania połączenia z daną witryną, a także właściciela witryny można sprawdzić, na przykład klikając ikonę po lewej stronie paska adresu w przeglądarce, przedstawiającą zwykle zamkniętą kłódkę, np.:  ,  .



Aby zapewnić bezpieczeństwo transakcji w e-banku, stosuje się **login** i **hasło** oraz dodatkowe **formy zabezpieczeń.** Podczas logowania podajemy identyfikator użytkownika i hasło. Hasło jest niekiedy **maskowane** – wpisuje się wtedy tylko wybrane znaki hasła.

Token (generator kodów jednorazowych)

Urządzenie elektroniczne, służące do uwierzytelniania transakcji internetowych, najczęściej bankowych.

Zabezpieczenia transakcji stosowane przez e-banki do uwierzytelnienia (potwierdzenia tożsamości) klienta

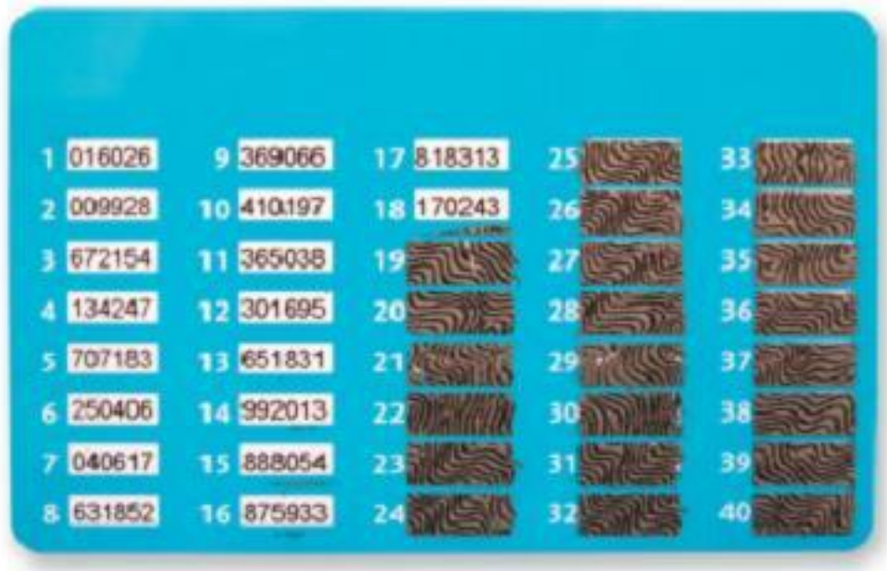
Specjalne urządzenia elektroniczne – do tej grupy zaliczamy tak zwany **token** (rys. 1.). Urządzenie co kilkadziesiąt sekund generuje i pokazuje na wyświetlaczu nowy kod, który trzeba wpisać w odpowiednim polu na stronie internetowej banku. Wpisany kod jest porównywany z kodem wygenerowanym w tym czasie na serwerze w banku. Zgodność kodów umożliwia wykonanie operacji. Niektóre tokeny mają

ponadto dodatkowe formy zabezpieczeń, np. żeby zobaczyć kod, należy wcześniej podać hasło do tokena.

Lista haseł jednorazowych lub **karta kodów jednorazowych** (rys. 2.) – lista, na której umieszczono kilkadziesiąt (np. 50) losowo wygenerowanych kodów, składających się z kombinacji cyfr lub liter i cyfr. Lista może mieć postać zwykłej kartki, na której wymienione są kolejne kody, lub tzw. zdrapki, czyli plastikowego nośnika, na którym zapisane kody stają się widoczne po zdrapaniu warstwy ochronnej.



***Rys. 1.** Przykładowy token z wygenerowanym kodem*



***Rys. 2.** Przykładowa lista haseł jednorazowych w postaci plastikowej karty*

Specjalne aplikacje zainstalowane na telefonach lub SMS-y przesyłane na telefon – ze względu na powszechny dostęp do telefonów komórkowych (smartfonów) banki coraz częściej używają zabezpieczeń transakcji za pomocą telefonu. Są dwa rodzaje takich zabezpieczeń:

1. SMS-y wysyłane na telefon komórkowy,
2. wysyłanie kodu do specjalnej aplikacji zainstalowanej na telefonie.

Zabezpieczenie za pomocą SMS-a jest rozwiązaniem mniej bezpiecznym – można bowiem przechwycić treść SMS-a. Drugi rodzaj zabezpieczenia jest zdecydowanie bezpieczniejszy. Wymaga zainstalowania aplikacji w sposób autoryzowany przez bank. Aby uzyskać kod, należy zalogować się do tej aplikacji. Kradzież kodu jest trudniejsza niż przechwycenie SMS-a z kodem.

Rozwiązania biometryczne – wykorzystują indywidualne cechy biologiczne (np. odcisk palca, rysy twarzy).

Dzięki zastosowaniu listy haseł jednorazowych, tokena, potwierdzania SMS-em i innych form uwierzytelniania klienta dostęp do konta internetowego przez niepowołane osoby jest niemożliwy nawet w sytuacji, gdy złodziej pozna (np. za pomocą programu szpiegującego) nasz login i hasło do konta. Z drugiej strony, kradzież tokena, telefonu komórkowego czy listy haseł również nie pozwoli na nieupoważnione pobranie pieniędzy z konta internetowego, gdyż złodziej musi jeszcze znać nasz login i hasło.

Poza uwierzytelnianiem klienta niektóre systemy bankowe umożliwiają specyficzne traktowanie wybranych odbiorców przelewów. Podczas zlecania przelewów możemy wskazać odbiorców, dla których transakcje nie będą wymagały weryfikacji, czyli tzw. odbiorców zaufanych. Jest to duże udogodnienie, ale jednocześnie niebezpieczeństwo. Należy bardzo ostrożnie wybierać odbiorców zaufanych – powinny to być osoby, do których mamy pełne zaufanie, np. rodzina, dobrzy znajomi. Raczej nie powinniśmy wybierać jako odbiorców zaufanych instytucji. Niebezpieczeństwo polega na tym, że jeśli ktoś ukradnie nam hasło dostępu do konta, to bez przeszkód będzie mógł dokonać przelewu do zaufanego klienta.

Banki mogą stosować dodatkowe zabezpieczenie w postaci telefonicznego potwierdzenia transakcji. Algorytm wyboru transakcji wymagającej dodatkowego potwierdzenia jest tajemnicą banku. Możemy przypuszczać, że wpływ na wybór mają np.: kwota transakcji, częstotliwość transakcji, historia dotychczasowych transakcji, odbiorca, państwo odbiorcy.

Aby bezpiecznie korzystać z elektronicznego konta bankowego, należy pamiętać o przestrzeganiu kilku ważnych zasad.

Jakie są zasady bezpiecznego korzystania z konta bankowego?

- Nie należy korzystać z konta na komputerach publicznych (znajdujących się w szkole, pracy czy kawiarence internetowej).
- Komputer, z którego korzystamy przy dostępie do konta internetowego, powinien być odpowiednio zabezpieczony. Powinny być na nim zainstalowane: firewall, program antywirusowy oraz aktualizacje bezpieczeństwa dla systemu operacyjnego i przeglądarki internetowej.
- Przy dostępie do konta internetowego należy korzystać wyłącznie z połączeń szyfrowanych, oznaczonych skrótem *https:* na początku adresu URL.
- Należy weryfikować poprawność adresu internetowego banku oraz certyfikatu strony WWW banku.
- Należy pamiętać identyfikatory, hasła, numery PIN – nie wolno ich zapisywać.
- Po zakończeniu korzystania z konta należy pamiętać o wylogowaniu z systemu banku.
- Nie wolno klikać linków umieszczonych w e-mailach zawierających prośby o zalogowanie się do swego konta celem aktualizacji danych, potwierdzenia tożsamości itp.; banki nigdy nie wysyłają takich listów.
- Podejrzane sytuacje należy zgłaszać bankowi, korzystając z infolinii.
- Należy ostrożnie wybierać tzw. zaufanych odbiorców.

Ćwiczenie 4. Sprawdzamy zasady bezpieczeństwa w wybranym e-banku

1. Odszukaj w Internecie strony oferujące bankowość elektroniczną. Podaj przynajmniej trzy adresy tego rodzaju stron.
2. Wejdź na stronę jednego z wyszukanych banków. Sprawdź, które z opisanych wyżej zasad bezpiecznego korzystania z elektronicznego konta bankowego stosuje ten bank. Czy w jego regulaminie opisano także inne od wymienionych wyżej zasad? Jeśli tak, podaj je.

5. E-zakupy



E-zakupy to kupowanie za pośrednictwem Internetu. Możemy w ten sposób kupić to samo, co w sklepach stacjonarnych (nawet artykuły spożywcze), a także dobra wirtualne (np. informacje, muzykę w formacie MP3, filmy, programy). Oczywiście, nie wszystko wygodnie jest kupować przez Internet, np. wciąż trudno przymierzyć w ten sposób garnitur...

Zakupy przez Internet umożliwiają:

- kupowanie bez konieczności wychodzenia z domu, co jest szczególnie ważne w przypadku osób niepełnosprawnych lub mieszkających z dala od dużych miast;
- wybór towarów z o wiele większej oferty niż w tradycyjnym sklepie; nawet w ogromnym magazynie nie da się zgromadzić kilku milionów książek, natomiast może ich tyle być w postaci wirtualnej;
- przed zakupem sprawdzenie dokładnie i w spokoju danych technicznych każdego towaru oraz porównanie ich z innymi produktami; także skontaktowanie się ze sprzedawcą za pomocą poczty elektronicznej lub telefonu i poproszenie o dodatkowe wyjaśnienia;
- porównanie ceny tego samego towaru w różnych sklepach internetowych (za pomocą specjalnego serwisu) i wybranie najkorzystniejszej oferty;
- zapoznanie się z opiniami wcześniejszych nabywców i przez to uniknięcie zakupu „bubla";
- korzystanie z oferty zakupu towarów opracowanej przez wirtualnego sprzedawcę na podstawie historii naszych wcześniejszych zakupów;
- dokonywanie zakupów po cenach niższych niż w tradycyjnych sklepach; sklep internetowy nie potrzebuje wielkich magazynów, powierzchni handlowej, wielu sprzedawców, ochroniarzy itp. Niektóre towary oferowane przez zagraniczne sklepy internetowe (np. elektronika) mogą być tańsze niż w Polsce, nawet jeśli uwzględni się koszty wysyłki. W przypadku kupowania towarów za granicą należy jednak pamiętać o dodatkowych opłatach (cło lub podatki) obowiązujących przy zakupie określonych produktów.

Wątpliwości dotyczące robienia zakupów przez Internet

- nie można obejrzeć ani dotknąć produktu, co utrudnia kupowanie niektórych towarów (np. odzieży, obuwia);
- do ceny towaru często należy doliczyć cenę przesyłki;
- trzeba czekać na dostarczenie towaru (od jednego do kilkunastu dni);
- istnieje niebezpieczeństwo zostania oszukanym.

Jak kupujemy?

- Wchodzimy do sklepu, czyli na odpowiednią stronę WWW.
- Przeglądamy towary (według kategorii, ceny, producenta itp.) bądź wyszukujemy interesujący artykuł, wpisując jego nazwę lub fragmenty opisu.
- Towary, które chcemy kupić, wkładamy do **wirtualnego koszyka** (jeżeli się potem rozmyślimy, możemy z nich zrezygnować).
- Wybieramy formę zapłaty i podajemy adres dostawy.
- Niekiedy zamówienie wymaga potwierdzenia dokonywanego telefonicznie lub przez wysłanie e-maila do sklepu internetowego.

Dostarczaniem przesyłek zajmuje się poczta i firmy kurierskie. Informacja o czasie realizacji zamówienia znajduje się najczęściej na witrynie sklepu; czas ten zależy od solidności sklepu oraz od odległości geograficznej. Na artykuły z zagranicy możemy oczekiwać nawet przez kilka tygodni.



Zgodnie z prawem Unii Europejskiej konsument, który zawarł umowę na odległość (np. kupna towaru w sklepie internetowym na terenie Unii), może od niej odstąpić bez podania przyczyn, składając stosowne oświadczenie na piśmie, w terminie czternastu dni.

Podstawowe sposoby płatności za zakupy dokonane w sklepach internetowych

Przelew – zlecamy w naszym banku dokonanie przelewu na podany przez sklep numer konta; po otrzymaniu przelewu firma wysyła zamówiony towar.

Karta płatnicza – po podaniu numeru karty odpowiednia kwota zostanie pobrana z naszego rachunku; metoda ta jest najwygodniejsza i pozwala na robienie zakupów również za granicą, należy jednak pamiętać o zasadach bezpieczeństwa, w szczególności dbanie o nieudostępnianie osobom trzecim numeru karty oraz trzycyfrowego kodu zabezpieczającego (CVC/CVV).

Poprzez serwis płatniczy – swoje konto w takim serwisie zasilamy w wybrany przez nas sposób (np. przelewem, poprzez obciążenie karty płatniczej), a zgromadzone na koncie środki możemy wykorzystywać przy płaceniu za towary lub przesłać innym osobom.

Przez SMS – po wysyłaniu SMS-a na numer podany na stronie sklepu operator doliczа odpowiednią kwotę do rachunku; tę metodę stosuje się przy płatnościach na niewielkie kwoty (np. kupowanie gier lub dzwonków na telefony komórkowe).

Za pobraniem – płacimy odpowiednią kwotę przy odbiorze towaru, np. listonoszowi lub kurierowi.

Jakich ważnych zasad należy przestrzegać?

- Kupuj w zaufanych sklepach. Przed zakupem zapoznaj się z renomą sklepu (zapytaj znajomych, poszukaj opinii o sklepie na forach dyskusyjnych lub porównaj ceny danego artykułu, korzystając z serwisów umożliwiających porównywanie cen).

- Nie wierz w „wyjątkowe okazje". Znacząco niższa cena towaru może wskazywać, że jest to próba wyłudzenia pieniędzy lub że towar pochodzi z przestępstwa.
- Jeżeli przesyłasz dane, zwłaszcza poufne (w szczególności numer karty płatniczej oraz hasło do konta w sklepie), upewnij się, że korzystasz z szyfrowanego połączenia.
- Dokładnie zapoznaj się z warunkami zakupu, w tym gwarancji. Należy np. pamiętać, że urządzenia elektroniczne w zależności od kraju mogą mieć różne parametry techniczne (pracować z innym napięciem sieciowym), a korzystanie z zakupionych towarów może być niemożliwe lub utrudnione ze względu na uwarunkowania prawne (np. tzw. blokada regionalna dla filmów DVD).

Ćwiczenie 5. Znajdujemy i porównujemy ceny tego samego modelu drukarki 3D

1. Znajdź cenę tego samego modelu drukarki 3D w trzech e-sklepach.
2. Odszukaj adres serwisu do porównywania cen. Porównaj ceny tego samego modelu z wybranych w punkcie 1. e-sklepów.

6. E-aukcje



Tradycyjne aukcje dotyczą sprzedaży dzieł sztuki, okazów dla kolekcjonerów itp. Zasada ich funkcjonowania jest prosta – osoba, która podczas licytacji zaoferuje najwyższą kwotę, nabywa towar. Tę samą zasadę przeniesiono do Internetu. W aukcjach internetowych przebijamy oferty zakupu danego towaru złożone przez innych internautów. Sprzedać i kupić w ramach aukcji można dosłownie wszystko – książki, płyty, zabawki, dzieła sztuki, samochody, domy, statek wycieczkowy. Sprzedającymi są najczęściej osoby indywidualne, czasami także małe firmy. Na aukcjach można również samemu sprzedawać towary.

Aukcje internetowe to bardzo wygodny sposób na pozbycie się wartościowych, ale niepotrzebnych nam rzeczy. Często możemy też znaleźć po okazyjnych cenach interesujące nas przedmioty. Należy jednak pamiętać, że niektóre „okazje" mogą być ryzykowne. Kupując w sklepie internetowym, dokonujemy transakcji z firmą. W przypadku aukcji sprzedający lub kupujący może okazać się niesolidny lub być po prostu oszustem. Serwisy aukcyjne próbują bronić nas przed nieuczciwymi kontrahentami, wprowadzając system rekomendacji. Każdą transakcję można skomentować, wystawiając drugiej stronie pozytywną bądź negatywną opinię. Liczba takich opinii jest pewną miarą rzetelności oferenta.

Jakich ważnych zasad należy przestrzegać?

- Bądź ostrożny w przypadku towarów o podejrzanie niskich cenach. Mogą one np. pochodzić z kradzieży. Kupno takich przedmiotów jest w Polsce przestępstwem. Należy uważać, aby nie stać się ofiarą internetowego oszustwa aukcyjnego.
- Kupuj tylko u osób o dobrej reputacji.

- Sprawdź zawartość paczki przy odbiorze towaru.
- Jeżeli zostaniesz oszukany, natychmiast zawiadom policję. Szybkie działanie pozwala zwykle na ujęcie sprawcy.

Ćwiczenie 6. Szukamy stron internetowych z e-aukcjami

1. Odszukaj w Internecie adresy stron internetowych serwisów aukcyjnych.
2. Sprawdź, czy można kupić na e-aukcji namiot czteroosobowy. Porównaj ceny, korzystając z serwisu umożliwiającego porównywanie cen.

Różnice pomiędzy sklepem a aukcją

Sklep	Aukcja
tylko kupujemy	kupujemy bądź sprzedajemy
ustalona cena	negocjowana cena
duże bezpieczeństwo zakupu	ograniczone bezpieczeństwo zakupu
nowe towary	towary nowe lub używane

Tabela 1. *Różnice pomiędzy sklepem a aukcją*

7. E-podpis

Żeby uwiarygodnić dokument papierowy, autor dokumentu odręcznie się pod nim podpisuje. **E-podpis** (**podpis elektroniczny**) ma spełniać podobną rolę, potwierdzając wiarygodność dokumentu elektronicznego.



E-podpis (podpis elektroniczny) to zbiór danych w postaci elektronicznej, jednoznacznie identyfikujących osobę podpisaną pod dokumentem elektronicznym oraz potwierdzających integralność dokumentu.

Certyfikat C

Dokument (dane) potwierdzające informacje m.in. o: właścicielu dokumentu (danych), okresie ważności dokumentu (danych), sposobach weryfikacji dokumentu (danych), poświadczone przez wystawcę dokumentu.

Aby złożyć podpis elektroniczny, niezbędne jest posiadanie **certyfikatu**. Certyfikaty wydawane są przez centra certyfikacji (organizacje, firmy). Najprostsze certyfikaty (wykorzystywane np. przy podpisywaniu listów elektronicznych) są darmowe, natomiast korzystanie z certyfikatów służących do składania tzw. **kwalifikowanego podpisu elektronicznego** (równoważnego podpisowi tradycyjnemu) kosztuje kilkaset złotych rocznie.

Inną formą e-podpisu jest **podpis zaufany**, dzięki któremu można podpisywać podania i wnioski składane do urzędów. Dzięki takiemu podpisowi możemy bez wychodzenia z domu złożyć wniosek o wyrobienie dowodu osobistego, paszportu czy

o odpis aktu urodzenia. Aby posiadać podpis zaufany, należy założyć **profil zaufany**, za pomocą którego skutecznie potwierdzisz swoją tożsamość w Internecie.

C **Centrum certyfikacji**

Organizacja (firma) zajmująca się wystawianiem certyfikatów.

Profil zaufany możesz założyć przez Internet, jeśli posiadasz konto w banku, który dostarcza taką usługę. Druga możliwość to wypełnienie wniosku i potwierdzenie go w odpowiednim punkcie (najczęściej jest to urząd miasta, urząd skarbowy czy oddział ZUS). Więcej informacji o tym, jakie sprawy można załatwić dzięki profilowi zaufanemu, oraz pełną listę punktów potwierdzających można znaleźć na stronie *pz.gov.pl*.

Od 4 marca 2019 roku wydawane są w Polsce dowody osobiste z warstwą elektroniczną (**e-dowód**). Na e-dowodzie może znaleźć się **podpis osobisty** lub kwalifikowany podpis elektroniczny.

Ćwiczenie 7. Poznajemy usługi wykorzystujące podpis elektroniczny

1. Wejdź na stronę *obywatel.gov.pl*.
2. Zapoznaj się ze sprawami, które musisz załatwić w przypadku wyjazdu za granicę. Które z nich możesz zrealizować, wykorzystując podpis elektroniczny?

Warto zapamiętać

- Rozwój technologii informacyjno-komunikacyjnej zapoczątkował powstawanie różnych elektronicznych usług wspomagających lub zastępujących tradycyjne formy nauczania, pracy, świadczenia usług (e-nauczanie, e-dziennik, e-praca, e-banki, e-zakupy, e-aukcje).
- E-nauka i e-praca polegają na wykonywaniu związanych z nimi obowiązków w dowolnej odległości od instytucji szkoleniowej lub miejsca pracy, z wykorzystaniem sieci komputerowej i Internetu.
- W e-bankach funkcjonuje zazwyczaj kilka metod zabezpieczeń: identyfikator, hasło, token, lista haseł jednorazowych, klucz prywatny itp.
- Korzystając z e-sklepu i e-aukcji, należy być ostrożnym i przestrzegać zasad bezpieczeństwa. Zawsze należy zapoznać się dokładnie z regulaminem korzystania z danej e-usługi.
- Podpis elektroniczny umożliwia potwierdzanie autentyczności cyfrowego dokumentu.

Pytania i polecenia

1. Jak oceniasz przyszłość pracy i nauki na odległość?
2. Wyjaśnij, dlaczego niektórzy podchodzą sceptycznie do e-pracy i e-nauki.
3. Podaj pięć zalet korzystania z dziennika elektronicznego.
4. Omów zalety telepracy. Czy uważasz, że są wystarczające, aby przysłonić jej wady?
5. O jakich zasadach trzeba pamiętać, dokonując zakupów w sklepie internetowym?
6. Dlaczego niektórzy ludzie obawiają się lokowania swoich pieniędzy na e-kontach?
7. Omów metody zabezpieczeń pieniędzy w bankach elektronicznych.
8. Na czym polega aukcja internetowa?
9. Jakich zasad należy przestrzegać, uczestnicząc w aukcji internetowej?

10. Z jakiej e-usługi najczęściej korzystasz? Podaj zalety korzystania z elektronicznej formy tej usługi.
11. Co to jest podpis elektroniczny?

Zadania

1. Przedstaw w postaci tabeli zalety i wady nauki oraz pracy na odległość. Wyszukaj dodatkowe informacje w Internecie.
2. Wejdź na stronę wybranego sklepu internetowego. Włóż kilka interesujących cię rzeczy do koszyka.
3. Sprawdź, gdzie najtaniej możesz kupić łyżwy. Skorzystaj z porównywarki cen.
4. Sprawdź, jak można w Internecie kupić… helikopter.
5. Sprawdź w Internecie, czy prawdziwe jest zdanie: *Ponad 15 milionów osób korzysta z serwisu Allegro*.

Dla zainteresowanych

6. Przygotuj w edytorze tekstu opracowanie na temat zdalnego nauczania. Zwróć szczególną uwagę na korzyści wynikające z tej formy nauczania dla osób niepełnosprawnych.
7. Chcesz założyć własny bank internetowy. Jakie formy działania i bezpieczeństwa zaproponowałbyś klientom?
8. W podpisie elektronicznym wykorzystuje się skomplikowane sposoby szyfrowania, oparte na tzw. kluczu prywatnym i publicznym. Znajdź więcej informacji na ten temat.

Przeczytaj, jeśli chcesz wiedzieć więcej…

W 2009 roku powstała pierwsza cyfrowa waluta: **Bitcoin**. Generowane z wykorzystaniem algorytmów kryptograficznych wirtualne pieniądze są przesyłane bezpośrednio między użytkownikami – podobnie jak pliki w sieciach P2P. Internetowe giełdy pozwalają zamienić prawdziwe pieniądze (w tym złotówki) na Bitcoiny i odwrotnie. Bitcoiny można generować także samemu, wymaga to jednak posiadania odpowiednio wydajnego sprzętu.

Bitcoin to pierwsza z tzw. **kryptowalut**. Obecnie na świecie funkcjonuje kilkadziesiąt innych kryptowalut.



Rys. 3. *Przykładowe (fikcyjne) mLegitymacje pokazane na stronie Ministerstwa Cyfryzacji*

Od września 2018 r. funkcjonują szkolne eLegitymacje (legitymacje elektroniczne) oraz mLegitymacje (legitymacje mobilne). Legitymację mobilną przygotowało Ministerstwo Cyfryzacji i jest ona częścią aplikacji mObywatel, którą instaluje się na telefonie komórkowym. W ten sposób uczeń dysponujący telefonem ma legitymację szkolną zawsze ze sobą. Legitymacja mobilna zawiera te same dane co tradycyjna i można jej używać w tych samych sytuacjach co tradycyjnej, np. do potwierdzenia zniżki przy kupnie biletu na pociąg czy do kina. Więcej na ten temat mLegitymacji – na stronie internetowej Ministerstwa Cyfryzacji: *https://www.gov.pl/web/cyfryzacja*.

Temat D3. Zadania projektowe

1. Przygotowanie projektu
   1.1. Etapy tworzenia projektu grupowego
   1.2. Debata ZA i PRZECIW
2. Zadania projektowe
   Zadanie 1. Rozwój systemów operacyjnych
   Zadanie 2. Korzystanie z wybranych e-usług

Warto powtórzyć

1. Jakie znasz możliwości opracowywania dokumentu tekstowego o rozbudowanej strukturze?
2. Jakie znasz zasady przygotowania prezentacji multimedialnej?

1. Przygotowanie projektu

Podczas pracy nad projektami wykorzystamy poznane dotychczas zagadnienia, m.in.: tworzenie prezentacji multimedialnej, edycję rozbudowanych dokumentów tekstowych, wyszukiwanie informacji w Internecie, pracę w edytorze grafiki.

Przed przystąpieniem do wykonania zadań szczegółowych należy zgromadzić odpowiednie materiały. Można skorzystać z fachowej literatury, Internetu, w tym z zasobów na portalach edukacyjnych.



Jeżeli wykorzystujemy cudze materiały, powinniśmy pamiętać o przestrzeganiu przepisów dotyczących korzystania z takich materiałów. Pamiętaj, że bez ograniczeń można korzystać z grafiki i innych utworów w tzw. domenie publicznej, ponieważ autorskie prawa majątkowe do tych utworów wygasły lub autorzy zrzekli się ich.

Autorskie prawa majątkowe wygasają po siedemdziesięciu latach od śmierci twórcy, a w przypadku utworu nieznanego autora – po siedemdziesięciu latach od daty pierwszego rozpowszechnienia (publicznego udostępnienia). W Wikipedii zbiory znajdujące się w domenie publicznej oznaczono symbolem: .

1.1. Etapy tworzenia projektu grupowego

1. Określenie zadania.
2. Analiza zadania i ustalenie zadań szczegółowych.
3. Określenie formy dokumentów oraz dobór nazw plików.
4. Wybranie odpowiednich narzędzi – programów komputerowych, np. wyszukiwarki internetowej, edytora tekstu i grafiki, programu do tworzenia prezentacji multimedialnej.

5. Ustalenie sposobu przechowywania dokumentów oraz założeń dotyczących łączenia dokumentów.
6. Przydzielenie zadań poszczególnym osobom (grupom).
7. Wykonanie zadań szczegółowych.
8. Połączenie dokumentów.
9. Sprawdzenie, czy zadania wykonano według wcześniejszych założeń.
10. Prezentacja projektu.

Nad projektem grupowym pracuje zespół uczniów. Dlatego należy wyznaczyć koordynatorów grup, którzy przydzielą zadania szczegółowe (w tym odpowiednie czynności) każdemu z członków grupy oraz będą czuwali nad przebiegiem realizacji projektu (nawet w przypadku grup dwuosobowych).

W każdej grupie („pod okiem" koordynatora) ustalamy sposób przechowywania i udostępniania dokumentów komputerowych (również z wykorzystaniem chmury).

Ważne jest wcześniejsze ustalenie nazw plików. Jeśli nie chcemy, aby pliki zostały nadpisane (w przypadku kopiowania plików do tego samego folderu), należy dopilnować, aby każdy został inaczej nazwany. Jeżeli komputery pracują w sieci lokalnej, należy korzystać z możliwości przesyłania plików poprzez sieć w celu skopiowania ich czy przeniesienia na inny komputer. W kolejnych punktach tematu przedstawiono zadania projektowe z krótkim opisem, przykładowymi zadaniami szczegółowymi i uwagami dotyczącymi ich wykonania.

Po otrzymaniu zadań szczegółowych gromadzimy odpowiednie materiały. Aby zebrać informacje, korzystamy z różnych źródeł, m.in. encyklopedii, w tym encyklopedii multimedialnych i internetowych, słowników (także internetowych), podręczników szkolnych, czasopism, prowadzimy wywiady.

Po zgromadzeniu materiałów porządkujemy je i wybieramy najistotniejsze. Należy pamiętać, że nie wszystkie informacje pochodzące z Internetu są rzetelne. Najlepiej korzystać z poleconych przez nauczyciela portali edukacyjnych i stron zaufanych wydawców.

Zadania projektowe mogą być wykonywane w postaci prezentacji multimedialnych (od pięciu do dziesięciu slajdów). Natomiast poszczególne zadania szczegółowe – w postaci dokumentów tekstowych, obrazów, filmów, odręcznych szkiców na kartkach.

Po wykonaniu zadań projektowych każda grupa powinna mieć możliwość zaprezentowania swojego projektu. W przypadku prezentacji multimedialnych wspomagających wystąpienie prelegenta – z wykorzystaniem projektora i ekranu. Można również przygotować prezentację do samodzielnego oglądania. Wybrane projekty można prezentować, przeprowadzając **debatę ZA i PRZECIW**.

W temacie omawiamy dwa przykładowe zadania projektowe. Inne zadania projektowe można wykonać w podobny sposób.

1.2. Debata ZA i PRZECIW

Debatę za i przeciw przeprowadza się w dwóch grupach. Każda grupa wybiera rzecznika i sekundanta. Pozostała część klasy jest publicznością.

Jedna osoba (uczeń lub nauczyciel) kieruje całą dyskusją.

Przebieg debaty:

- Rzecznik grupy ZA przedstawia tezę, popierając ją argumentami.
- Rzecznik grypy PRZECIW stara się obalić tezę.
- Sekundant grupy ZA podtrzymuje tezę.
- Sekundant grupy PRZECIW popiera argumentami wypowiedzi rzecznika grupy PRZECIW.
- Do debaty włącza się publiczność. Każda osoba może zabrać głos tylko jeden raz i powinna ustosunkować się do argumentacji ZA lub PRZECIW. Osoba prowadząca debatę pilnuje czasu wystąpień i zamyka debatę.
- Rzecznicy obu grup wygłaszają końcowe komunikaty.
- Osoba prowadząca debatę przeprowadza głosowanie.

Tezy i argumenty poszczególnych osób powinny być zapisywane na bieżąco (z wykorzystaniem edytora tekstu) i wyświetlane na ekranie, aby wszyscy uczestnicy je widzieli i nie powtarzali tych samych wypowiedzi. Najlepiej umieszczać je w tabeli (przykłady pokazano w tabeli 1.).

2. Zadania projektowe

Zadanie 1. Rozwój systemów operacyjnych

Celem projektu jest przedstawienie najważniejszych elementów procesu rozwoju systemów operacyjnych.

Punktem wyjścia powinna być krótka historia systemów operacyjnych (w tym systemu Windows) przedstawiona w temacie A2 – chronologiczne ujęcie ważniejszych wydarzeń. W projekcie należy pokazać, jakie funkcje systemu pojawiały się w kolejnych wersjach oraz czym charakteryzują się poszczególne systemy. Należy również poszukać dodatkowych informacji w Internecie.

Projekt przygotujemy w postaci prezentacji multimedialnej zgodnie z zasadami omówionymi w temacie B4, m.in. wykorzystując gotowe szablony i grafikę SmartArt oraz ikony.

Zadania szczegółowe

1. Zebranie materiałów dotyczących historii powstania systemów operacyjnych i opracowanie tych materiałów w postaci dokumentu tekstowego.
2. Przygotowanie przykładów pokazujących najważniejsze funkcje wybranych systemów operacyjnych.
3. Przygotowanie materiałów dotyczących wybranych systemów operacyjnych, m.in. zrzutów pulpitów.
4. Wyszukanie i opracowanie ciekawostek dotyczących powstania lub funkcjonowania danego systemu operacyjnego.

Uwagi:

- Klasę dzielimy na maksymalnie dwuosobowe grupy. Każda grupa przygotowuje materiały dotyczące jednego systemu operacyjnego.
- Tekst do umieszczenia na slajdach prezentacji można przygotować w dokumencie tekstowym i sformatować tak, by można go było zaimportować do prezentacji multimedialnej (temat B4).
- Podczas pracy nad dokumentami tekstowymi warto korzystać z trybu recenzji (temat B2).
- Koordynator czuwa nad przestrzeganiem zasad prawidłowego przygotowania prezentacji multimedialnej.

Zadanie 2. Korzystanie z wybranych e-usług

Celem projektu jest określenie wpływu informatyki i technologii komputerowej na najważniejsze sfery życia osobistego i zawodowego w przypadku korzystania z wybranych e-usług. W projekcie należy pokazać, w jaki sposób rozwój e-usług wpłynął na kształtowanie się obrazu społecznego i gospodarczego kraju i jakie dał szanse społeczeństwu. Należy zwrócić szczególną uwagę na wpływ rozwoju różnych form kształcenia na odległość i e-pracy na naukę i na rynek pracy.

Rozwój informatyki i technologii informacyjno-komunikacyjnych zmienił też w wielu przypadkach standard życia osób niepełnosprawnych, m.in. umożliwił zdobycie pracy, a przede wszystkim ułatwił wykonywanie codziennych czynności i kontakt z innymi ludźmi.

Projekt zaprezentujemy w postaci debaty ZA i PRZECIW i przygotujemy go na podstawie tematu D2 i dodatkowych informacji znalezionych w Internecie.

Klasę dzielimy na grupy (przynajmniej czteroosobowe). Koordynator projektu przydziela grupom wybrane usługi. W każdej grupie muszą znaleźć się zwolennicy i przeciwnicy danej e-usługi (przeciwnicy popierają usługi tradycyjne). W każdej grupie wybieramy też rzecznika i sekundanta. Podczas poszczególnych debat pozostała część klasy będzie publicznością.

Debata powinna być wcześniej wstępnie przygotowana. Jeśli w trakcie jej trwania zostaną postawione nowe tezy, strona przeciwna powinna umieć znaleźć kontrargumenty. Grupy ZA i PRZECIW nie powinny się ze sobą kontaktować i przekazywać sobie tez i argumentów przed debatą. Należy jednak przygotować główne zagadnienia, według których ma się toczyć debata (tabela 1.).

Przykładowe zadania szczegółowe do debaty, np. na temat e-banków

1. Przygotowanie prezentacji danej e-usługi (np. e-banku).
2. Przygotowanie głównych zagadnień, wokół których ma toczyć się debata.
3. Wstępne przygotowanie tezy ZA i argumentów ZA daną e-usługą.
4. Wstępne przygotowanie tezy PRZECIW i argumentów PRZECIW danej e-usłudze, czyli za usługą w wersji tradycyjnej.
5. Wstępne przygotowanie końcowych komunikatów.

Uwaga: Przedstawienie danej e-usługi można wykonać w postaci krótkiej prezentacji multimedialnej (do trzech slajdów). Można też pokazać przykładową stronę internetową oferującą daną e-usługę.

	ZA, czyli e-banki	**PRZECIW, czyli banki tradycyjne**
kontakty międzyludzkie	ułatwienie kontaktów osobom niepełnosprawnym i niemogącym wyjść z domu	zanik bezpośrednich kontaktów
zakres usług		
bezpieczeństwo		
ograniczenia		
komfort życia		

Tabela 1. *Przykładowa tabela do zadania 2. – e-banki*



Przykłady innych zadań projektowych

1. Elementy komputera, urządzenia pamięci masowej i peryferyjne.
2. Rozwój języków programowania.
3. Rozwój oprogramowania użytkowego.
4. Zasady opracowywania dokumentów tekstowych o rozbudowanej strukturze.
5. Opracowywanie grafiki rastrowej na przykładzie pracy w edytorze GIMP.
6. Zasady przygotowywania prezentacji multimedialnych.
7. Programowanie w języku Python.
8. Programowanie w języku C++.
9. Historia Internetu i piramida rozwoju usług internetowych.

Uwaga: Jeśli klasa uczyła się obydwu języków programowania, można przygotować debatę na temat języków programowania. W grupie ZA znajdą się np. zwolennicy programowania w języku C++, a w grupie PRZECIW – zwolennicy programowania w języku Python.

Dla zainteresowanych

10. Wybierz (wybierzcie) własny problem do opracowania i wykonaj (wykonajcie) projekt według schematu podanego w temacie.

Słownik pojęć

A

Adres IP
Unikatowy numer przyporządkowany urządzeniom sieci komputerowych (pracującym zarówno w Internecie, jak i w sieciach lokalnych) funkcjonujących na podstawie protokołu IP (str. 170).

Akapit
W informatyce to fragment tekstu pisany bez naciśnięcia klawisza **Enter**. Naciskając klawisz **Enter**, kończymy jeden akapit i zaczynamy następny (str. 38).

Algorytm
Uporządkowany i uściślony sposób rozwiązywania danego problemu, zawierający szczegółowy opis wykonywanych czynności w skończonej liczbie kroków (str. 100).

B

BIOS
(z ang. *Basic Input Output System*) Podstawowy system wejścia/wyjścia. Oprogramowanie pośredniczące pomiędzy sprzętem a systemem operacyjnym komputera (str. 11).

C

Centrum certyfikacji
Organizacja (firma) zajmująca się wystawianiem certyfikatów (str. 195).

Certyfikat
Dokument (dane) potwierdzające informacje m.in. o: właścicielu dokumentu (danych), okresie ważności dokumentu (danych), sposobach weryfikacji dokumentu (danych), poświadczone przez wystawcę dokumentu (str. 194).

F

Format pliku
Reguły określające wewnętrzną budowę pliku (str. 65).

Formatowanie dysku
Przygotowanie dysku do współpracy z wybranym systemem plików. Formatowanie dysku usuwa zapisane na nim informacje (str. 16).

H

Hipertekst
Sposób organizacji danych (w szczególności tekstowych) polegający na połączeniu niezależnych fragmentów tekstu hiperłączami (str. 170).

Hiperłącze
(ang. *hyperlink* – inaczej: odsyłacz, łącze, link) Umieszczone w dokumencie komputerowym odwołanie do innego dokumentu (hipertekstowego, tekstowego, graficznego, animacji, wideo) lub innego miejsca w tym samym dokumencie (str. 170).

I

Implementacja
Zapisanie algorytmu w postaci kodu źródłowego (str. 107).

Internet
Rozległa sieć komputerowa łącząca inne sieci z całego świata. Jest wszechstronnym systemem informacyjnym i komunikacyjnym. Udostępnia informacje ze wszystkich dziedzin życia, zgromadzone na komputerach całego globu oraz umożliwia komunikację między ludźmi nawet z najbardziej odległych miejsc (str. 169).

Interpretacja
Tłumaczenie programu utworzonego w języku programowania instrukcja po instrukcji, tak aby komputer mógł wykonać każdą z nich. Tłumaczenie następuje każdorazowo przy uruchomieniu programu (str. 107).

J

Język programowania
Specjalny język służący do pisania programów komputerowych. Jest on zbiorem określonych instrukcji i zasad składni (str. 101).

K

Karta rozszerzenia
Element składowy jednostki systemowej komputera. Karta rozszerzenia umożliwia komputerowi realizowanie dodatkowych funkcji (np. generowanie realistycznej grafiki komputerowej, odbieranie sygnału telewizyjnego). Ma postać płytki drukowanej ze złączem, za którego pomocą jest montowana w odpowiednim slocie na płycie głównej, oraz tzw. śledziem, na którym znajdują się gniazda służące do podłączenia urządzeń zewnętrznych (np. monitora) (str. 11).

Kod maszynowy
(program maszynowy, kod wynikowy, program wynikowy) Program napisany w języku wewnętrznym (maszynowym), wykonywany przez procesor (str. 107).

Kompilacja
Przetłumaczenie programu w całości na język zrozumiały dla procesora, tak by mógł go wykonać komputer. Jeśli kompilacja przebiegnie poprawnie, można uruchomić program. Raz skompilowany program nie wymaga już powtórnej operacji tłumaczenia (str. 107).

Komórka pamięci
Część pamięci operacyjnej komputera (pamięci RAM). Pamięć operacyjna jest podzielona logicznie na komórki pamięci. Każda komórka pamięci ma swój unikatowy adres. W komórkach pamięci mogą być przechowywane dane reprezentujące instrukcje procesora lub dane dla tych instrukcji (str. 116).

L

Lista kroków
Przedstawienie algorytmu w kolejnych punktach (krokach). Każdy punkt takiej listy zawiera opis wykonywanej czynności. Kolejność punktów nie jest przypadkowa – musi być zgodna z działaniem algorytmu (str. 102).

M

Moduł
Plik zawierający kod przeznaczony do wykorzystania w innych programach (str. 149).

N

Nagłówek i stopka
Odpowiednio górna i dolna część strony dokumentu, w której można umieścić tekst lub inne obiekty (np. grafikę) (str. 36).

Napęd optyczny
Urządzenie, które za pomocą wiązki lasera odczytuje dane lub zapisuje je na tzw. nośnikach optycznych (płyty CD, DVD, Blu-ray) (str. 18).

Nazwa pliku
Składa się z dwóch członów: nazwy właściwej i rozszerzenia (str. 33).

O

Odwołania
Umożliwiają przechodzenie do konkretnych fragmentów tekstu i obiektów w ramach jednego dokumentu (str. 43).

P

Partycja dyskowa
(dysk logiczny) Wydzielony z fizycznego obszaru dysku spójny fragment, z punktu widzenia logiki komputera traktowany jako osobne urządzenie (dysk) (str. 16).

Piksel
Najmniejszy element obrazu wyświetlanego na ekranie monitora (str. 17).

Prezentacja multimedialna
Przedstawienie danego zagadnienia w postaci elektronicznego dokumentu, zawierającego zwykle: tekst, obrazy, wykresy, schematy, dźwięki, filmy (str. 81).

Program komputerowy
(program źródłowy, kod źródłowy) Ciąg instrukcji języka programowania, realizujący algorytm (str. 100).

Preprocesor
Program, który przetwarza wstępnie kod źródłowy przed rozpoczęciem zasadniczej kompilacji. Polecenia preprocesora (zwane dyrektywami) rozpoczynają się od znaku `#`. Np. dyrektywa `#include` nakazuje dołączenie do kodu źródłowego kodu z innych plików źródłowych (str. 112).

Proporcja obrazu
Stosunek szerokości obrazu do jego wysokości (str. 17).

R

RAM
(z ang. *Random Access Memory*)
Pamięć o dostępie swobodnym do chwilowego przechowywania danych i kodów działających programów (str. 11).

ROM
(z ang. *Read Only Memory*)
Pamięć tylko do odczytu (str. 13)

Rozdzielczość
Określa liczbę pikseli obrazu wyświetlanych na ekranie monitora. Rozdzielczość wyraża się w postaci liczby pikseli w poziomie i w pionie (str. 17).

Rozszerzenie pliku
Określa typ pliku i zwykle wskazuje program, w którym utworzono plik (str. 33).

S

Schemat blokowy
Przedstawienie poszczególnych operacji algorytmu za pomocą odpowiednio połączonych figur (bloków). Połączenia określają, w jakiej kolejności i w jaki sposób będą wykonywane operacje realizujące dany algorytm (str. 102).

Sekcja
Część dokumentu, wyodrębniona np. w celu zastosowania innych od użytych w pozostałych częściach tekstu parametrów formatowania (str. 51).

Serwer HTTP
Program działający na serwerze internetowym, udostępniający informacje poprzez protokół HTTP (str. 171).

Słowo kluczowe
Słowo mające szczególne znaczenie w danym języku programowania, np. oznaczające określony rozkaz, instrukcję lub polecenie (str. 106).

Specyfikacja zadania
(problemu) Szczegółowy opis zadania, w którym określa się dane wejściowe i oczekiwane wyniki oraz związek między danymi a wynikami (warunki, jakie muszą spełniać wyniki) (str. 103).

Styl tekstu
Zbiór parametrów formatowania tekstu, takich jak: krój i rozmiar czcionki, odstęp między wierszami (interlinia), położenie tabulatorów i wielkość wcięć, sposób obramowania, sposób numerowania itp. (str. 38).

Szablon
Wzorzec podstawowej struktury dokumentu. Może określać m.in.: style, klawisze skrótów przypisane poleceniom, makra, czcionki, ustawienia marginesów, układ strony (str. 52).

T

Token
(generator kodów jednorazowych) Urządzenie elektroniczne, służące do uwierzytelniania transakcji internetowych, najczęściej bankowych (str. 188).

Typ zmiennej
Określa rodzaj danych przechowywanych w zmiennej – mogą to być np. liczby, znaki, wartości logiczne (typy proste) lub teksty, tablice, rekordy i obiekty (typy złożone) (str. 116).

U

UEFI
(z ang. *Unified Extensible Firmware Interface*) Oprogramowanie pośredniczące pomiędzy oprogramowaniem sprzętowym (ang. firmware) a systemem operacyjnym komputera, następca BIOS-u (str. 11).

URL
(z ang. *Uniform Resource Locator* – uniwersalny lokalizator zasobów) Standard tworzenia nazw zasobów Internetu, jednoznacznie określający adres zasobu i sposób dostępu do niego (str. 171).

W

Wielodostępowość
Cecha systemu operacyjnego, pozwalająca na korzystanie z tego samego komputera różnym użytkownikom, mającym swoje konta i własne ustawienia (str. 29).

Wielozadaniowość
Możliwość równoczesnego uruchamiania wielu aplikacji (str. 27).

Z

Zintegrowane środowisko programistyczne
(IDE – z ang. *Integrated Development Environment*) Aplikacja lub zespół aplikacji do tworzenia, modyfikowania, testowania i utrzymywania oprogramowania (str. 111).

Podręcznik jest przeznaczony do nauczania informatyki w szkołach ponadpodstawowych na poziomie podstawowym.

Podręcznik składa się z czterech modułów, a każdy z nich z tematów o wyrazistej strukturze merytorycznej i metodycznej. Modułowa budowa podręcznika pozwala realizować tematy w podanym układzie, ale również dostosować kolejność ich realizacji do warunków szkoły i grupy uczniów.

Prezentowane w podręczniku metody, pojęcia i przykłady uwzględniają różnorodność oprogramowania, jaka występuje w szkołach.

Uczeń poznaje zagadnienia w sposób systematyczny i uporządkowany, a rozwiązując niebanalne zadania, rozwija wyobraźnię twórczą i uczy się logicznego myślenia.

Podręcznik zawiera:

- ponad 260 ćwiczeń i zadań sprawdzających, z tego ponad 100 z programowania
- ponad 140 pytań sprawdzających
- 12 zadań projektowych

