Grażyna Koba

TERAZBAJTY

Informatyka

dla szkół ponadpodstawowych

Zakres podstawowy

Klasa II

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
Michał Koba
Marta Skała-Kowalczyk
Michał Łętowski
Piotr Babij

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: 2020.

Numer ewidencyjny w wykazie: 1042/2/2020

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

ISBN 978-83-964794-7-1

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

Wydanie I

Fotografie: Michał Koba, Marcin Mytych

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 i sieci komputerowych 7

Temat A1. Sieci komputerowe 8

1. Dlaczego komputery łączy się w sieć? 8
2. Podział sieci 10
3. Praktyczne sposoby tworzenia sieci komputerowej 12
4. Podstawy konfiguracji sieci 13
   - 4.1. Protokoły sieciowe 14
   - 4.2. Identyfikacja sieciowa 16
5. Co jest niezbędne do budowy sieci? 17
   - 5.1. Sieć domowa 17
   - 5.2. Szkolna sieć komputerowa 19
6. Podstawy pracy w sieci 20
   - 6.1. Logowanie 20
   - 6.2. Udostępnianie zasobów 21
   - 6.3. Mapowanie zasobów 22

Temat A2. Bezpieczeństwo i ochrona danych w komputerach i sieciach komputerowych 26

1. Wprowadzenie 26
2. Ochrona danych przed nieupoważnionym dostępem 27
   - 2.1. Kontrola dostępu do danych 27
   - 2.2. Nieupoważniony dostęp do danych w wyniku nieświadomych działań użytkownika 29
   - 2.3. Nieupoważniony dostęp do danych spowodowany działaniem innych osób 30
   - 2.4. Odmiany złośliwego oprogramowania 32
   - 2.5. Oprogramowanie zabezpieczające komputer 33
   - 2.6. Szyfrowanie danych 34
3. Ochrona przed utratą danych 36
   - 3.1. Odzyskiwanie przypadkowo usuniętych danych 36
   - 3.2. Odzyskiwanie danych w przypadku awarii komputera lub systemu operacyjnego 36
   - 3.3. Odzyskiwanie danych w przypadku ich utraty spowodowanej czynnikami zewnętrznymi 39

Moduł B Wokół dokumentów komputerowych 41

Temat B1. Formuły, funkcje i wykresy w arkuszu kalkulacyjnym 42
1. Tworzenie formuł w arkuszu kalkulacyjnym 42
2. Formaty danych 43
3. Formatowanie tabeli arkusza kalkulacyjnego 45
4. Stosowanie funkcji arkusza kalkulacyjnego 46
   4.1. Wstawianie funkcji 46
   4.2. Funkcja logiczna JEŻELI 47
   4.3. Funkcja statystyczna LICZ.JEŻELI 48
   4.4. Funkcje matematyczne 49
   4.5. Funkcje daty i czasu 51
5. Przedstawianie danych w postaci wykresu 52
   5.1. Najczęściej używane typy wykresów 52
   5.2. Wykresy funkcji trygonometrycznych i liniowych 54
6. Przykłady rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego 55

Temat B2. Filtry oraz tabele i wykresy przestawne w arkuszu kalkulacyjnym 72
1. Korzystanie z filtrów 72
2. Tabele i wykresy przestawne 74
3. Stosowanie filtru w tabeli przestawnej 77
4. Praktyczny poradnik 79

Temat B3. Opracowywanie grafiki wektorowej 84
1. Stosowanie narzędzi edytora tekstu do tworzenia rysunków w grafice wektorowej 84
   1.1. Korzystanie z kształtów 84
   1.2. Edytowanie punktów w kształtach 87
2. Przykładowy edytor grafiki wektorowej – program Inkscape 88
3. Tworzenie obrazu 89
   3.1. Rysowanie figur 89
   3.2. Rysowanie ścieżek 93
   3.3. Rysowanie przestrzenne 95
4. Przekształcenia 96
5. Warstwy 97

Moduł C Wokół algorytmiki i programowania 101

Temat C1. Więcej o stosowaniu instrukcji iteracyjnych w językach C++ i Python 102
1. Stosowanie instrukcji iteracyjnej `for` w językach C++ i Python – powtórzenie ... 102
2. Stosowanie instrukcji `while` w językach C++ i Python 104
3. Stosowanie instrukcji `do ... while` w języku C++ 106

Temat C2. Stosowanie funkcji w językach C++ i Python 110
1. Funkcje w językach C++ i Python 110
2. Funkcje zwracające wartość w językach C++ i Python 111
3. Funkcje niezwracające wartości w językach C++ i Python 117

Temat C3. Stosowanie tablic w języku C++ i list w języku Python 122
1. Zmienne indeksowane 122
2. Deklarowanie tablicy w języku C++ 123
3. Definiowanie listy w języku Python 124

4. Wprowadzanie elementów do tablicy i wyprowadzanie elementów na ekran 125
5. Wykonywanie operacji na elementach tablicy lub listy 127

Temat C4. Wybrane algorytmy na liczbach naturalnych i algorytmy porządkowania 130
1. Algorytm badania pierwszości liczby 130
2. Obliczanie wartości elementów ciągu Fibonacciego 132
3. Algorytmy porządkowania 135
   3.1. Porządkowanie metodą bąbelkową 135
   3.2. Porządkowanie przez wstawianie 138

Moduł D Wokół Internetu i projektów 143

Temat D1. Wybrane przepisy prawa dotyczące technologii informacyjno-komunikacyjnych 144
1. Prawo autorskie 144
2. Korzystanie z cudzych utworów 145
3. Ochrona wizerunku 148
4. Zasady korzystania z programów komputerowych 149
5. Przestępczość komputerowa 150

Temat D2. Komunikacja i wymiana informacji w Internecie 154
1. Wybrane formy komunikacji i wymiany informacji 154
2. Działanie poczty elektronicznej 156
3. Zasady netykiety 157
4. Chmura obliczeniowa 159
5. Korzyści i zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnych 161

Temat D3. Tworzenie stron WWW z wykorzystaniem stylów i elementów programowania 166
1. Jak zapisana jest strona WWW? 166
2. Przykład tworzenia strony w języku HTML 169
   2.1. Stosowanie podstawowych znaczników języka HTML 169
   2.2. Tworzenie łączy hipertekstowych 173
   2.3. Tworzenie podstrony 174
   2.4. Wstawianie obrazów 174
   2.5. Wstawianie tabel 176
   2.6. Stosowanie kolorów 177
3. Kaskadowe arkusze stylów CSS 177
   3.1. Najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości 179
   3.2. Formatowanie hiperłączy 180
   3.3. Tło strony 180
   3.4. Klasy elementów 181
4. Elementy dynamiczne na stronie – skrypty w języku JavaScript 181
5. Publikowanie i promowanie stron internetowych w Internecie 183
   5.1. Publikowanie strony WWW 183
   5.2. Promowanie strony WWW 185

Temat D4. Zadania projektowe 187
1. Przygotowanie projektu 187
2. Zadania projektowe 188

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

Słownik – wyjaśnienie ważnych terminów

Czy wiesz, że:

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 i sieci komputerowych

A

Moduł

Temat A1. Sieci komputerowe

Temat A2. Bezpieczeństwo i ochrona danych w komputerach i sieciach komputerowych

Temat A1. Sieci komputerowe

1. Dlaczego komputery łączy się w sieć?
2. Podział sieci
3. Praktyczne sposoby tworzenia sieci komputerowej
4. Podstawy konfiguracji sieci
   - 4.1. Protokoły sieciowe
   - 4.2. Identyfikacja sieciowa
5. Co jest niezbędne do budowy sieci?
   - 5.1. Sieć domowa
   - 5.2. Szkolna sieć komputerowa
6. Podstawy pracy w sieci
   - 6.1. Logowanie
   - 6.2. Udostępnianie zasobów
   - 6.3. Mapowanie zasobów

Warto powtórzyć

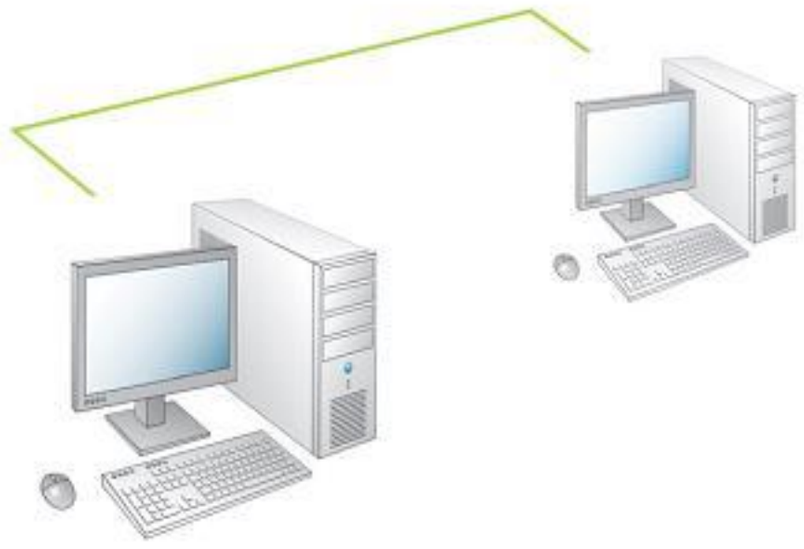
1. W jaki sposób była zorganizowana pracownia komputerowa w twojej szkole podstawowej?
2. Ile komputerów potrzeba, aby utworzyć sieć komputerową?
3. Jaką siecią jest Internet?
4. Co oznacza skrót TCP/IP?
5. Jakie znasz rodzaje nośników pamięci masowej?
6. W jaki sposób można wykonać kopie zapasowe plików?

1. Dlaczego komputery łączy się w sieć?

Sieć komputerowa łączy ze sobą komputery w celu wymiany danych. Najmniejsza sieć komputerowa składa się z dwóch komputerów. Największą siecią jest Internet, łączący setki milionów komputerów i innych urządzeń elektronicznych na całym świecie.

Sieć komputerowa umożliwia udostępnianie danych innym użytkownikom sieci – zarówno tym, którzy znajdują się w sąsiednim pomieszczeniu, jak i w innym mieście czy na innym kontynencie. Połączenie komputerów w sieć pozwala na przesyłanie plików, wspólne korzystanie z baz danych, gier komputerowych, Internetu.

Użytkownicy sieci uzyskują dostęp do dużych przestrzeni dyskowych, programów i urządzeń zewnętrznych, co pozwala obniżyć koszty organizacji pracy. Jedna drukarka może służyć kilku połączonym siecią użytkownikom. Posiadacze komputerów przenośnych mogą za pomocą połączeń bezprzewodowych korzystać z **zasobów** sieci.



***Rys. 1.** Najmniejsza sieć komputerowa*

Zasoby sieciowe to wszelkiego rodzaju dane i urządzenia, do których uzyskujemy dostęp w obrębie sieci. Zasobem sieciowym jest zarówno folder na komputerze kolegi, serwer plików w pracowni szkolnej, zainstalowany program, napęd CD/DVD, jak i drukarka przeznaczona do wspólnego użytkowania.

Szybka i bezpośrednia wymiana danych	Odbywa się bez pośrednictwa dodatkowych nośników, co pozwala na znaczną oszczędność czasu; nie ogranicza wielkości plików; zapobiega zagubieniu danych oraz błędom kopiowania.
Współdzielenie zasobów sprzętowych	Umożliwia korzystanie z jednej drukarki przez użytkowników z kilku stanowisk komputerowych.
Centralne składowanie programów i danych	Polega na dostępie do wspólnej bazy danych i korzystaniu ze wspólnych aplikacji przez użytkowników z wielu stanowisk komputerowych. Dzięki temu można uniknąć np. dwukrotnej sprzedaży tego samego towaru czy rezerwacji tych samych biletów lotniczych.
Automatyzacja wykonywania kopii zapasowych ważnych plików	Umożliwia kontrolę nad danymi z jednego miejsca, co pozwala na skuteczne i szybkie archiwizowanie i tworzenie kopii zapasowych. Awarie sprzętu nie rzutują wówczas na ciągłość pracy np. przedsiębiorstwa.
Zdalny dostęp	Umożliwia korzystanie z zasobów sieciowych przez dostęp z dowolnego miejsca na świecie.
Łatwa rozbudowa struktury np. przedsiębiorstwa	Włączanie do sieci nowych użytkowników wraz z przydzielaniem im odpowiednich uprawnień nie wymaga dużo czasu. Posiadacze komputerów przenośnych mogą błyskawicznie dołączać swój sprzęt w celu pobrania lub składowania danych.

***Tabela 1.** Korzyści płynące z połączenia komputerów w sieć*

Pracą komputerów w sieci zarządza **sieciowy system operacyjny**. Najczęściej wykorzystywane systemy sieciowe to: Linux, Windows Server (w wersjach: 2003, 2008, 2012, 2016 i 2019), różne warianty systemu Unix.

2. Podział sieci

Podział sieci ze względu na wielkość

Sieci lokalne (z ang. **LAN** – *Local Area Network*) – obejmują komputery połączone na ograniczonym obszarze (najczęściej w jednym budynku lub w kilku budynkach), zazwyczaj w ramach jednej jednostki organizacyjnej (szkoła, biuro, firma). Jeśli w sieci lokalnej połączenia zrealizowano bez użycia przewodów, mówimy o **bezprzewodowej sieci lokalnej** (z ang. **WLAN** – *Wireless Local Area Network*);

Sieci miejskie (z ang. **MAN** – *Metropolitean Area Network*) – łączą sieci lokalne na terenie jednego miasta;

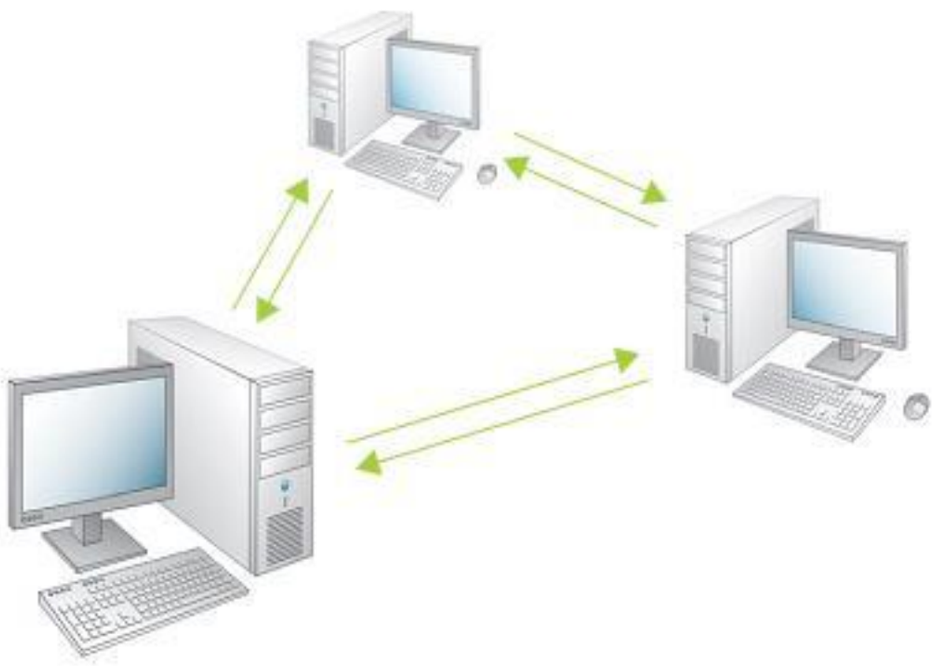
Sieci rozległe (z ang. **WAN** – *Wide Area Network*) – łączą ze sobą wybrane sieci lokalne na większym obszarze;

Internet – łączy ze sobą prawie wszystkie sieci na świecie.

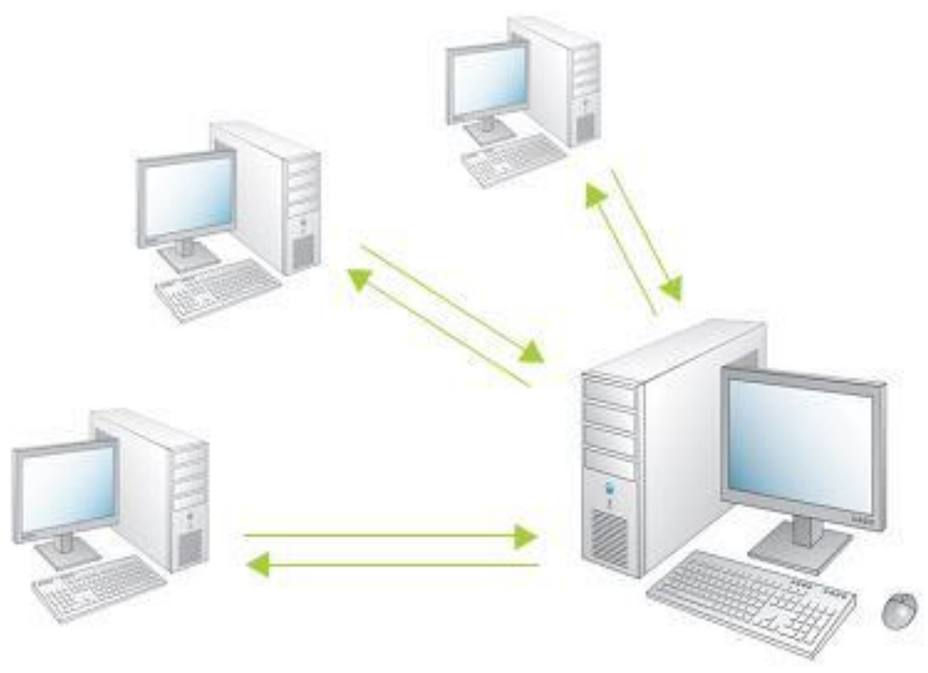
Podział sieci ze względu na model funkcjonowania

Sieć typu „każdy z każdym” (ang. *peer to peer*)

Najczęściej stosowana do łączenia komputerów w domach czy małych firmach, gdzie liczba działających komputerów nie przekracza kilkunastu, a organizacja pracy i użytkowanie oprogramowania nie wymaga szczególnego nadzoru. Każdy komputer będący jej składnikiem ma takie same prawa – może być równocześnie **serwerem** i **klientem**.



Rys. 2. *Sieć typu „każdy z każdym”*



Rys. 3. *Sieć typu „klient-serwer”*

Serwer S

Komputer, który udostępnia zasoby.

Klient K

Komputer, który korzysta z zasobów.

Sieć typu „klient-serwer”

W tym przypadku co najmniej jeden komputer pełni rolę serwera, udostępniając swoje zasoby wielu klientom. Typowy przykład to internetowy serwer WWW, do którego klienci sięgają po interesujące ich informacje.

Ten typ sieci dominuje w szkolnych pracowniach i firmach. Jeśli liczba komputerów przekracza 10, dobrze jest wyznaczyć jednemu z nich rolę nadrzędną (serwera) i składować na nim dane oraz programy, z których będą korzystać

upoważnieni użytkownicy. Dzięki zainstalowaniu na serwerze sieciowego systemu operacyjnego, korzystanie z tego typu sieci jest bardzo efektywne.

Centralizacja danych zapewnia lepszą kontrolę, umożliwia ich integralną archiwizację oraz zwiększa dyscyplinę użytkowania zasobów. Serwer powinien być zawsze włączony, tak by każda upoważniona osoba mogła mieć do niego dostęp (w ramach przydzielonych jej praw) bez konieczności oczekiwania na włączenie komputera przez innego użytkownika. Osoba nadzorująca pracę serwera, czyli administrator, decyduje o tym, komu i jakie zasoby można przydzielić. Określa również sposób korzystania z tych zasobów. Może je np. udostępnić tylko do odczytu lub ograniczyć dostęp czasowo.

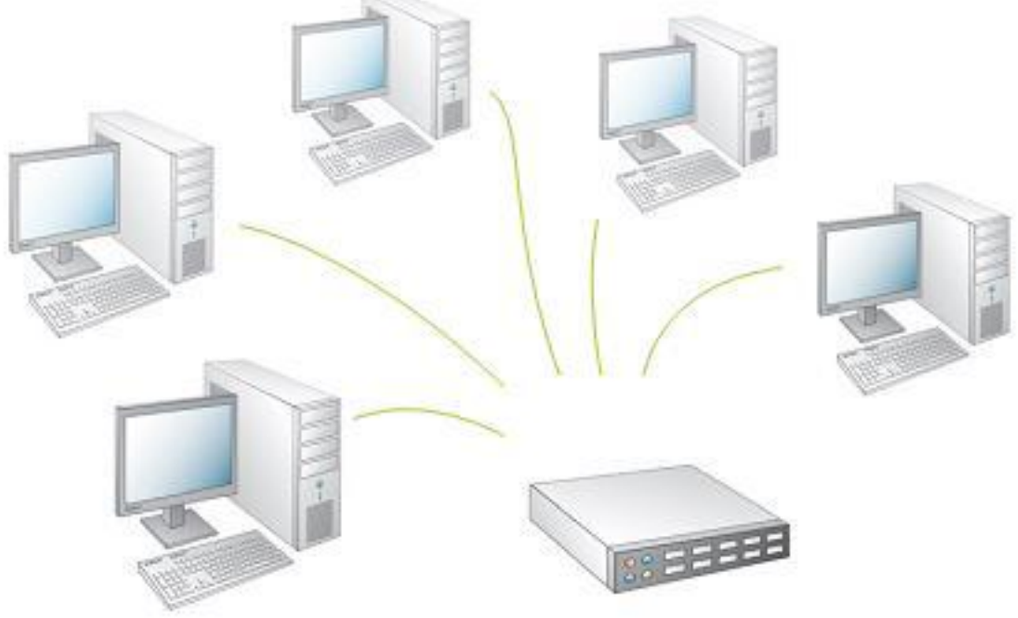
Podział sieci ze względu na topologię (układ przestrzenny)

Topologia gwiazdy (rys. 4.) – komputery są ze sobą połączone przez jeden punkt centralny.

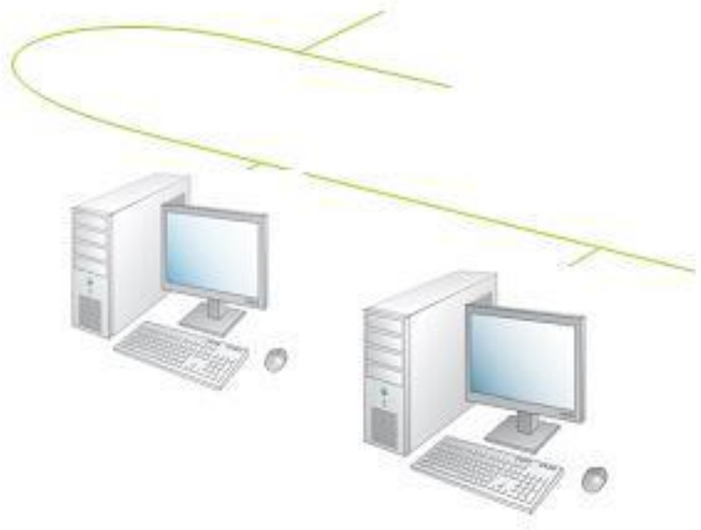
Topologia szyny (magistrali) (rys. 5.) – komputery są dołączone do wspólnego przewodu, szeregowo lub równolegle.

Topologia pierścienia (rys. 6.) – podobna do topologii szyny, ale końce przewodu są ze sobą połączone, tworząc pierścień.

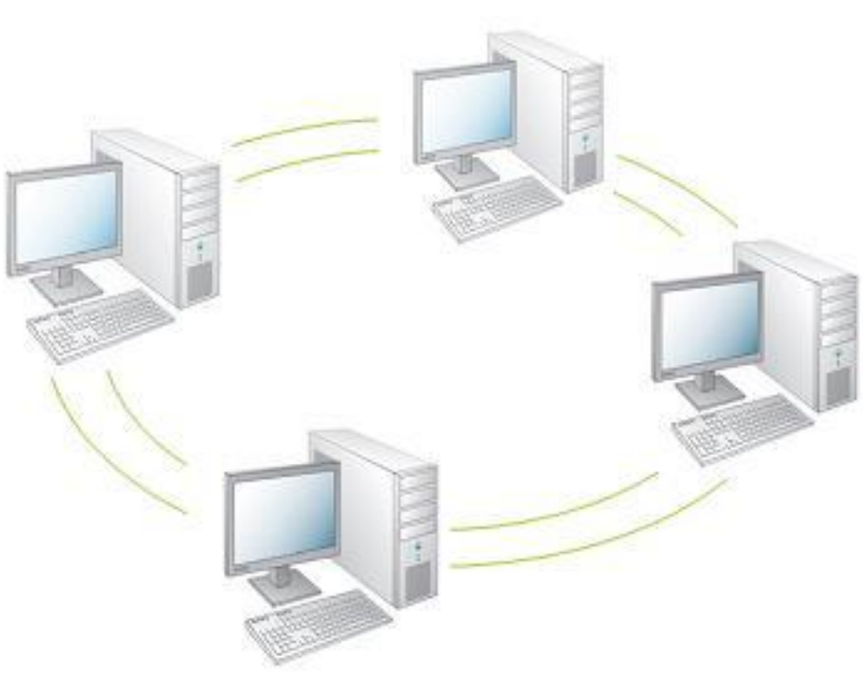
W dużych sieciach komputerowych podane topologie mogą się ze sobą łączyć, np. często spotykany jest układ z wieloma gwiazdami (rys. 7.).



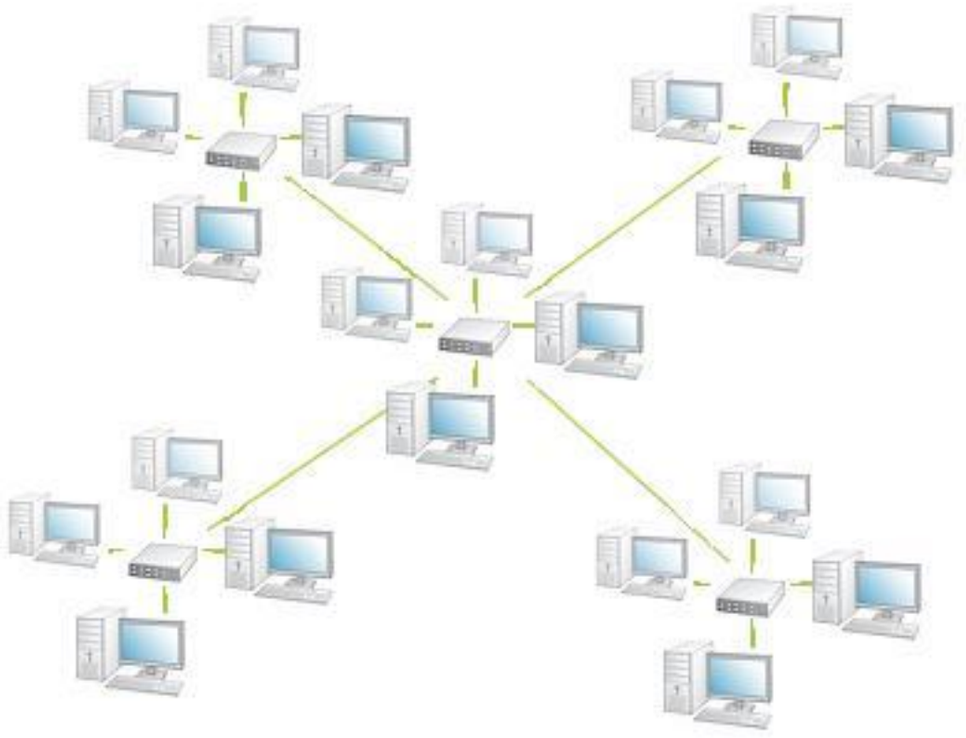
Rys. 4. *Sieć komputerowa w układzie gwiazdy*



Rys. 5. *Sieć komputerowa w układzie szyny*



Rys. 6. *Sieć komputerowa w układzie pierścienia*



Rys. 7. *Sieć komputerowa w układzie wielu gwiazd*

3. Praktyczne sposoby tworzenia sieci komputerowej

Przewodowe sieci lokalne tworzy się najczęściej w standardzie Ethernet, w topologii gwiazdy. W tym układzie realizowane są również radiowe sieci bezprzewodowe.

Przełącznik – ang. *switch*.
Koncentrator – ang. *hub*.

W sieci przewodowej poszczególne komputery są dołączone do jednego centralnego urządzenia, którym najczęściej jest **przełącznik** (dawniej – **koncentrator**), czasami serwer wyposażony w kilka kart sieciowych. Każdy komputer jest podłączony do urządzenia centralnego własnym przewodem, dzięki czemu system uzyskuje wysoką odporność na awarie. Uszkodzenie przewodu powoduje odłączenie od sieci tylko jednej stacji, nie zakłócając pracy pozostałych.

Mbps M
(z ang. *Megabit per second*)
Jednostka szybkości przesyłania danych; liczba megabitów przesyłanych w ciągu sekundy.

Istnieje kilka standardów realizacji sieci, wymagających odpowiedniego sprzętu i różniących się szybkością przesyłania danych. Najczęściej spotykane to: 100BASE-T (szybkość 100 **Mbps**) i 1000BASE-T (szybkość 1000 Mbps).

W sieci bezprzewodowej centralnym urządzeniem jest **punkt dostępowy** (ang. *access point* lub *wireless access point*). Sieci bezprzewodowe również mogą być realizowane w kilku standardach: 802.11b (szybkość do 11 Mbps), 802.11g (szybkość do 54 Mbps), 802.11n (szybkość do 450 Mbps), 802.11ac (szybkość do 6,93 Gbps) lub 802.11ax (szybkość do 14 Gbps). Standardy te różnią się też zasięgiem działania sieci.

Dzięki temu, że transmisja danych odbywa się w różnych pasmach częstotliwości (zwanych **kanałami**), możliwe jest funkcjonowanie kilku sieci bezprzewodowych na jednym obszarze (np. w jednym bloku).

Ćwiczenie 1. Szukamy informacji o standardach sieciowych

Znajdź w Internecie informacje na temat różnych standardów sieciowych. Sprawdź, jak duży zasięg mogą mieć sieci w opisanych standardach.

Ćwiczenie 2. Sprawdzamy czas przesyłania plików w różnych standardach sieci

Przyjmując, że jeden bajt równa się 8 bitom, i korzystając z arkusza kalkulacyjnego, oblicz, ile czasu zajmie przesłanie w różnych standardach sieci przewodowej i bezprzewodowej:

a. dokumentu komputerowego o wielkości 1 MB.
b. utworu muzycznego w formacie MP3 (plik o wielkości 6 MB).
c. filmu w formacie AVI (plik o wielkości 700 MB).

4. Podstawy konfiguracji sieci

Każdy komputer pracujący w sieci powinien mieć zainstalowane: **kartę sieciową**, odpowiedni **protokół sieciowy**, oprogramowanie **klienta sieci** oraz oprogramowanie realizujące **usługi**, z których chcemy korzystać.

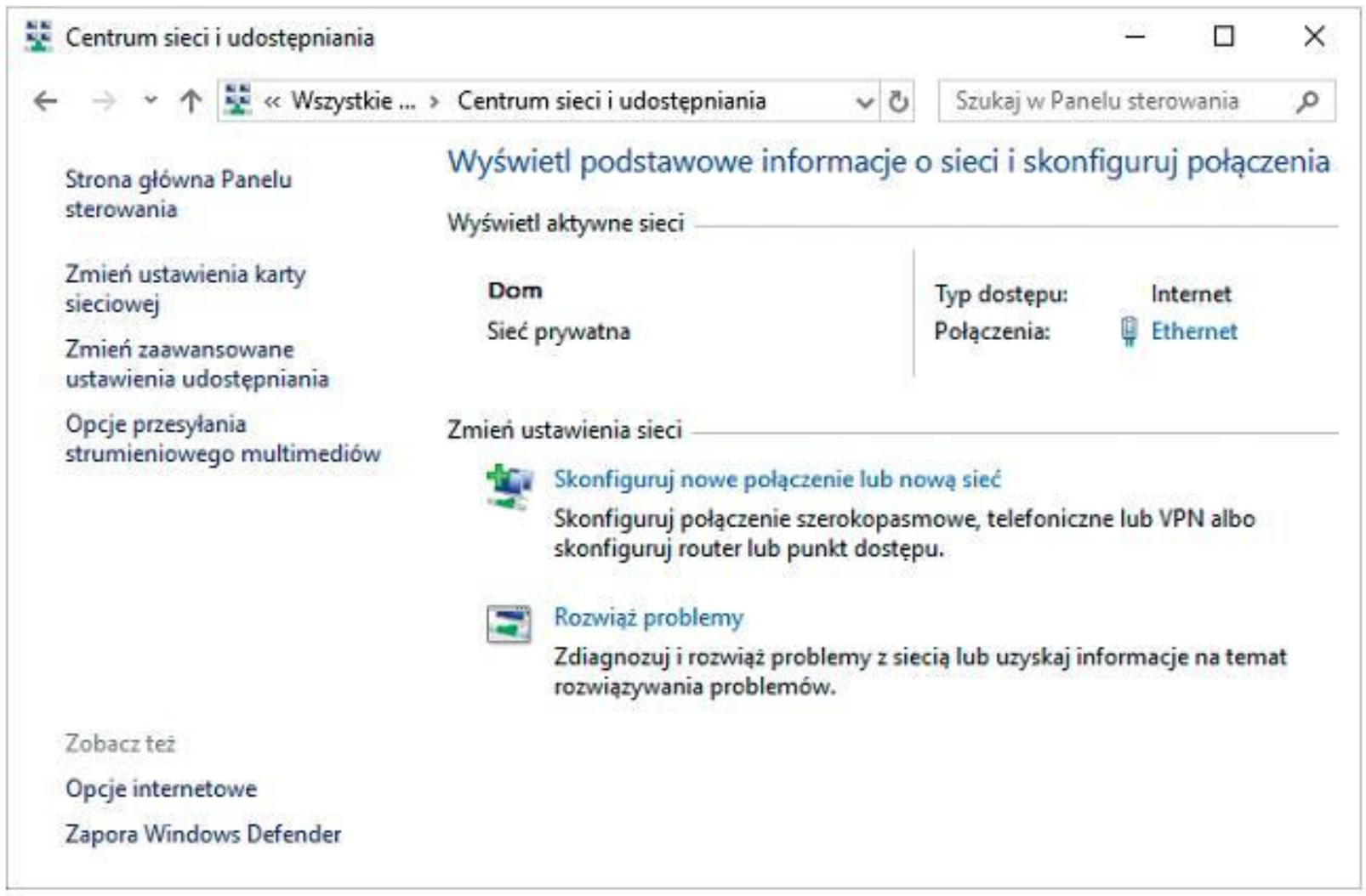
Karta sieciowa to urządzenie umożliwiające połączenie z innymi komputerami. Większość komputerów dostępnych obecnie w sprzedaży jest fabrycznie wyposażona w taką kartę, zintegrowaną z płytą główną. W przypadku komputerów przenośnych (laptopów, tabletów, smartfonów) jest to zwykle karta sieci bezprzewodowej. Do komputera można podłączyć zewnętrzną kartę sieciową, najczęściej za pomocą portu USB.

Protokół sieciowy to zbiór reguł, którym podlega komunikacja pomiędzy komputerami. Reguły te określają sposób adresowania informacji, dzielenia ich na fragmenty (zwane pakietami) oraz kontrolowania poprawności transmisji danych.

Klient sieci to oprogramowanie pozwalające korzystać z zasobów sieci pracującej pod kontrolą konkretnego systemu operacyjnego.

Usługa to określenie funkcji realizowanej przez komputer pracujący w sieci. Najbardziej popularną usługą jest udostępnianie folderów, plików i drukarek.

Podstawowe składniki konfiguracji sieci można zobaczyć, wybierając w oknie **Panel sterowania** polecenie **Sieć i Internet/Centrum sieci i udostępniania** (Windows 7, 8 i 8.1) lub **Centrum sieci i udostępniania** (Windows 10 i 11).



Rys. 8. *Otwarte okno* ***Centrum sieci i udostępniania*** *(Windows 10 i 11)*

Uwaga

W systemie Windows 10 ustawienia systemu operacyjnego (w tym **Sieć i Internet**) można znaleźć w aplikacji **Ustawienia**, dostępnej z menu **Start**.

4.1. Protokoły sieciowe

Najbardziej popularne rodzaje protokołów sieciowych to: NetBEUI, IPX/SPX, TCP/IP. Najczęściej stosowanym protokołem (zbiorem protokołów) jest protokół **TCP/IP**, używany zarówno w sieciach rozległych (Internet), jak i w małych sieciach lokalnych. Każdy pracujący w sieci komputer musi mieć adres **IP**, bez względu na to, czy jest podłączony do Internetu, czy tylko do sieci lokalnej.

Adres IP nadawany jest przez administratora konkretnej sieci lub przydzielany automatycznie przez odpowiednie oprogramowanie (tzw. **serwer DHCP**). Serwer DHCP jest często elementem **routerów** wykorzystywanych do budowy sieci domowych.

Router R

Urządzenie sieciowe umożliwiające przesyłanie pakietów danych pomiędzy różnymi sieciami komputerowymi.



Różne komputery pracujące w obrębie tej samej sieci nie mogą mieć takich samych adresów IP.

Oktet O

Oznacza osiem bitów. W postaci dziesiętnej przedstawiany jest jako liczba z zakresu 0-255. Teoretycznie za pomocą czterech oktetów można przydzielić 4 294 967 296 adresów, bo 256 x 256 x 256 x 256 = $= 256^4 =$ = 4 294 967 296.

Najczęściej obecnie używana wersja protokołu (IPv4) stosuje adresy IP w formacie 32-bitowej liczby. Dla ułatwienia jest ona zapisywana w postaci czterech **oktetów** oddzielonych kropkami (np. 192.168.100.100).

Ważnym, zastrzeżonym adresem jest np. adres 127.0.0.1, zarezerwowany dla lokalnego komputera – stąd jego nazwa: **localhost**.



Każdy komputer pracujący w sieci opartej na protokole TCP/IP nazywamy **węzłem** (ang. *host*).



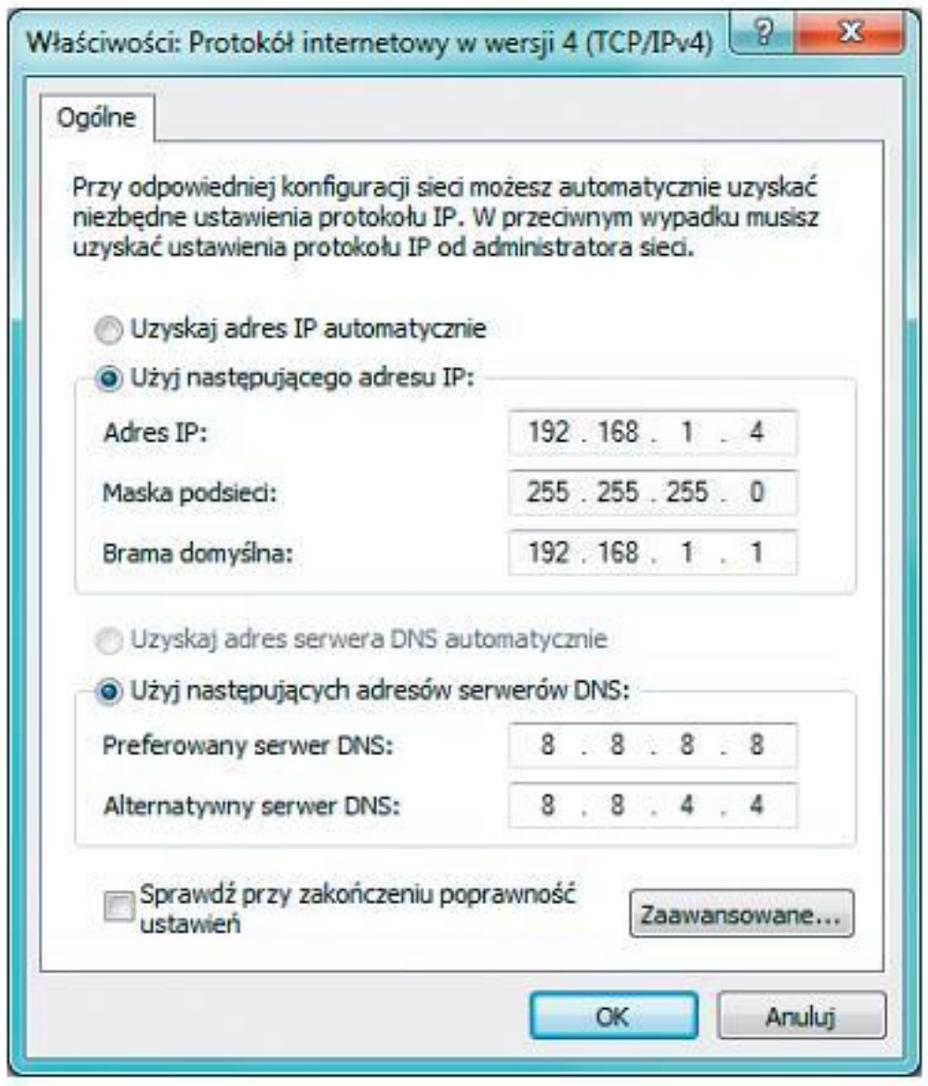
Przykład 1. Konfigurowanie protokołu TCP/IP

Aby otworzyć okno **Właściwości protokołu internetowego TCP/IP**, należy:

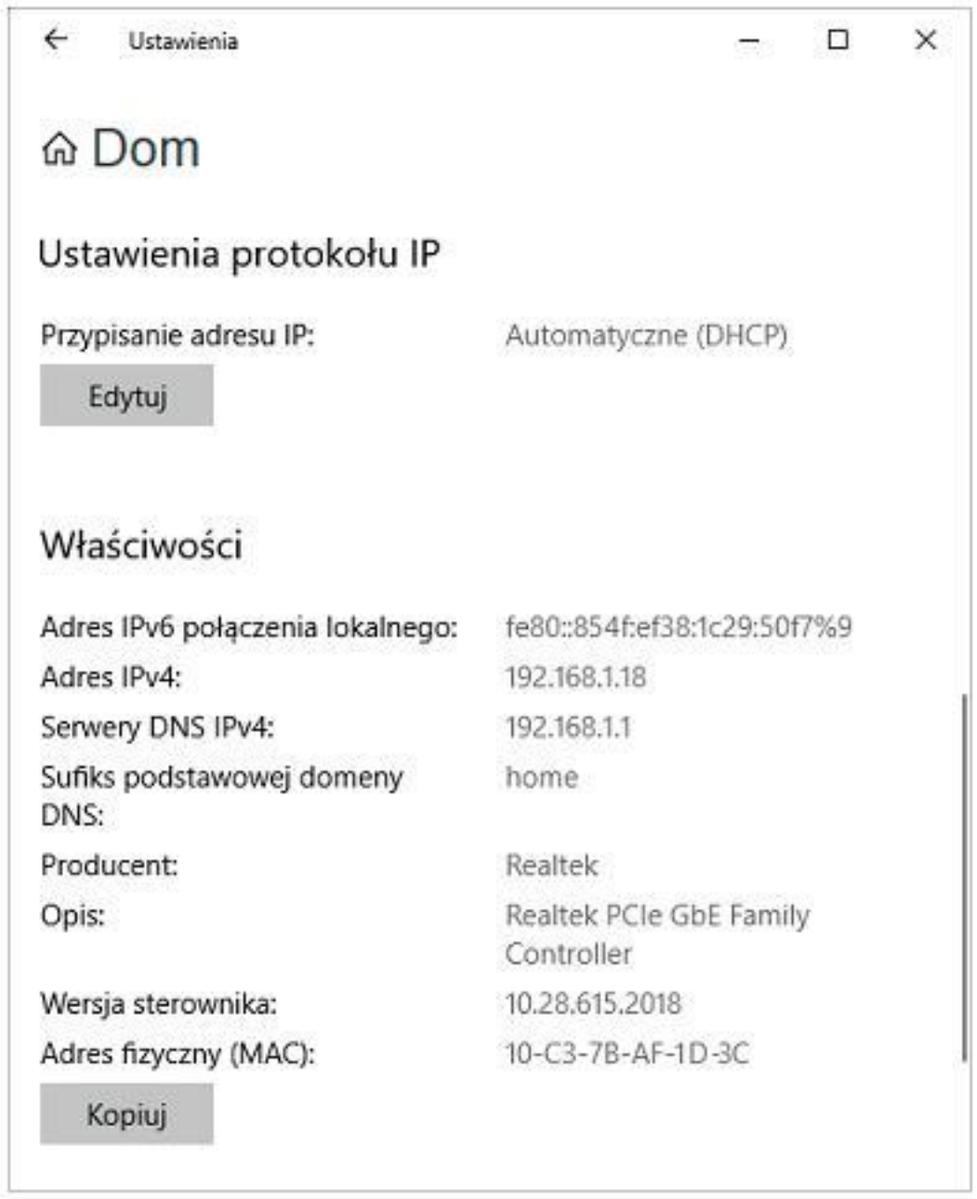
- w **Windows 7, 8, 8.1** – w oknie **Centrum sieci i udostępniania** (rys. 8.) wybrać polecenie **Połączenie lokalne**, w otwartym oknie kliknąć przycisk **Właściwości**, a w kolejnym oknie – polecenie **Protokół internetowy w wersji 4 (TCP/IPv4)**; otworzy się okno pokazane na rys. 9a.
- w **Windows 10** i **11** – w aplikacji **Ustawienia** wybrać polecenie **Sieć i Internet**; następnie w Windows 10 wybrać **Stan/Zmień właściwości połączenia** (rys. 9b), a w Windows 11 **Ethernet/Edytuj/Ręczne/IPv4**.

Uwagi:

- Pokazany na rys. 9a i 9b adres (192.168.1.4) należy do puli tzw. adresów wewnętrznych. W sieciach lokalnych powinno się korzystać z takich adresów.
- Komputery pracujące w tej samej sieci muszą mieć takie same początkowe liczby adresu (maska podsieci określa, ile bitów adresu powinno być zgodnych).
- Jeżeli w sieci pracuje serwer DHCP, powinny zostać zaznaczone opcje **Uzyskaj adres IP automatycznie** oraz **Uzyskaj adres serwera DNS automatycznie**.



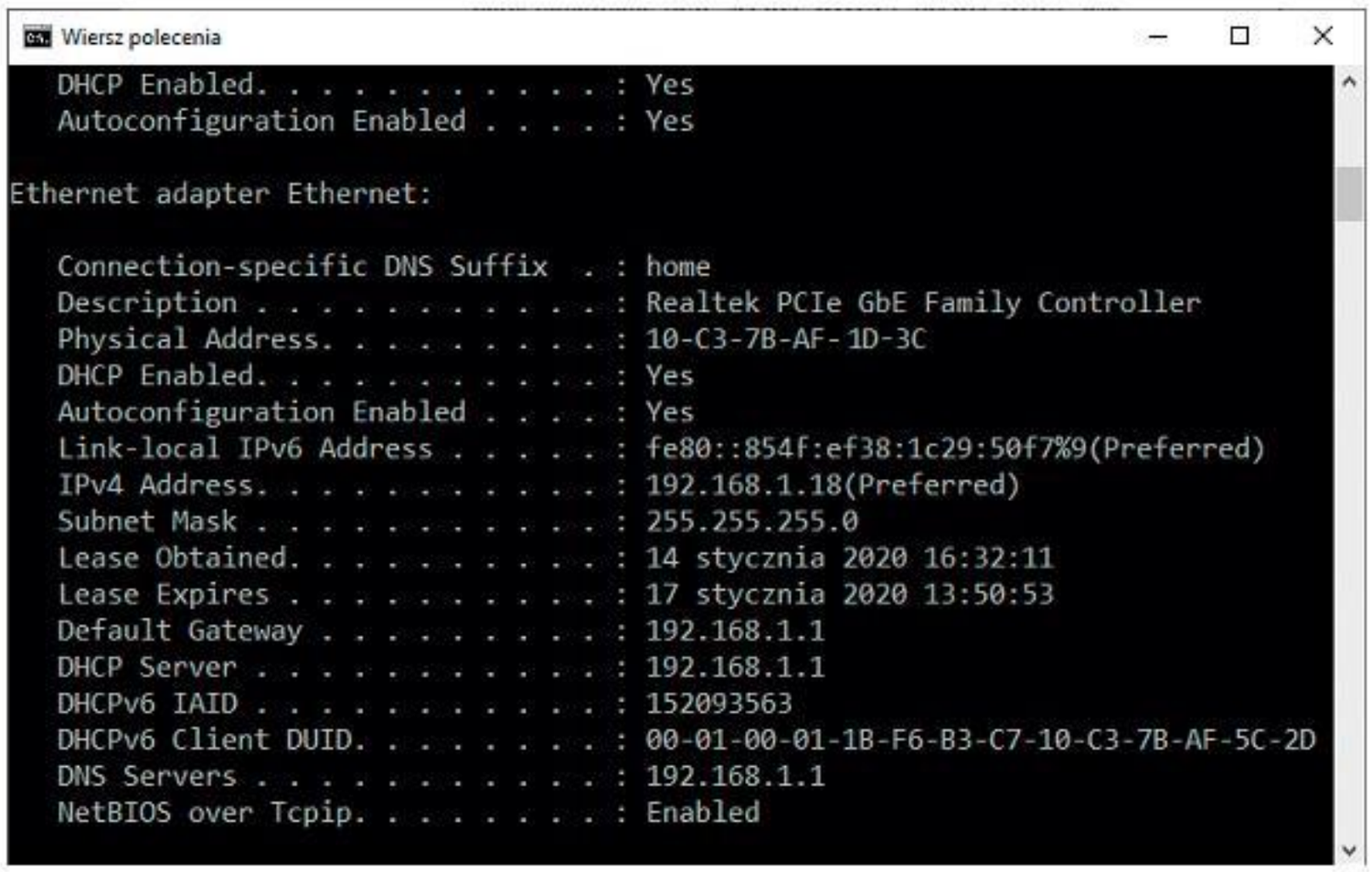
Rys. 9a. *Okno właściwości protokołu internetowego TCP/IP (Windows 7, 8, 8.1)*



Rys. 9b. *Ustawienia protokołu internetowego TCP/IP (Windows 10)*

Aby sprawdzić adres IP komputera, należy w **Wierszu polecenia** wpisać: *ipconfig /all*.

Wiersz polecenia znajdziemy na ekranie **Aplikacje** lub wpisując „Wiersz polecenia” w polu wyszukiwania na ekranie **Start**.



Rys. 10. *Okno konfiguracji protokołu IP z widocznymi: adresem IP komputera, maską podsieci, adresem bramy internetowej, adresem fizycznym karty sieciowej*

Ćwiczenie 3. Sprawdzamy IP komputera

Sprawdź adres IP swojego komputera.

4.2. Identyfikacja sieciowa

Każdy komputer pracujący w sieci posiada identyfikującą go **Nazwę** oraz jest przydzielony do konkretnej **Grupy**.

Zespołowi komputerów wykonujących zbliżone funkcje system Windows nadaje umowną nazwę: **Grupa robocza** (ang. *workgroup*). Grupą roboczą może być np. *Księgowość* w dużej firmie, *Pracownia informatyczna* w szkole, czy grupa *Rodzice* w sieci domowej. W małych grupach roboczych komputery często nazywamy imieniem osoby, która je obsługuje.

Aby w Windows 7 szybko wyszukać dane polecenie, np. **Grupa domowa**, można je wpisać w polu **Wyszukaj programy i pliki** w menu **Start**; a w nowszych wersjach (Windows 8, 8.1, 10) – w polu **Wyszukaj** na ekranie **Start**.

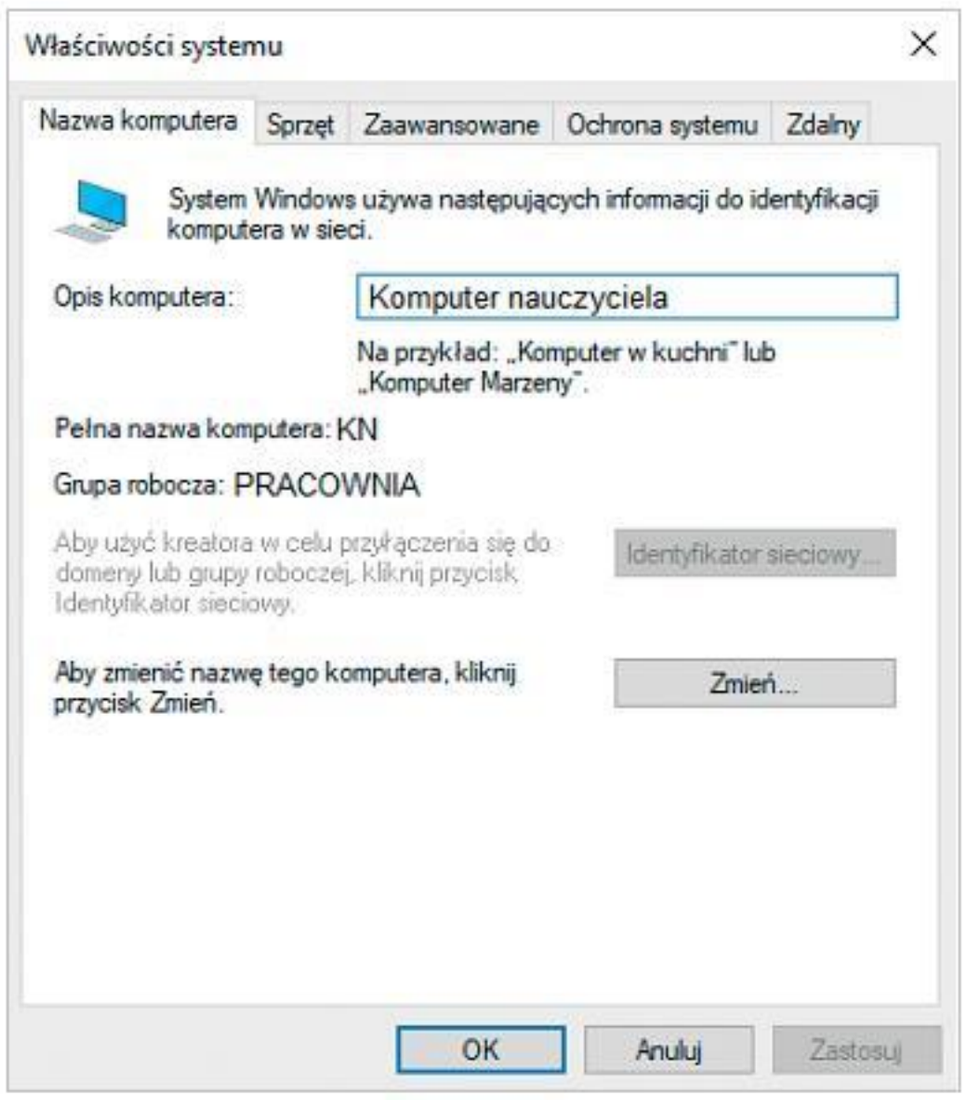
Przykład 2. Identyfikacja sieciowa

Aby otworzyć okno **Właściwości systemu**, należy w oknie **Panel sterowania** wybrać:

- w Windows 7 opcje **System i zabezpieczenia/System** i kliknąć **Zmień ustawienia**,
- w Windows 10 opcje **System/Zaawansowane ustawienia systemu.**

W zakładce **Nazwa komputera** (rys. 11.) należy wpisać wybraną przez siebie lub nadaną przez administratora nazwę komputera.

Jeżeli chcemy zmienić nazwę komputera lub grupy roboczej, należy kliknąć przycisk **Zmień**, a następnie ustalić te parametry w okienku **Zmiana nazwy komputera** (pole **Domena:** wypełnia się w przypadku dużych sieci w przedsiębiorstwach).



Rys. 11. *Okno* ***Właściwości systemu*** *(Windows 10)*

Aby możliwe było udostępnianie zasobów w sieci, komputery muszą należeć do tej samej grupy roboczej.

W systemie operacyjnym Windows 7 (i w nowszych wersjach) można w prosty sposób połączyć komputery w domu lub w małej firmie w tzw. **Grupę domową**. Takie połączenie pozwala na automatyczne udostępnianie drukarek i bibliotek (muzyki, filmów, dokumentów) innym użytkownikom grupy, pod warunkiem, że wszystkie komputery mają zainstalowany system Windows 7 (lub nowszy).

Ćwiczenie 4. Sprawdzamy konfigurację sieci w pracowni komputerowej

Sprawdź na komputerze, na którym pracujesz, jak skonfigurowana jest sieć w szkolnej pracowni.

Ćwiczenie 5. Szukamy możliwości tworzenia grupy roboczej

Jeśli na twoim komputerze zainstalowano system Windows 7 lub nowszy, odszukaj możliwość tworzenia **Grupy domowej**.

5. Co jest niezbędne do budowy sieci?

5.1. Sieć domowa

Sieci domowe tworzy się najczęściej po to, aby wielu komputerom (a także urządzeniom elektroniki użytkowej) należącym do domowników zapewnić dostęp do Internetu.

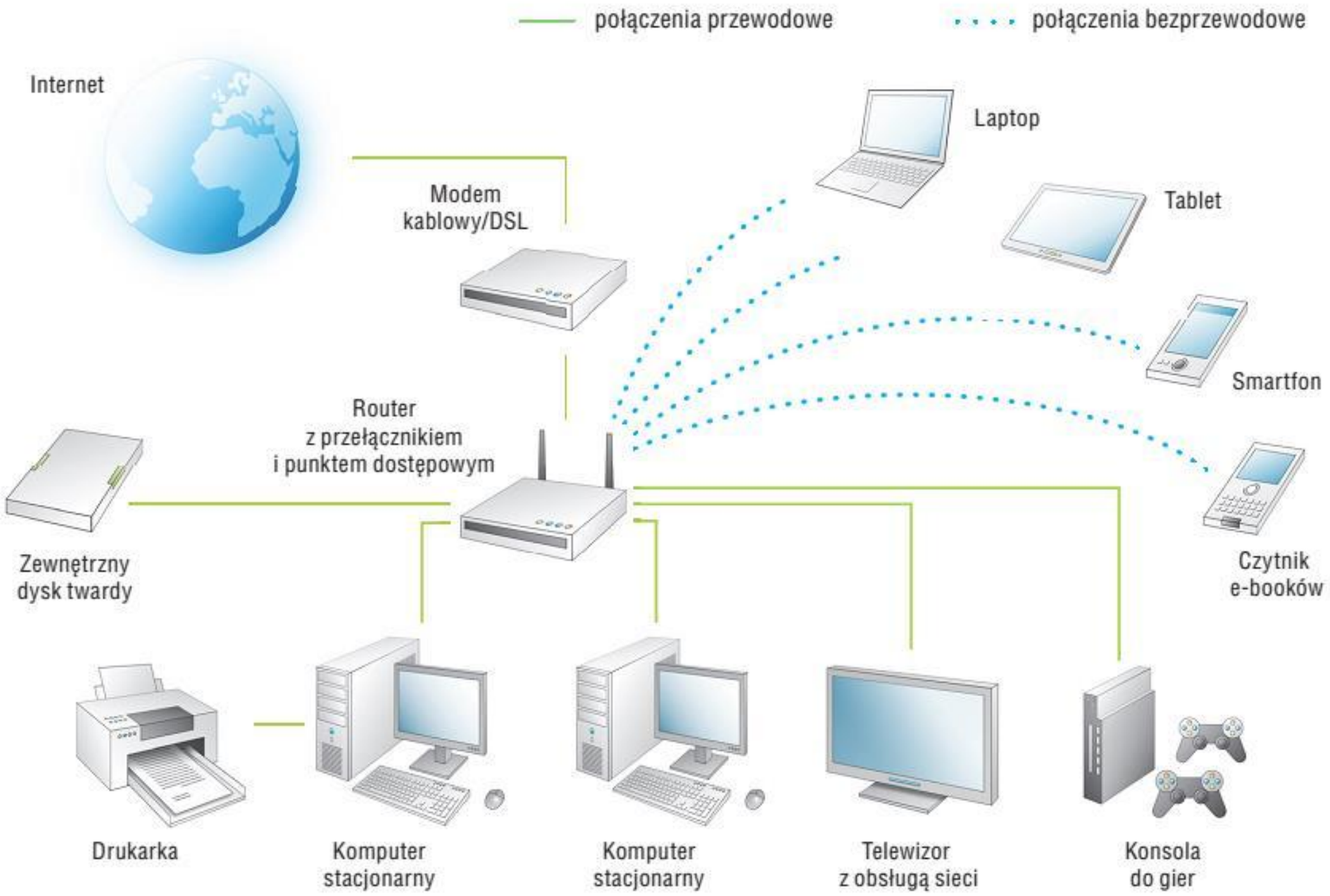
Dostawca usług internetowych zapewnia zwykle jeden modem kablowy (w przypadku udostępniania Internetu za pomocą telewizji kablowej) lub jeden modem DSL (w przypadku udostępniania Internetu za pomocą łączy telefonicznych). Aby umożliwić dostęp większej liczbie komputerów, należy skorzystać z routera, który z jednej strony łączy się z modemem, a z drugiej – z komputerami pracującymi w domu (rys. 13.). Dostępne obecnie na rynku routery posiadają często funkcję przełącznika i punktu dostępowego, dzięki czemu można do nich podłączać zarówno komputery pracujące przewodowo, jak i bezprzewodowo. Zdarzają się także urządzenia łączące funkcję modemu, routera i punktu dostępowego (rys. 12.).

Aby sieć bezprzewodowa miała jak największy zasięg (i szybkość przesyłania danych), punkt dostępowy powinien znajdować się jak najbliżej centrum domu i być zawieszony wysoko, daleko od innych urządzeń wytwarzających pole elektromagnetyczne. Sieci bezprzewodowe najlepiej działają w pomieszczeniach o drewnianych ścianach, a najgorzej – w pomieszczeniach o ścianach z żelazobetonu (żelbetonu).



Rys. 12. *Router z funkcją modemu, przełącznika i punktu dostępowego*

Router należy skonfigurować do pracy, na przykład za pomocą oprogramowania dołączonego na płycie CD-ROM. Jeżeli Internet jest dostarczany za pomocą linii DSL, zwykle należy wprowadzić identyfikator i hasło dostępu do Internetu, udostępnione przez dostawcę usług internetowych. Większość routerów posiada wbudowany serwer DHCP, dzięki czemu parametry protokołu TCP/IP kolejno podłączanych urządzeń konfigurują się automatycznie. Dostęp do routera powinien zostać zabezpieczony hasłem. Sieć bezprzewodową należy nazwać, wymyślając własną nazwę (tzw. SSID).



Rys. 13. *Schemat przykładowej domowej sieci komputerowej*

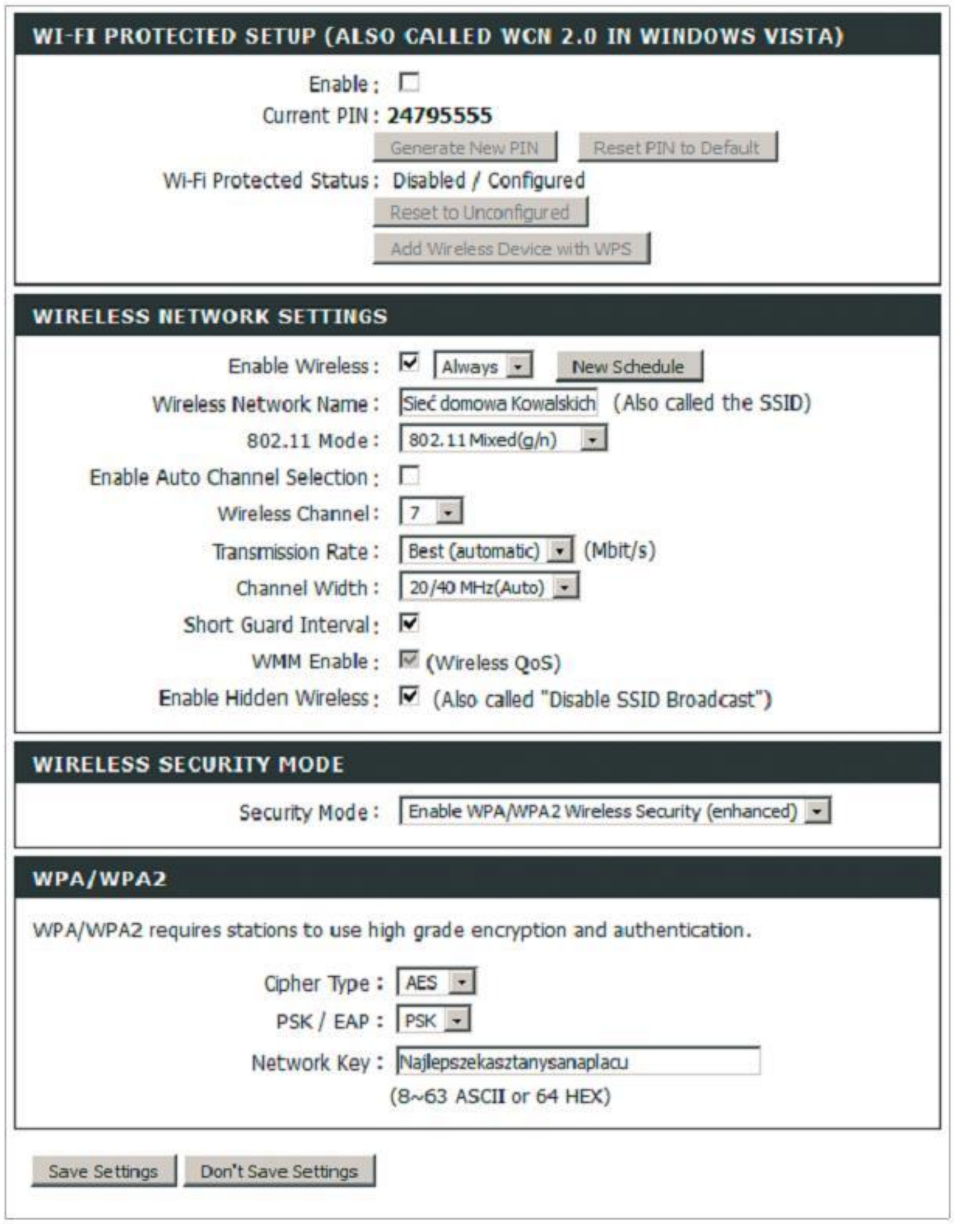


Nie należy pozostawiać domyślnej nazwy sieci bezprzewodowej, proponowanej przez punkt dostępowy, gdyż stanowi to zagrożenie bezpieczeństwa sieci.

W przypadku sieci bezprzewodowych kwestie bezpieczeństwa są bardzo ważne. Niedopuszczalne jest tworzenie otwartej sieci bezprzewodowej, z której każdy mógłby skorzystać. Niepowołane osoby mogłyby wykorzystać naszą sieć, np. do popełniania przestępstw komputerowych na nasze konto.

Dostęp do sieci bezprzewodowej należy zabezpieczyć, wymyślając hasło, które zostanie ustawione w punkcie dostępowym, a następnie na komputerach podłączanych bezprzewodowo. Istnieje kilka protokołów bezpieczeństwa – w miarę możliwości należy korzystać z protokołu WPA2, ewentualnie WPA. Starszy protokół WEP nie zapewnia praktycznie żadnego bezpieczeństwa – hasło dostępowe tego protokołu można złamać w ciągu kilku minut za pomocą powszechnie dostępnych programów.

Z zasobów sieci (np. z utworów muzycznych i filmowych zgromadzonych na dysku komputera) możemy korzystać za pośrednictwem wielu urządzeń elektroniki użytkowej (np. nowoczesnych telewizorów, odtwarzaczy multimedialnych, sprzętu Hi-Fi). Umożliwia to protokół DLNA.



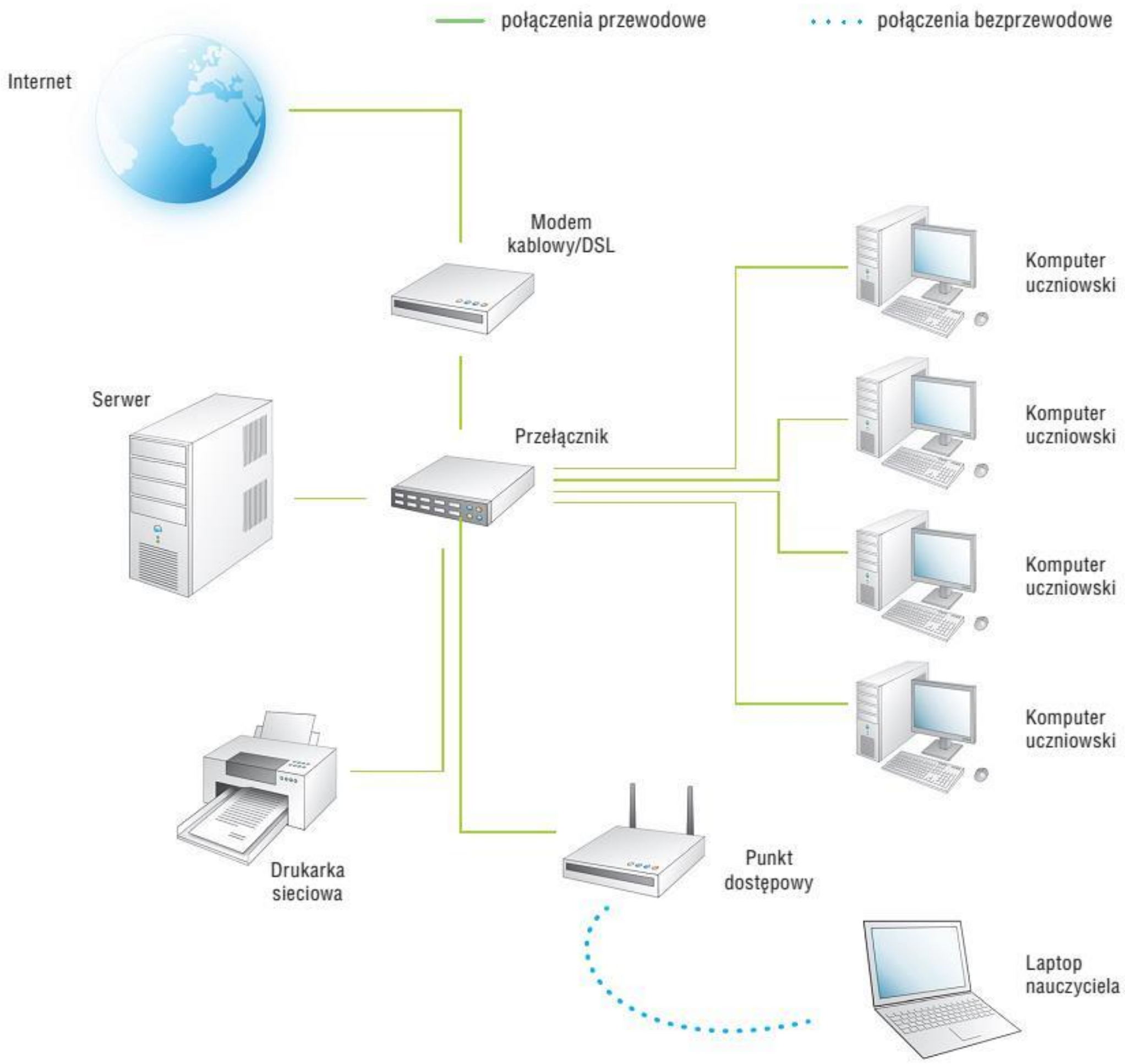
Rys. 14. *Konfiguracja dostępu do sieci bezprzewodowej w routerze*

5.2. Szkolna sieć komputerowa

Szkolne sieci komputerowe tworzy się na podobnych zasadach jak sieci w przedsiębiorstwach.

Oprócz zapewnienia komputerom dostępu do Internetu ważne jest efektywne udostępnianie zasobów – stąd zazwyczaj jeden z komputerów pełni rolę serwera (a także, dzięki odpowiedniemu oprogramowaniu, routera). W takiej sieci rozdziela się zazwyczaj funkcje urządzeń, np. punkty dostępowe są często oddzielnymi urządzeniami i może ich być wiele (tak, aby swoim zasięgiem pokryły całą szkołę).

Adresy IP, hasła i inne parametry sieci komputerowej definiuje najczęściej administrator i nie powinny być one samodzielnie zmieniane. Schemat przykładowej sieci szkolnej pokazano na rysunku 15.



Rys. 15. *Schemat przykładowej szkolnej sieci komputerowej*

6. Podstawy pracy w sieci

6.1. Logowanie

Każdy użytkownik sieci powinien posiadać swój identyfikator oraz hasło. Wymóg ten nie jest istotny w małych sieciach domowych, ale powinien być bezwzględnie przestrzegany w sieciach z serwerem. Identyfikator i hasło początkowe nadaje administrator sieci. Hasło (za zgodą administratora) może zostać zmienione przez użytkownika, np. przy pierwszym korzystaniu z komputera sieciowego.

Proces wprowadzania identyfikatora i hasła użytkownika, który rozpoczyna pracę w sieci, nazywamy **logowaniem**.
Nigdy nie należy podawać swojego hasła innym osobom.

Identyfikator i hasło są sprawdzane z listą użytkowników i odpowiadających im haseł na serwerze. Po pozytywnej weryfikacji użytkownik zostaje dopuszczony do pracy.

W niektórych systemach prawidłowe zakończenie pracy wymaga wylogowania.

6.2. Udostępnianie zasobów

Sieć zyskuje pełną funkcjonalność, kiedy umożliwia dostęp do zasobów innych komputerów. Gwarantuje to usługa **Udostępnianie plików i drukarek** (rys. 16a i 16b).

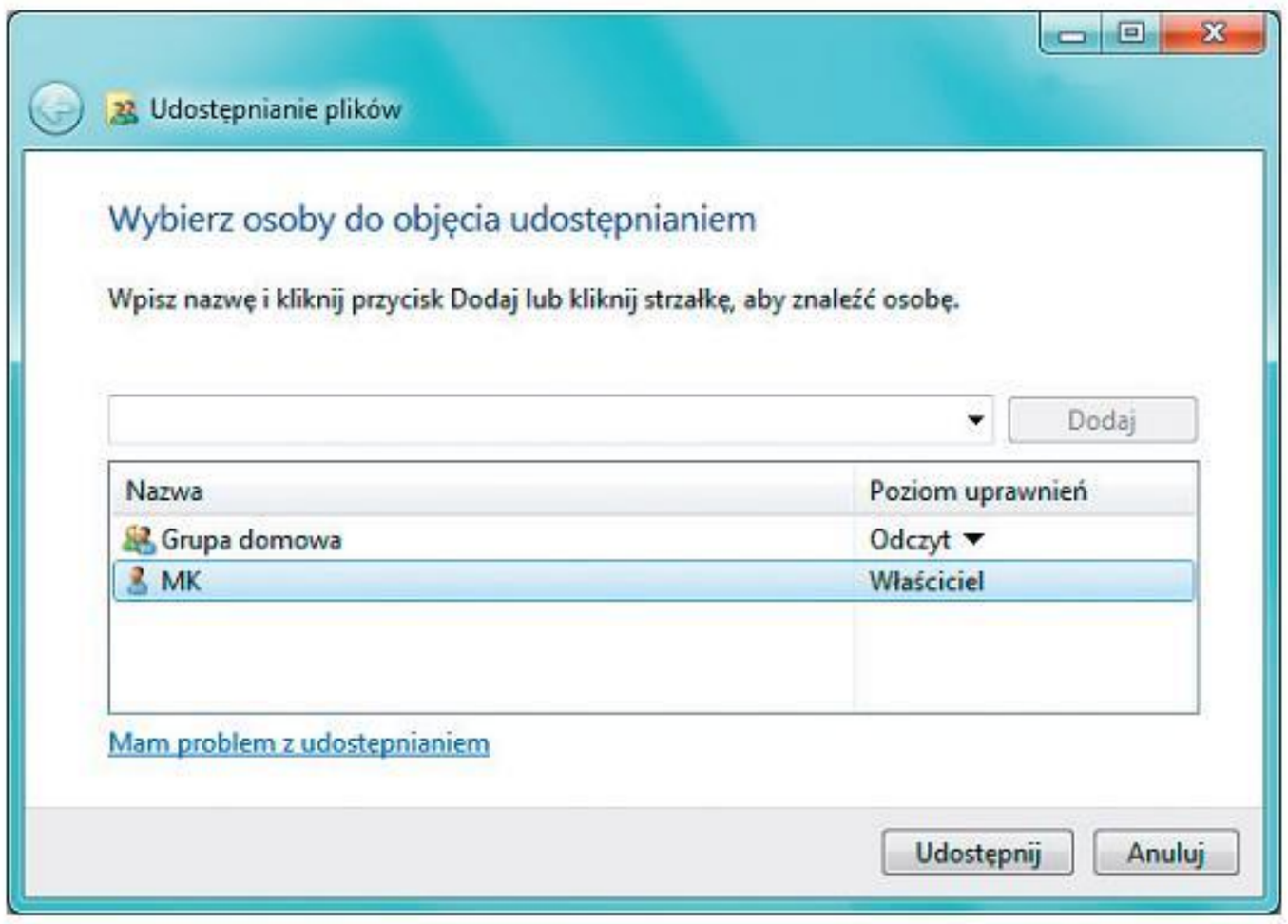
U **Udostępnianie**

Zezwolenie na korzystanie z zasobów lokalnego komputera przez innych użytkowników sieci.

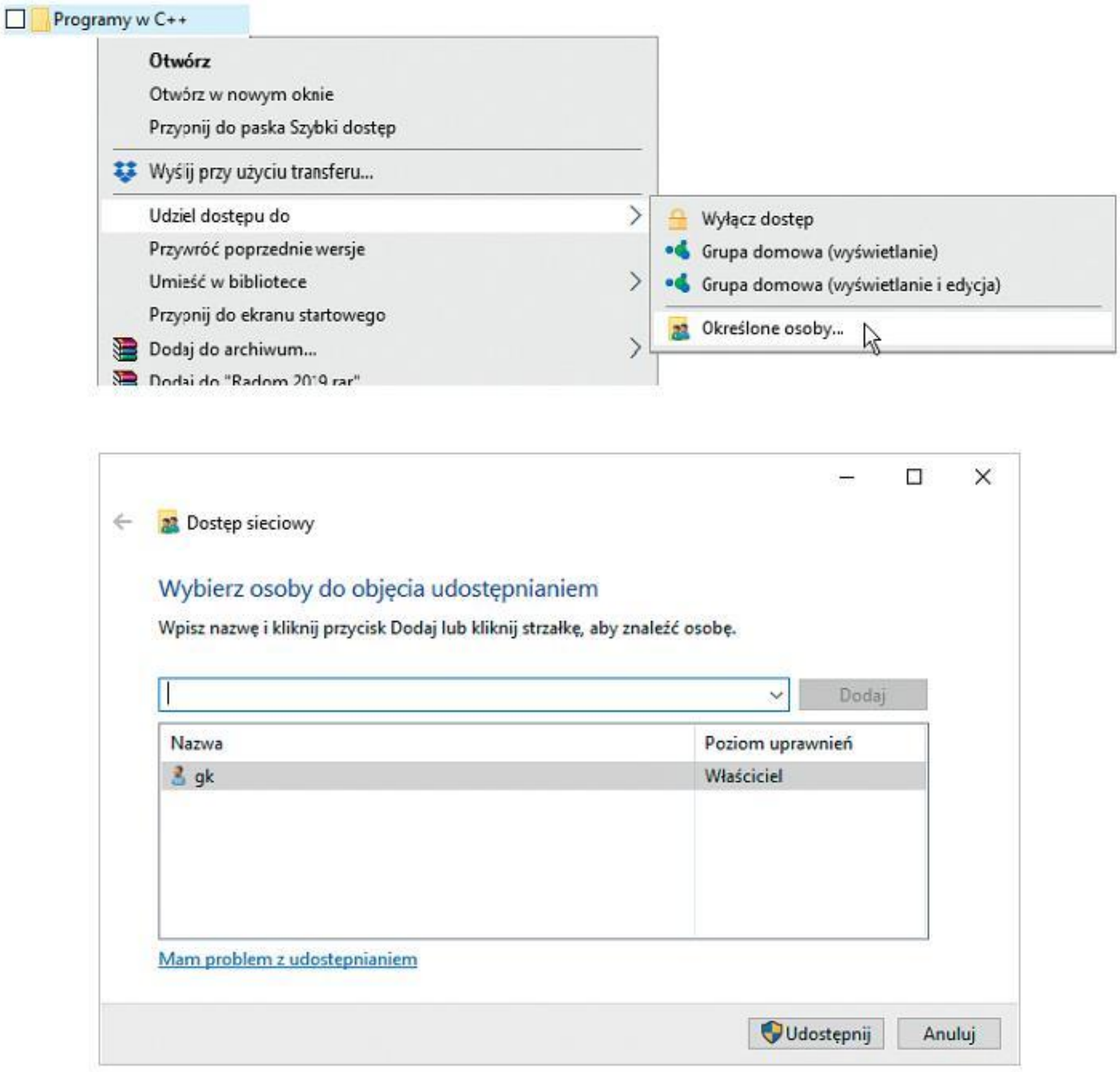
Przykład 3. Udostępnianie zasobów

Dyski i foldery, które mają być udostępnione, można wskazać, korzystając z **Eksploratora plików (Eksploratora Windows)**.
Aby udostępnić zasób, należy kliknąć prawym przyciskiem myszy ikonę dysku lub folderu i:

- w Windows 7, 8 i 8.1 – po wybraniu odbiorcy, czyli określeniu, komu chcemy udostępnić zasób, np. **Grupie domowej** lub **Określonym osobom**, otworzy się okno (rys. 16a), w którym wybieramy np. konkretne osoby. Po tej operacji przy ikonie folderu lub dysku pojawi się rysunek przedstawiający kłódkę.
- w Windows 10 – wybrać polecenie **Udziel dostępu do**; po wybraniu polecenia **Konkretne osoby** otworzy się okno jak na rysunku 16b.



***Rys. 16a**. Okno **Udostępnianie plików** (Windows 7)*



***Rys. 16b**. Okno **Dostęp sieciowy** (Windows 10 i 11)*



Nigdy nie należy udostępniać: plików systemowych, sterowników, danych osobistych.



Ćwiczenie 6. Udostępniamy folder

Udostępnij folder *Moje obrazy* komputera, na którym pracujesz, kolegom z twojej grupy roboczej.

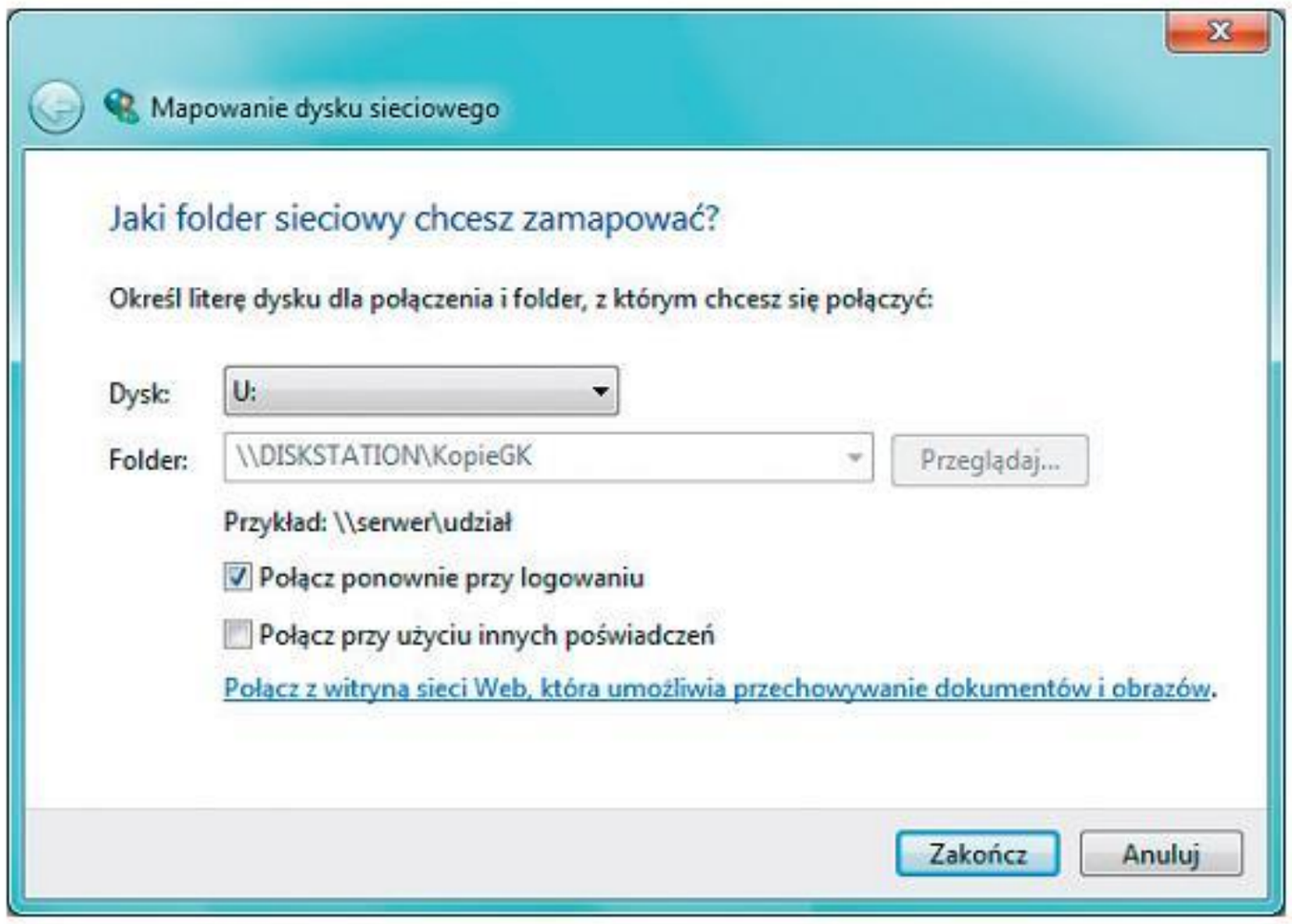
6.3. Mapowanie zasobów

Mapowanie M

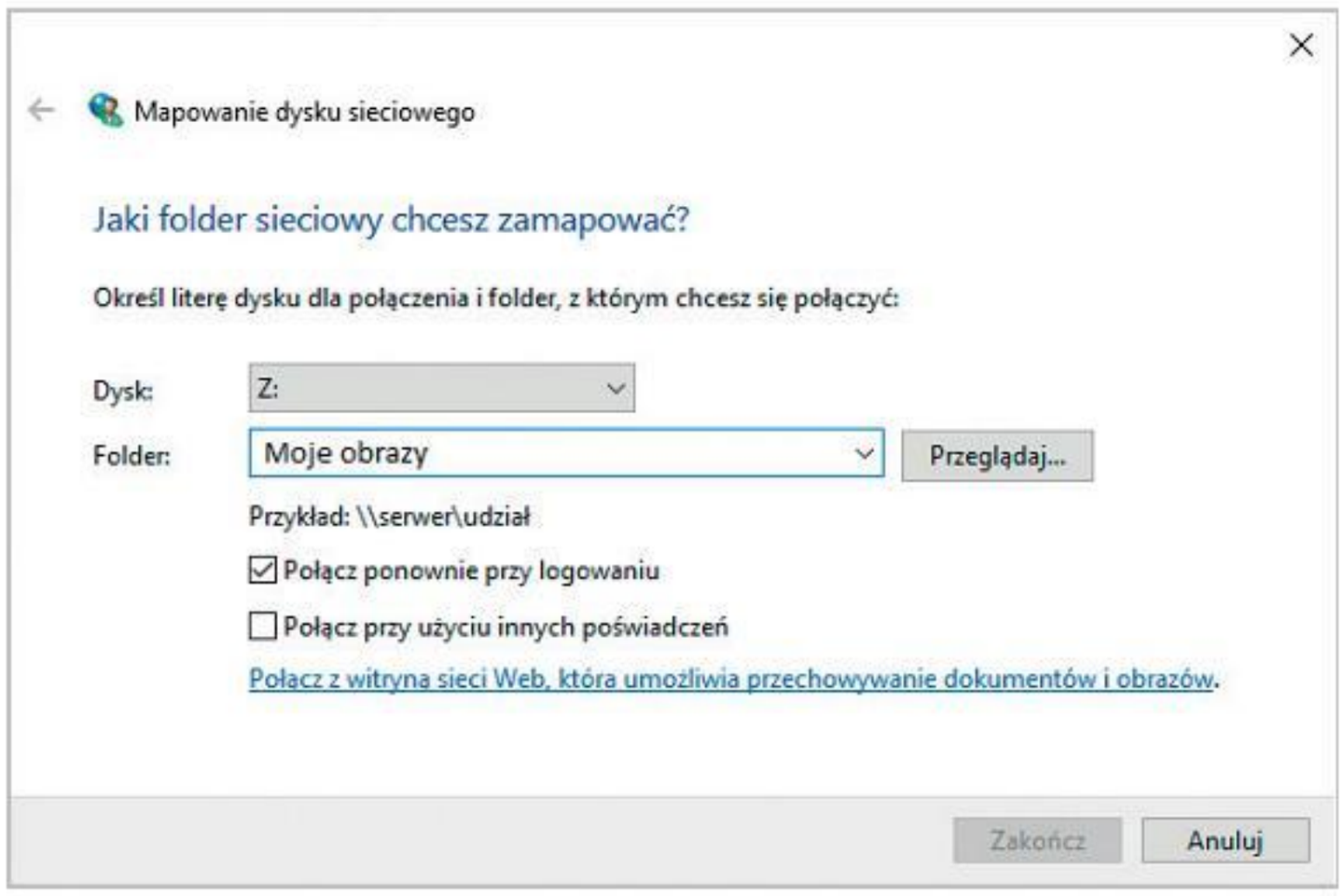
Przyporządkowanie oznaczenia dysku komputera lokalnego wybranemu zasobowi na zdalnym komputerze.

Udostępniony zasób jest widoczny na innych komputerach w ich **Otoczeniu sieciowym**. Aby uzyskać dostęp do zasobu, konieczne jest otwieranie kolejnych okien folderów.

Dostęp do zasobu można uprościć, wykonując **mapowanie** – w tym celu z menu kontekstowego nazwy dysku (np. *Ten komputer* w Windows 10) należy wybrać polecenie **Mapuj dysk sieciowy**.



***Rys. 17a**. Zasób udostępniony na serwerze jest widoczny jako dysk U: komputera lokalnego (Windows 7)*



***Rys. 17b**. Zasób udostępniony na serwerze jest widoczny jako dysk Z: komputera lokalnego (Windows 10 i 11)*

Podobnie jak foldery czy dyski, udostępnić możemy drukarki. Aby korzystać z udostępnionej drukarki, należy ją zainstalować na komputerze, z którego będziemy wysyłać pliki do druku. Proces instalacji przebiega podobnie, jak w przypadku instalacji drukarki lokalnej, jednak w czasie instalacji należy zaznaczyć odpowiednią opcję, np. **Drukarka sieciowa**. W polu z poleceniem podania ścieżki dostępu do drukarki sieciowej można użyć przycisku **Przeglądaj** i, wędrując przez drzewo folderów, wskazać drukarkę podłączoną do innego komputera.

W zależności od używanego systemu operacyjnego komputer pobiera sterowniki obsługujące drukarkę z komputera udostępniającego drukarkę albo prosi o dysk instalacyjny i dokonuje instalacji na komputerze lokalnym. Od tej chwili wybranie polecenia drukowania na drukarce sieciowej spowoduje przesłanie danych do komputera zdalnego, uruchomienie drukarki i wydrukowanie naszego dokumentu.

Ćwiczenie 7. Omawiamy sposób udostępniania drukarki

Sprawdź, czy drukarka znajdująca się w szkolnej pracowni jest udostępniona w sieci. Jeżeli tak, to omów sposób jej udostępnienia.

Warto zapamiętać

- Komputery łączy się w sieć m.in. w celu udostępniania zasobów (folderów, dysków, urządzeń), wymiany danych i korzystania ze wspólnych urządzeń, np. drukarki.
- Ze względu na wielkość, sieci dzielimy na: lokalne, miejskie, rozległe oraz globalne (Internet).
- Ze względu na model funkcjonowania rozróżniamy dwa typy sieci: „każdy z każdym" i „klient-serwer".
- Komputery mogą być połączone w różnych układach, np. szyny, gwiazdy, pierścienia. Najczęściej stosuje się układ gwiazdy.
- Karta sieciowa, protokół sieciowy, oprogramowanie klienta sieci oraz oprogramowanie realizujące usługi, z których korzystamy – to niezbędne składniki konfiguracji komputera pracującego w sieci.
- Sieć bezprzewodową zawsze należy zabezpieczyć przed nieupoważnionym dostępem.
- Reguły komunikowania się między komputerami w sieci określają protokoły sieciowe. Najbardziej rozpowszechnionym protokołem jest protokół TCP/IP.

Pytania i polecenia

1. Uzasadnij celowość łączenia w sieć komputerów w szkolnej pracowni.
2. Omów podział sieci ze względu na wielkość.
3. Czym różni się model sieci „każdy z każdym" od modelu „klient-serwer"?
4. W co powinien być wyposażony komputer pracujący w sieci?
5. Jaką funkcję pełnią protokoły sieciowe?
6. W jaki sposób komputery są identyfikowane w sieci?
7. Co jest niezbędne do budowy sieci domowej?
8. Na czym polega logowanie?
9. Jak udostępnić dysk swojego komputera innemu użytkownikowi sieci?
10. Na czym polega mapowanie zasobów?

Zadania

1. Udostępnij koleżance lub koledze, którzy pracują w tej samej sieci, twój zbiór utworów muzycznych (folder *Moja muzyka*). Zwróć uwagę na zmianę wyglądu ikony udostępnionego folderu.
2. Dokonaj mapowania udostępnionego przez koleżankę (kolegę) folderu *Moje obrazy* (ćwiczenie 6.). Następnie sprawdź, jak zmieniła się struktura zasobów twojego komputera, np. klikając ikonę **Eksplorator plików** (**Eksplorator Windows**).
3. Planujecie zorganizować w domu małą sieć lokalną. Przedstaw listę niezbędnych zakupów oraz czynności, jakie trzeba wykonać. Chcecie korzystać nie tylko ze wspólnych zasobów dyskowych, ale i z jednej drukarki.
4. Jeżeli masz w domu sieć bezprzewodową, sprawdź, czy jest właściwie zabezpieczona.

Dla zainteresowanych

5. Narysuj schemat sieci komputerowej działającej w twojej szkole. Możesz skorzystać ze specjalnych programów do rysowania schematów sieci.
6. Korzystając z **Pomocy** systemu Windows lub innych źródeł, znajdź więcej informacji na temat tworzenia **Grupy domowej** w systemie Windows 10, a zwłaszcza funkcjonalności takiego połączenia.

Przeczytaj, jeśli chcesz wiedzieć więcej...

Poszczególne fragmenty adresu IP określają albo numer sieci, albo numer konkretnego komputera (hosta) w ramach tej sieci. Do rozróżnienia używa się tzw. **maski podsieci**. Ustawia się ją w oknie przedstawionym na rysunkach 9a lub 9b. Maska odróżnia sieć wewnętrzną od sieci zewnętrznej. Komputery pracujące w tym samym fragmencie sieci mają taką samą maskę podsieci. Maska jest liczbą 32-bitową, ale składającą się z samych jedynek w części oznaczającej adres sieci i samych zer w części oznaczającej adres hosta, np. maska:

	Maska adresu sieci			Hosty
Zapis dziesiętny	255	255	255	000
Zapis binarny	11111111	11111111	11111111	00000000

– oznacza, że w danym segmencie sieci może być maksymalnie 256 adresów.

Przykład:

Adres IP	192	168	10	13
Maska podsieci	255	255	255	0

Adres IP 192.168.10.13 w podsieci o masce 255.255.255.0 oznacza komputer nr 13 w sieci o adresie 192.168.10.0.

Podstawowe narzędzia do diagnostyki sieci to polecenia **ping** i **tracert**.

Polecenie **ping** *adres_komputera* sprawdza połączenie komputera lokalnego z komputerem zdalnym.

W **Wierszu polecenia** wpisujemy *ping* i adres IP lub adres domenowy komputera zdalnego.

Polecenie **tracert** *adres_komputera* pokazuje drogę, jaką przebywa pakiet, by dotrzeć do zdalnego komputera.

W **Wierszu polecenia** wpisujemy *tracert* i adres IP lub adres domenowy komputera zdalnego.

Wiele wskazuje na to, że kolejnym etapem rozwoju Internetu będzie **Internet rzeczy** (ang. *Internet of Things* – w skrócie IoT). W Internecie rzeczy dane przetwarzają i wymieniają między sobą urządzenia: elektroniki domowej (np. smartfony, asystenci), infrastruktury miejskiej (np. inteligentne rozkłady jazdy), samochody, urządzenia noszone jako część odzieży itp.

Temat A2. Bezpieczeństwo i ochrona danych w komputerach i sieciach komputerowych

1. Wprowadzenie
2. Ochrona danych przed nieupoważnionym dostępem
   - 2.1. Kontrola dostępu do danych
   - 2.2. Nieupoważniony dostęp do danych w wyniku nieświadomych działań użytkownika
   - 2.3. Nieupoważniony dostęp do danych spowodowany działaniem innych osób
   - 2.4. Odmiany złośliwego oprogramowania
   - 2.5. Oprogramowanie zabezpieczające komputer
   - 2.6. Szyfrowanie danych
3. Ochrona przed utratą danych
   - 3.1. Odzyskiwanie przypadkowo usuniętych danych
   - 3.2. Odzyskiwanie danych w przypadku awarii komputera lub systemu operacyjnego
   - 3.3. Odzyskiwanie danych w przypadku ich utraty spowodowanej czynnikami zewnętrznymi

Warto powtórzyć

1. Jakie formy zabezpieczeń stosuje się w e-bankach?
2. Jakie są zasady bezpiecznego korzystania z konta bankowego?
3. Jak sprawdzić informacje dotyczące szyfrowanego połączenia z daną witryną internetową?

1. Wprowadzenie

Coraz więcej informacji istnieje wyłącznie w formie cyfrowej. Pieniądze na koncie, cyfrowe zdjęcia, cyfrowe pliki muzyczne czy cyfrowe książki nie posiadają formy fizycznej – istnieją jedynie w postaci zapisu w systemach komputerowych. Usunięcie zapisu cyfrowego jest o wiele prostsze niż zniszczenie materialnego przedmiotu, dlatego ochrona cyfrowych informacji i danych jest bardzo istotna.

Możemy wyróżnić dwie kategorie ochrony danych:

- ochrona danych przed nieupoważnionym dostępem,
- ochrona przed utratą danych.

2. Ochrona danych przed nieupoważnionym dostępem

2.1. Kontrola dostępu do danych

W systemach komputerowych podstawowym mechanizmem kontroli dostępu do danych są **konta użytkowników**. Do każdego konta przypisane są **prawa dostępu**, określające, jakiego rodzaju operacje może wykonywać użytkownik i do jakich danych ma dostęp. Weryfikowanie tożsamości użytkownika odbywa się za pomocą **uwierzytelniania**.

Istnieją trzy podstawowe metody uwierzytelniania:

1. „Coś, co wiem" – do uwierzytelniania wykorzystywane są informacje, które powinien posiadać wyłącznie właściciel konta, np. składający się z cyfr PIN (ang. *Personal Identification Number*) lub alfanumeryczne hasło.
2. „Coś, co mam" – do uwierzytelniania wykorzystywane są przedmioty będące w posiadaniu właściciela konta, np. klucz do zamka, klucz elektroniczny, karta dostępu, token (generator kodów).
3. „Coś, co mnie charakteryzuje" – do uwierzytelniania wykorzystywane są cechy osobowe (biometryczne), np. odcisk palca, wizerunek twarzy, wzór siatkówki, tembr głosu itp.



Pamiętaj!

Numerów PIN i haseł nie powinniśmy zapisywać ani udostępniać innym osobom! Jeżeli musimy zapisać tego rodzaju dane, należy przechowywać je w miejscu, gdzie nie mogą zostać odczytane przez niepowołane osoby.

Pamiętaj, aby unikać szczególnie niebezpiecznych zachowań takich, jak:

- zapisywanie kodu PIN na karcie płatniczej;
- „dzielenie się" kontami (kilka osób korzysta z jednego konta);
- zapisywanie hasła do konta na karteczce przyklejonej do monitora;
- korzystanie z łatwych do odgadnięcia haseł, np. imienia osoby, ciągu alfabetycznego liter, ciągu znaków odpowiadających kolejnym klawiszom;
- wykorzystywanie tego samego hasła do logowania w różnych serwisach.

Czy wiesz, że...

najczęściej wykorzystywane hasła (wg danych z roku 2018) to: *123456*, *password*, *123456789*, *12345678*, *12345*, *111111*, *1234567*, *sunshine*, *qwerty*, *iloveyou*, *princess*, *admin*, *welcome*, *666666*, *abc123*? Tych haseł nigdy nie używaj!

Ćwiczenie 1. Obliczamy, ile kodów PIN można utworzyć za pomocą czterech cyfr

Odpowiedz na pytanie: *Ile można utworzyć czterocyfrowych kodów PIN?*
Oblicz, ile czasu zabrałoby sprawdzenie wszystkich możliwych kombinacji przy założeniu, że co sekundę sprawdzamy jeden kod.

Ćwiczenie 2. Obliczamy, ile haseł można utworzyć z danej liczby znaków

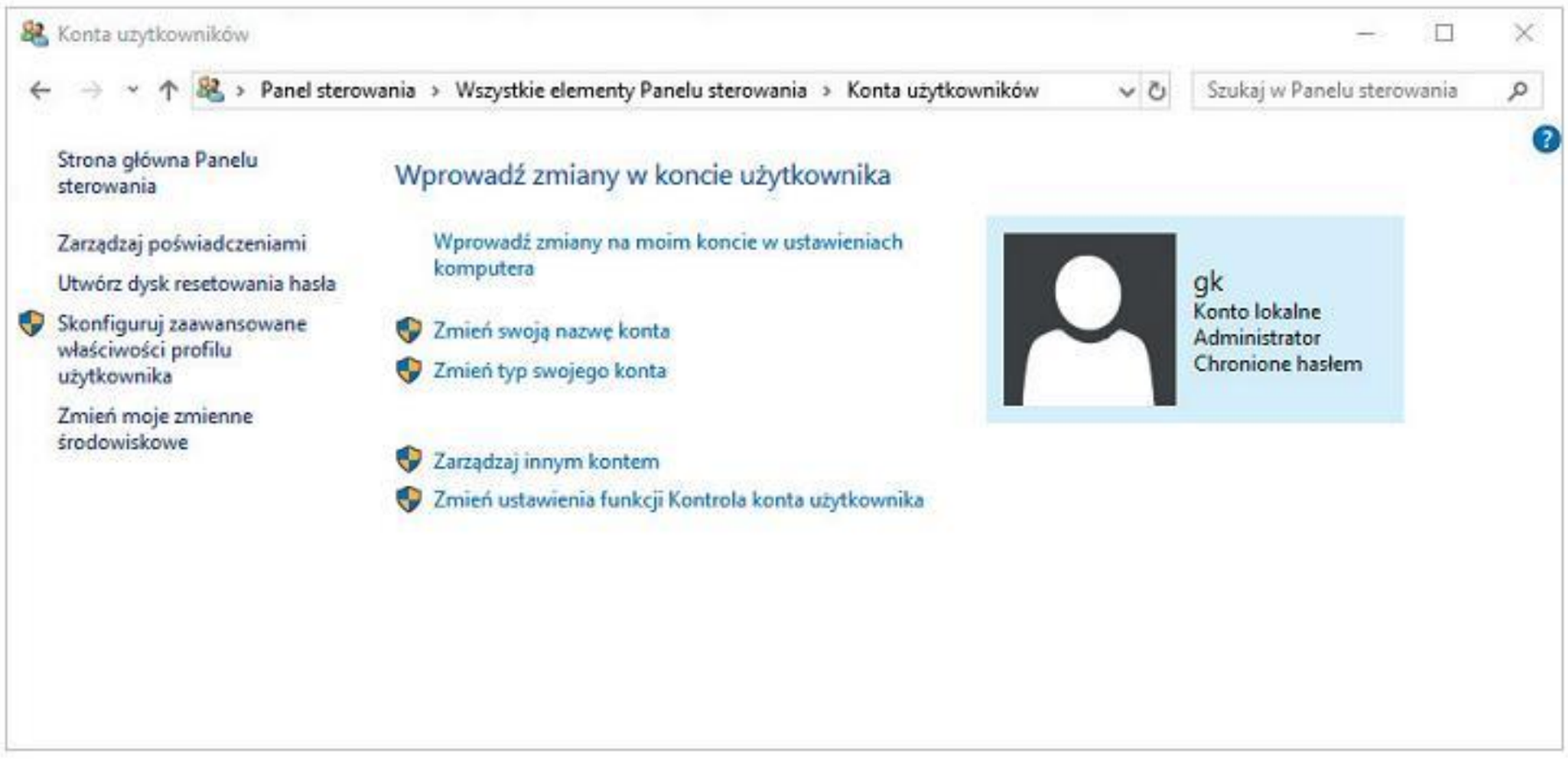
1. Liczba haseł możliwych do utworzenia za pomocą m znaków to m^n, gdzie m – liczba możliwych do wykorzystania znaków (np. liter w alfabecie), n – liczba znaków w haśle. Oblicz, ile można utworzyć haseł składających się z:
   a. sześciu liter (małe i wielkie litery są traktowane tak samo);
   b. ośmiu liter lub cyfr (małe i wielkie litery są traktowane tak samo);
   c. dziesięciu liter lub cyfr (małe i wielkie litery są rozróżnialne).
2. Oblicz, ile czasu zabrałoby sprawdzenie wszystkich możliwych haseł przy założeniu, że w ciągu jednej sekundy sprawdzamy tysiąc, milion i miliard haseł.

Uwaga: Znaki w haśle mogą się powtarzać.

W przypadku, gdy do uwierzytelnia wykorzystywane są dwie metody, mówimy o uwierzytelnianiu dwuetapowym (ang. *Two-factor Authentication*, w skrócie 2FA). Taki system wykorzystuje się powszechnie w bankach internetowych, gdzie do zalogowania do banku wystarczy zazwyczaj hasło, ale zlecenie operacji finansowej wymaga dodatkowego potwierdzenia (kodem odczytanym z karty kodów, z tokena lub przesłanym przez bank w wiadomości SMS).

Konta użytkowników są powszechnie stosowane w sieciach komputerowych. W systemie Windows można je tworzyć, modyfikować i usuwać za pomocą funkcji **Konta użytkowników**, dostępnej w **Panelu sterowania** (rys. 1.).

Przez nieupoważniony dostęp do danych należy rozumieć sytuacje, gdy dostęp do istotnych, prywatnych danych i informacji uzyskują osoby trzecie, którym na to nie zezwoliliśmy.



Rys. 1. Konta użytkowników *(Windows 10)*

Sytuacja nieupoważnionego dostępu do danych może być wynikiem:

- nieświadomych działań użytkownika,
- celowych działań osób trzecich.

2.2. Nieupoważniony dostęp do danych w wyniku nieświadomych działań użytkownika

Nieupoważniony dostęp do danych może być skutkiem braku świadomości użytkownika, który nie zdaje sobie sprawy z konsekwencji swoich poczynań. Z sytuacją taką spotykamy się często np. w serwisach społecznościowych. Umieszczając tam dane (np. fotografie, komentarze, dane osobowe), zwykle zgadzamy się na ich udostępnianie innym członkom serwisu, zdefiniowanym przez odpowiednie ustawienia konta (np. tzw. „znajomym"). W zależności od serwisu zakres widoczności danych może być jednak większy, niż się spodziewamy – np. dostęp do danych mogą mieć także „znajomi znajomych". Niektóre dane mogą być również publiczne i podlegać np. wyszukiwaniu w wyszukiwarkach internetowych.

Sekcja ustawień związanych z prywatnością danych w serwisach społecznościowych jest zwykle dość rozbudowana, więc stosunkowo łatwo jest się pomylić i np. udostępnić prywatne zdjęcia wszystkim użytkownikom serwisu.

O zagrożeniach prywatności piszemy więcej w temacie D2.



Ćwiczenie 3. Sprawdzamy, jakie dane udostępniamy innym użytkownikom serwisu

Jeżeli korzystasz z jakiegoś serwisu społecznościowego, sprawdź w ustawieniach swojego konta, jakie dane na twój temat są udostępniane innym użytkownikom.

Umieszczając w Internecie, w szczególności w serwisach społecznościowych, informacje o sobie (np. zdjęcia), należy pamiętać, że raz opublikowane informacje mogą być bez problemu kopiowane, w związku z czym ich usunięcie z sieci może być bardzo trudne lub wręcz niewykonalne. Warto też mieć świadomość, że nasze dane mogą zostać wykorzystane przez osoby trzecie, np. w celach przestępczych.



Nie należy podawać swoich pełnych danych, o ile nie jest to konieczne.



Zgodnie z polskim prawem, każdej osobie przysługuje prawo wglądu do zgromadzonych o niej danych oraz ich poprawiania. Można także domagać się zaprzestania przetwarzania swoich danych osobowych (co powinno skutkować ich usunięciem).

Częstą sytuacją skutkującą uzyskaniem dostępu do poufnych lub podlegających ochronie danych przez osoby nieupoważnione jest zgubienie wymiennych nośników pamięci, zawierających tego typu dane. Aby zapobiec takiej sytuacji, ważne dane, zwłaszcza dane osobowe, finansowe, będące tajemnicą, powinny zostać **zaszyfrowane.**

2.3. Nieupoważniony dostęp do danych spowodowany działaniem innych osób

Nieupoważniony dostęp do danych może mieć miejsce, gdy zostaną one skopiowane (utracone) wskutek celowych poczynań innych osób. Niektóre dane (np. informacje o kontrahentach firmy, o wynikach finansowych, zbiory danych osobowych, kody dostępu, numery kart kredytowych, prywatne zdjęcia) przedstawiają dla przestępców dużą wartość. Dane takie powinny być jak najlepiej chronione. Aby jednak móc skutecznie je chronić, należy znać metody, jakimi posługują się przestępcy.

Phishing

Jest to sposób zdobywania poufnych informacji oparty na inżynierii społecznej – poprzez podszywanie się. Użytkownik jest zazwyczaj zachęcany do odwiedzenia strony internetowej (np. serwisu społecznościowego, poczty elektronicznej, e-banku), wyglądającej jak prawdziwa, lecz kontrolowanej przez przestępców. Wpisując na takiej stronie swój identyfikator i hasło, użytkownik przekazuje te dane przestępcom. Najpopularniejszą metodą zachęcenia użytkownika do odwiedzenia spreparowanej strony jest wysłanie listu elektronicznego (z informacją np. o wygranej w konkursie, o wygaśnięciu ważności konta, o konieczności potwierdzenia hasła itp.) z hiperłączem prowadzącym do fałszywej strony.



Aby ustrzec się przed phishingiem:

- nie należy klikać na linki znajdujące się w otrzymywanych wiadomościach; w razie potrzeby należy wejść na stronę, wpisując jej adres w przeglądarce lub wybierając dodaną przez siebie zakładkę;
- w przypadku bezpiecznych połączeń należy zawsze sprawdzać certyfikat strony i nie kontynuować procesu logowania, jeżeli przeglądarka informuje, że certyfikat jest niepoprawny.



Ćwiczenie 4. Szukamy grafiki przedstawiającej phishing

Poszukaj w Internecie grafiki (rysunków, zrzutów ekranowych) ilustrującej przykłady phishingu.

Exploit

Technika ta polega na wykorzystywaniu (ang. *exploitation*) błędów w systemie operacyjnym lub programach do uruchomienia nieautoryzowanego, złośliwego kodu (ang. *malicious software*, w skrócie **malware**).

Szczególnie narażone na tego rodzaju atak są: system operacyjny, przeglądarki internetowe, wtyczka do przeglądarek Adobe Flash Player i czytnik PDF-ów Adobe Reader. Typowy scenariusz zakłada wysłanie ofierze odpowiednio spreparowanego pliku (np. w formacie PDF, MP3, WMV, JPEG), który po otwarciu powoduje uaktywnienie złośliwego kodu; niekiedy wystarczy samo wejście na odpowiednio spreparowaną stronę internetową. Złośliwe oprogramowanie jest często rozprowadzane z pirackimi kopiami programów.

Aby zabezpieczyć się przed zagrożeniem typu exploit, należy regularnie aktualizować wszystkie zainstalowane w komputerze programy.

Programy komputerowe posiadają często funkcję automatycznej aktualizacji lub informują o pojawieniu się nowej wersji oprogramowania.

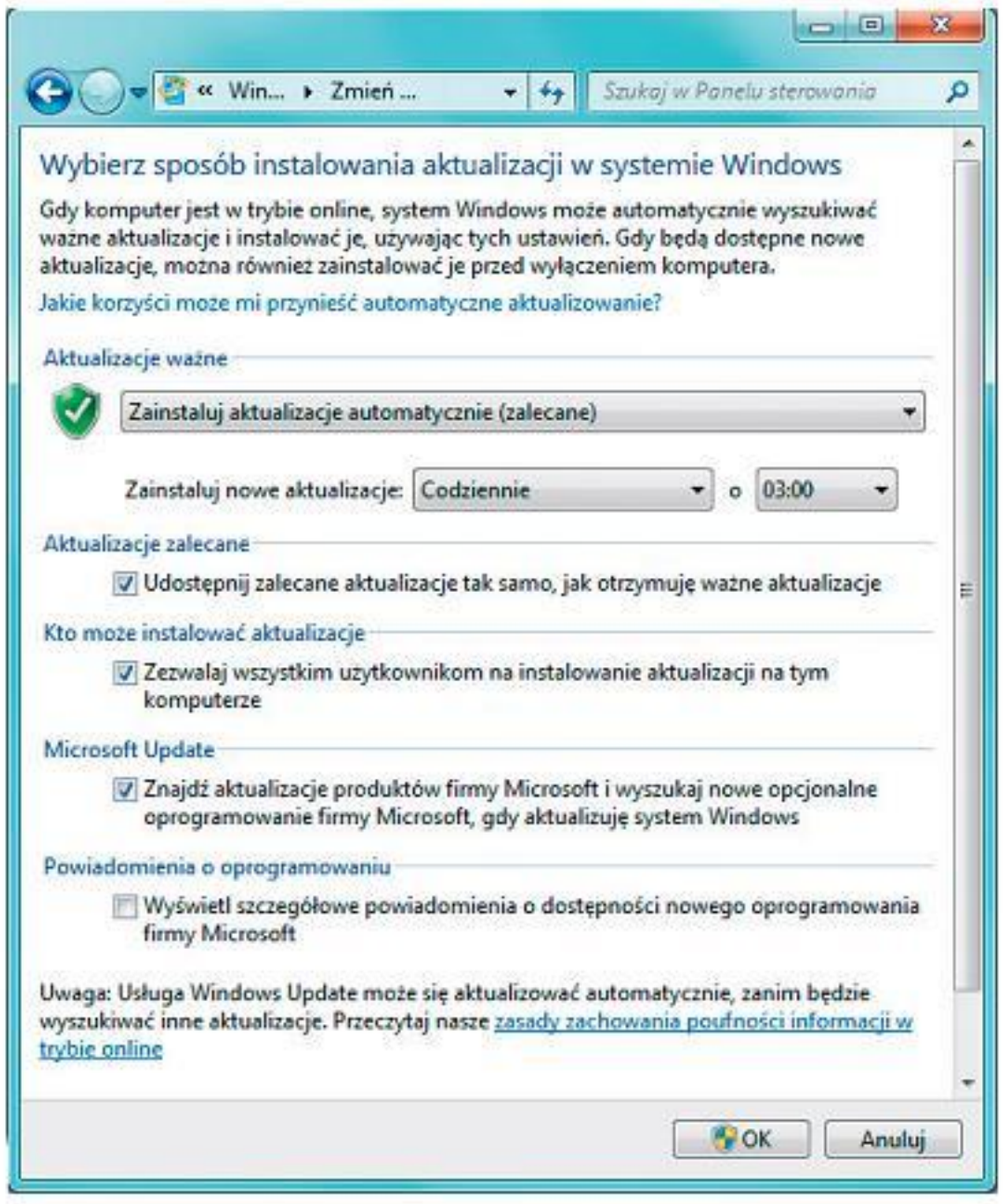
Przykład 1. Włączanie automatycznej aktualizacji systemu Windows

Windows 7, 8, 8.1

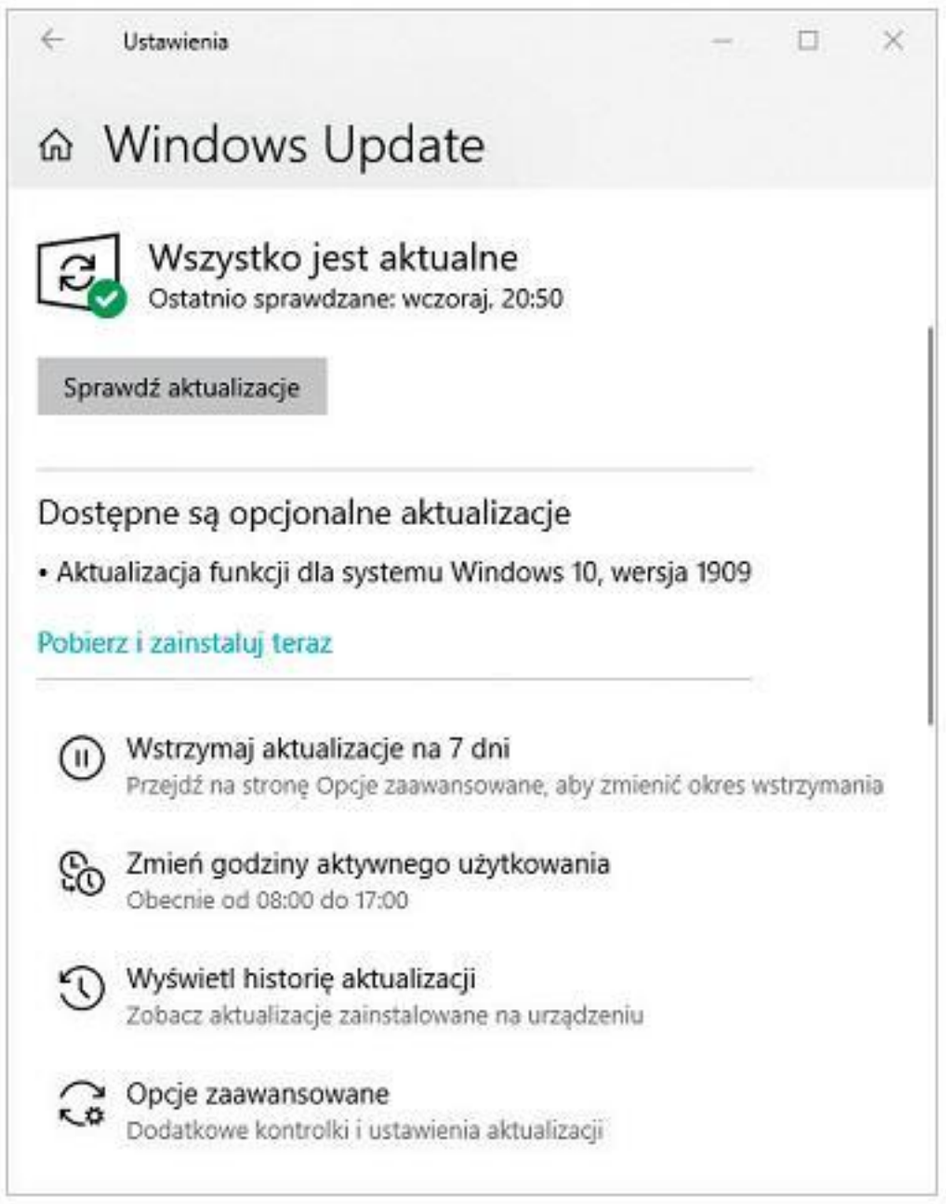
W oknie **Panel sterowania** należy wybrać opcję **System i zabezpieczenia**, a następnie kliknąć link **Włącz lub wyłącz automatyczne aktualizowanie**. Otworzy się okno podobne do pokazanego na rys. 2a.

Windows 10

W aplikacji **Ustawienia** należy kliknąć link **Aktualizacja i zabezpieczenia**, a następnie **– Windows Update.** Otworzy się okno podobne do pokazanego na rys. 2b. Klikając np. link **Zmień godziny aktywnego użytkowania**, możemy określić godziny aktywnej pracy na komputerze. Aktualizowanie systemu zostanie w tym czasie ograniczone i nie będzie spowalniało pracy komputera.



Rys. 2a. *Zarządzanie funkcją automatycznych aktualizacji systemu Windows 7*



Rys. 2b. *Zarządzanie funkcją automatycznych aktualizacji systemu Windows 10*

Ćwiczenie 5. Sprawdzamy funkcje automatycznej aktualizacji systemu operacyjnego

Sprawdź, czy w komputerze, z którego korzystasz, włączona jest funkcja automatycznej aktualizacji systemu operacyjnego.

Ćwiczenie 6. Sprawdzamy wersję przeglądarki internetowej

Sprawdź, jaki jest numer wersji przeglądarki internetowej, z której korzystasz. Następnie, korzystając z Internetu, sprawdź, czy jest to najnowsza wersja.

Uwaga: W szkołach często zainstalowane są starsze wersje programów, z uwagi na ich mniejsze wymagania sprzętowe.

2.4. Odmiany złośliwego oprogramowania

Keylogger

Keylogger (z ang. *key* – klawisz, *log* – zapisać) to specjalny program, który funkcjonuje w systemie w niezauważalny sposób (np. nie widać go na pasku zadań). Jego zadaniem jest przechwytywanie i gromadzenie informacji o wszystkich naciśniętych klawiszach (a klawiatury używa się m.in. do wprowadzania haseł), a następnie przesyłanie zebranych danych do osoby kontrolującej keyloggera, która może ich użyć do uzyskania dostępu do kont użytkownika w serwisach internetowych – w tym do kont bankowych.

Bardziej rozbudowane oprogramowanie tego typu zawiera dodatkowe funkcje, np. może w regularnych odstępach czasu rejestrować obrazy z podłączonej do komputera kamery internetowej lub wykonywać zrzuty ekranu.

Istnieją również keyloggery sprzętowe – przejściówki, które wpina się pomiędzy port PS/2 lub USB komputera a przewód klawiatury komputera.

Spyware

Spyware (z ang. *spy* – szpiegować) to określenie odnoszące się do programów służących do wykradania z komputera różnego rodzaju danych (plików, haseł dostępu, historii odwiedzanych stron).

Botnet

Złośliwe oprogramowanie instalowane na wielu komputerach (liczba komputerów w największych botnetach to kilka milionów) i kontrolowane przez jedną osobę (grupę przestępczą). Może zostać wykorzystane np. do wysyłania spamu, przeprowadzania ataków na serwery, operacji prania brudnych pieniędzy itp. Komputery wchodzące w skład botnetu nazywa się czasami **komputerami zombie**.

Ransomware

Złośliwe oprogramowanie szyfrujące wszystkie pliki znajdujące się na dysku. Do ich odszyfrowania niezbędny jest klucz, którym dysponują wyłącznie przestępcy. Klucz ten udostępniają po zapłaceniu „okupu" (ang. *ransom*).

Trojan

Nazwa tego rodzaju programów nawiązuje do mitologicznego konia trojańskiego. Mianem tym określa się wszelkiego rodzaju programy, które pod pozorem wykonywania pożytecznych czynności w istocie wykradają dane lub wykonują inne niepożądane operacje. Przykładem tego rodzaju programów są fałszywe programy antywirusowe.

Wirusy komputerowe

Wirusy to programy, których głównym celem jest powielanie się, poprzez zarażanie innych programów, rzadziej – dokumentów (tzw. makrowirusy). Dodatkowym celem działania wirusa może być uszkadzanie oprogramowania zainstalowanego w komputerze lub systemu plików, a także wykonywanie innych niepożądanych operacji.

Robaki komputerowe

Robaki komputerowe są podobne do wirusów, jednak rozprzestrzeniają się nie poprzez doklejanie kodu do innych programów, lecz poprzez sieci komputerowe (pocztę elektroniczną, komunikatory, serwisy społecznościowe).

2.5. Oprogramowanie zabezpieczające komputer



Podstawową metodą ochrony przed złośliwymi programami jest stosowanie oprogramowania antywirusowego.

Na rynku dostępne są zarówno programy komercyjne, jak i darmowe, różniące się liczbą oferowanych funkcji i skutecznością (liczbą wykrywanych zagrożeń). Ważne jest, aby program antywirusowy w komputerze był nie tylko aktywny, lecz również regularnie aktualizowany. Większość programów antywirusowych posiada funkcję automatycznej aktualizacji swoich baz danych, służących do wykrywania złośliwego oprogramowania.

Bezpieczeństwo komputera wzrośnie, jeżeli będziemy regularnie (np. raz w tygodniu) uruchamiali funkcję skanowania systemu programem antywirusowym w celu wykrycia złośliwego oprogramowania. Jeżeli podejrzewamy, że nasz komputer został zarażony takim oprogramowaniem (typowe objawy: spowolniona praca, niedziałające programy, pojawiające się nagle komunikaty o błędach, wolne pobieranie danych z Internetu), należy natychmiast przeprowadzić skanowanie. Na rynku dostępne są również specjalne programy wykrywające i usuwające najczęściej spotykane złośliwe oprogramowanie.



Ćwiczenie 7. Sprawdzamy, jaki program antywirusowy zainstalowano na naszym komputerze

Sprawdź, czy na komputerze, z którego korzystasz, dostępny jest program antywirusowy. Jeżeli tak, sprawdź, czy jego bazy danych są aktualne.

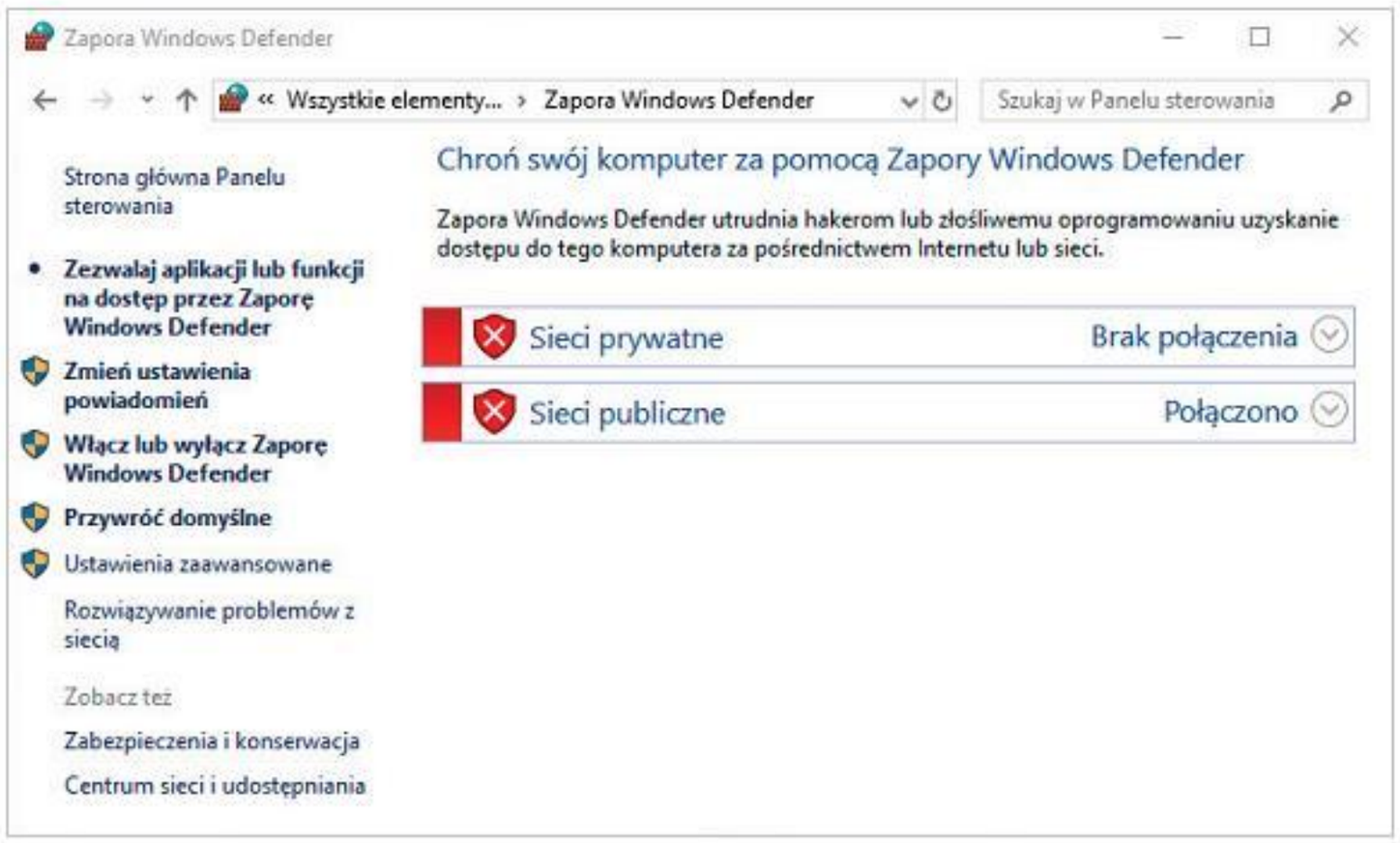
Inną skuteczną formą zabezpieczenia systemu komputerowego jest firewall. Firewall jest elementem systemów Windows (od wersji XP) oraz Linux. Firewall wbudowany w system Windows ma stosunkowo ograniczone możliwości, dlatego wiele programów antywirusowych oferuje własne rozwiązania, o lepszym działaniu.

Firewall jest często elementem oprogramowania routerów – w takim wypadku również powinien być włączony i odpowiednio skonfigurowany.

Przykład 2. Włączanie firewalla

Aby włączyć i skonfigurować firewall (zwany **Zaporą systemu Windows**), należy w oknie **Panel sterowania**:

- w Windows 7, 8, 8.1, kliknąć link **System i Zabezpieczenia**,
- w Windows 10 – kliknąć link **Zapora Windows Defender** (rys. 3.).



Rys. 3. *Okno konfiguracji* ***Zapory Windows Defender*** *(Windows 10)*

Ćwiczenie 8. Sprawdzamy, czy zainstalowano firewall

Sprawdź, czy w komputerze, z którego korzystasz, zainstalowano i włączono firewall.

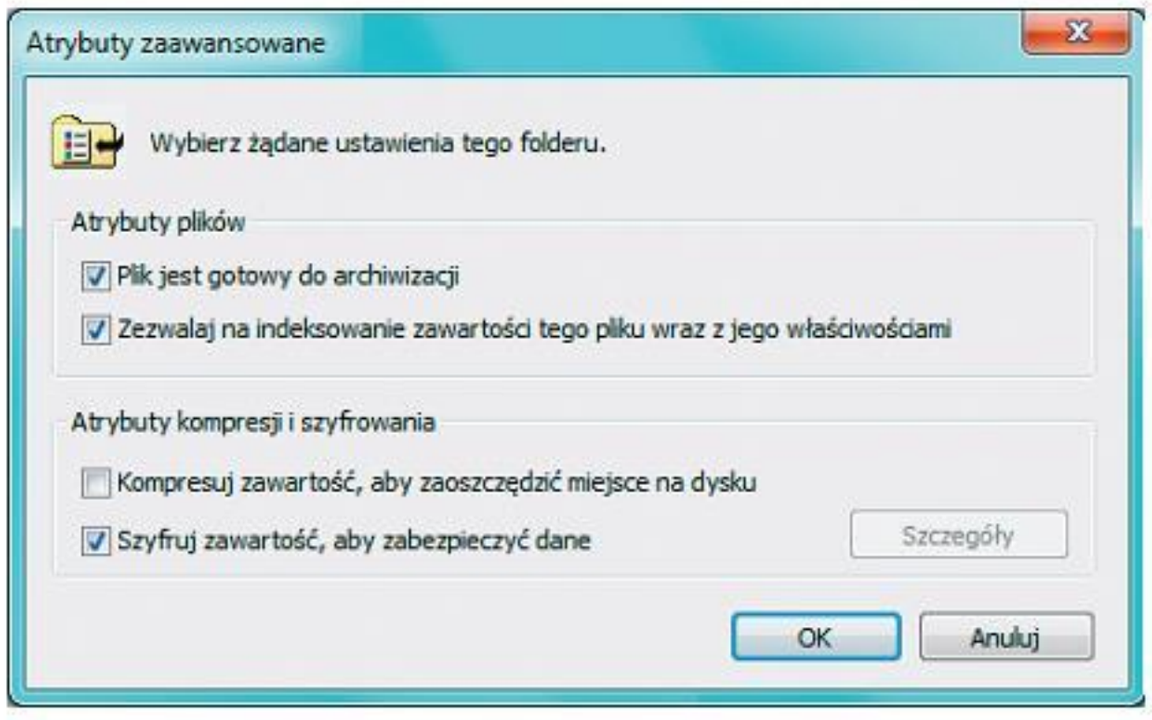
2.6. Szyfrowanie danych

Najwyższy poziom ochrony danych zapewnia **szyfrowanie**. Szyfrowanie to proces matematyczny, w wyniku którego informacja zostaje zakodowana w postaci uniemożliwiającej jej odczytanie bez znajomości **klucza**. Na rynku dostępnych jest wiele rozwiązań, zarówno komercyjnych, jak i darmowych, pozwalających na szyfrowanie danych.

Niektóre aplikacje posiadają funkcję szyfrowania zapisywanych dokumentów. Dostępna jest ona np. w programach z rodziny Microsoft Office (podczas zapisywania pliku można podać hasło ochrony, co powoduje zaszyfrowanie pliku).

W systemie Windows możliwe jest zaszyfrowanie zawartości pliku. Do szyfrowania wykorzystywane jest hasło używane do logowania użytkownika do systemu Windows, nie trzeba go więc dodatkowo podawać.

Aby uaktywnić możliwość szyfrowania plików, należy z menu kontekstowego pliku wybrać **Właściwości**, a następnie kliknąć przycisk **Zaawansowane**. W oknie **Atrybuty zaawansowane** można uaktywnić opcję szyfrowania zawartości pliku lub folderu (rys. 4.).

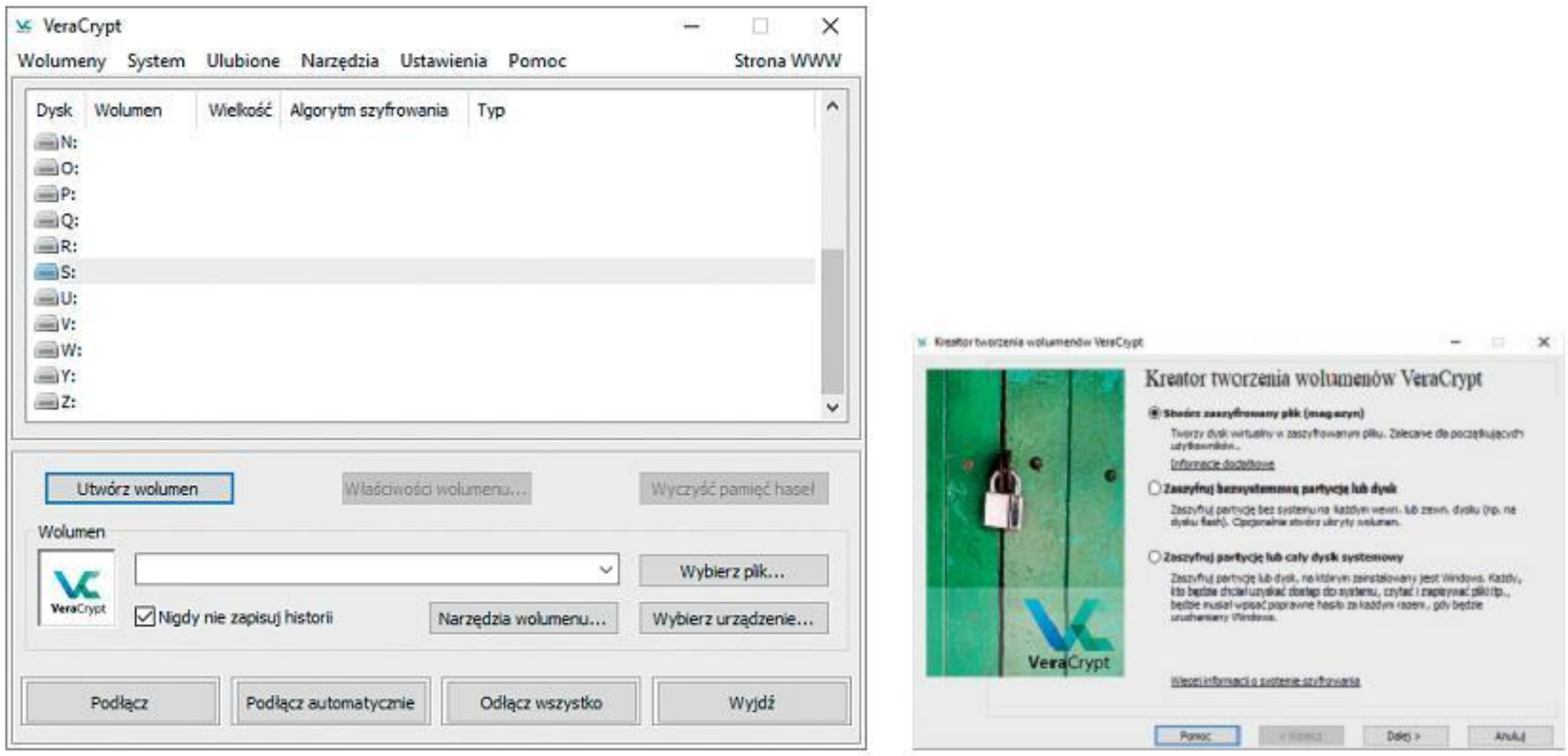


Rys. 4. *Okno, w którym można zaszyfrować zawartość pliku (Windows 7)*

Uwaga: Szyfrowanie podaną metodą działa tylko dla plików zapisanych na dyskach lokalnych komputera. Nazwy zaszyfrowanych plików zaznaczane są kolorem zielonym.

Niektóre odmiany systemów Windows Vista, Windows 7 i Windows 8 posiadają wbudowane narzędzie BitLocker, umożliwiające szyfrowanie całego dysku.

Darmowy program VeraCrypt, dostępny dla systemów Windows, MacOS i Linux, pozwala na szyfrowanie całych dysków (w tym wymiennych, np. typu pendrive), partycji dyskowych, a także umożliwia tworzenie w systemie dodatkowych woluminów (dysków logicznych), których zawartość jest szyfrowana (rys. 5.).



Rys. 5. *Okna programu VeraCrypt i* ***Kreatora tworzenia wolumenów***



Jedynym sposobem odczytania zawartości zaszyfrowanego pliku (dysku) jest podanie poprawnego hasła. Jeśli nie pamiętamy hasła, odczytanie danych nie będzie możliwe. Złamanie nowoczesnych algorytmów szyfrowania wymaga wielu lat pracy tysięcy komputerów.

3. Ochrona przed utratą danych

Utrata danych może być spowodowana kilkoma czynnikami:

- przypadkowym działaniem użytkownika (np. zaznaczenie folderu i omyłkowe naciśnięcie klawisza **Delete**, co spowoduje usunięcie folderu wraz z zawartością),
- awarią komputera lub systemu operacyjnego,
- działaniem złośliwego oprogramowania,
- czynnikami zewnętrznymi (np. kradzież sprzętu, powódź, pożar, katastrofa budowlana).

W zależności od czynnika powodującego utratę danych, inne będą metody ich zabezpieczenia oraz odzyskiwania.



Podstawą ochrony ważnych danych przed utratą jest regularne wykonywanie kopii bezpieczeństwa.

3.1. Odzyskiwanie przypadkowo usuniętych danych



Usunięcie pliku z dysku nie powoduje skasowania danych zapisanych w pliku – usuwane są jedynie informacje o pliku, a miejsce na dysku zajmowane przez plik jest oznaczane jako wolne.

Istnieje wiele programów (w tym darmowych) służących do odzyskiwania usuniętych plików. Programy te skanują dysk w poszukiwaniu usuniętych plików, a następnie próbują odzyskać zapisane w nich informacje. Warunkiem powodzenia takiej operacji jest przeprowadzenie jej możliwie szybko po usunięciu plików (dopóki nie nastąpiło nadpisanie miejsca na dysku nowymi plikami).

W ten sam sposób można odzyskać dane z zewnętrznych nośników danych, np. urządzeń pendrive lub kart pamięci aparatów cyfrowych.

W przypadku, gdy nie będziemy potrzebować nośnika danych i chcemy go np. sprzedać lub przekazać innej osobie, zapisane na nim informacje należy trwale usunąć (by kolejny właściciel nie mógł ich odzyskać). Służą do tego celu odpowiednie programy, w tym darmowe.

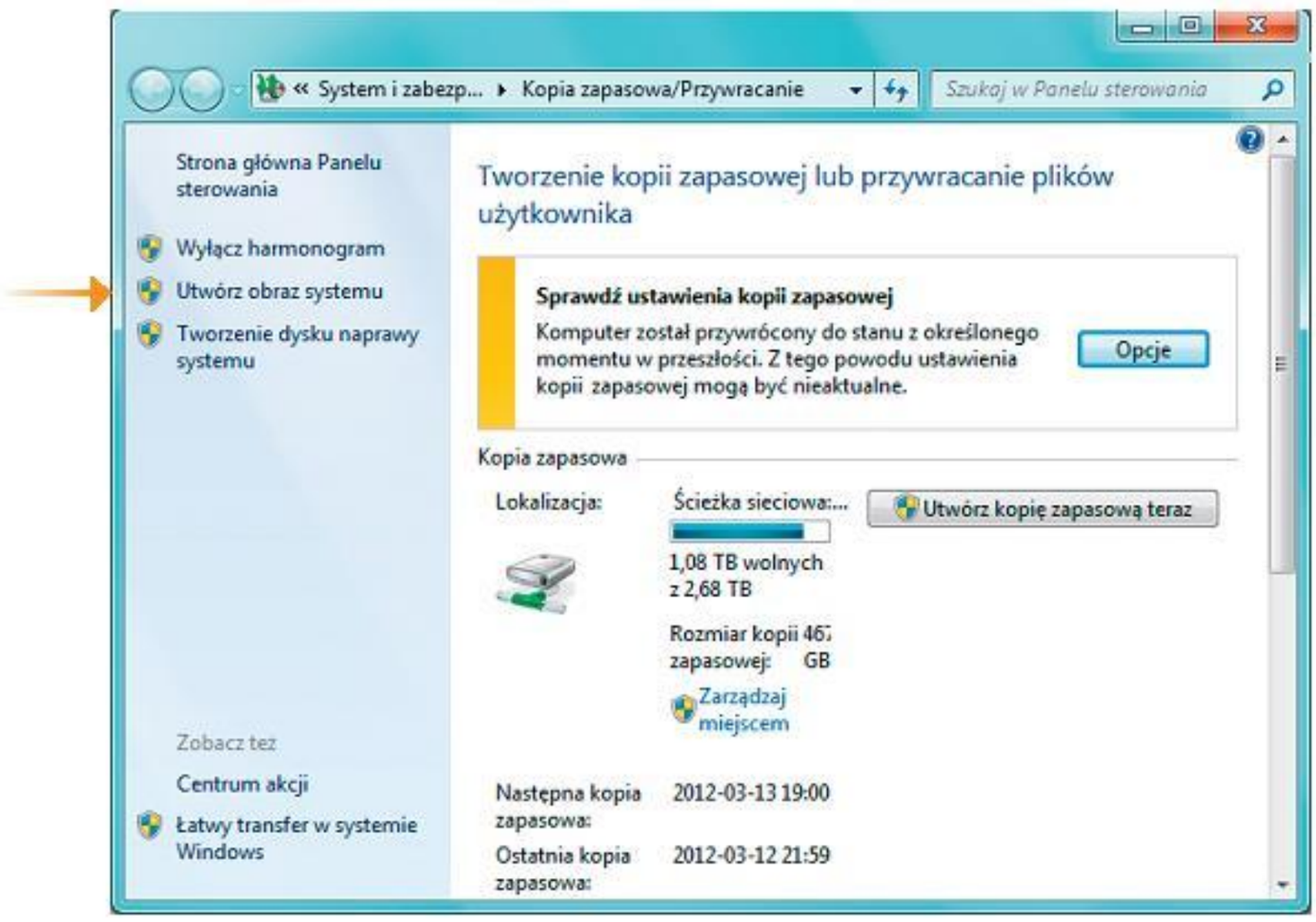
3.2. Odzyskiwanie danych w przypadku awarii komputera lub systemu operacyjnego

Podstawą procesu odzyskiwania danych w przypadku awarii komputera lub systemu operacyjnego jest posiadanie kopii bezpieczeństwa plików. Najprostszą metodą wykonania kopii bezpieczeństwa plików jest ich skopiowanie (np. za pomocą Eksploratora Windows) na inny, najlepiej zewnętrzny, nośnik danych. Nie da się jednak wykonać w ten sposób kopii systemu operacyjnego, ponieważ na system operacyjny składają się także pliki ukryte na dysku lub niedające się skopiować z powodu braku

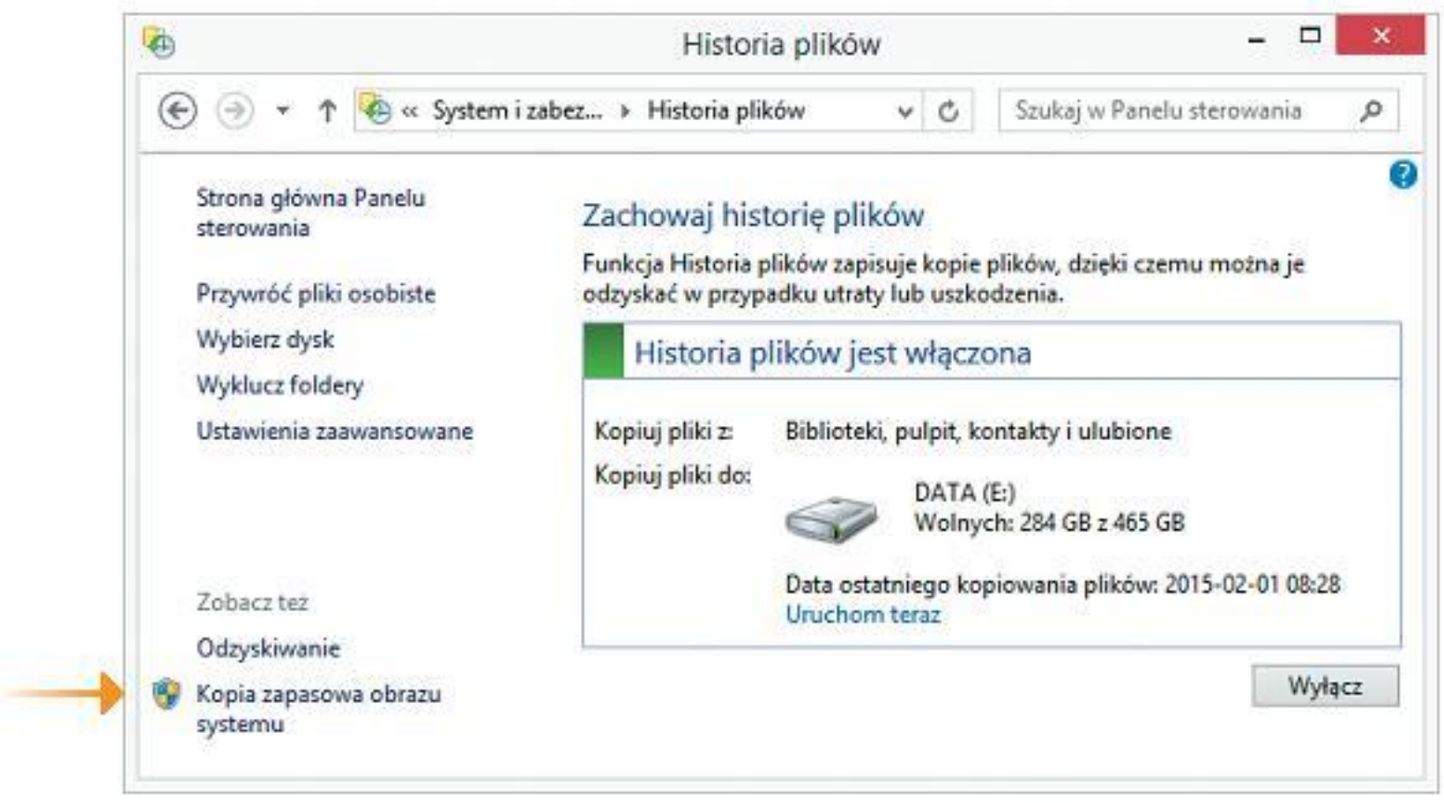
odpowiednich uprawnień. W takich przypadkach warto sięgnąć po specjalny program do wykonywania kopii bezpieczeństwa.

Prosty program tego rodzaju, o ograniczonych możliwościach, wbudowany jest w system Windows i nosi nazwę **Kopia zapasowa** (**Historia plików** w Windows 8, 8.1, 10) – rys. 6a i 6b.

Bardziej zaawansowane programy do tworzenia kopii bezpieczeństwa dostępne są odpłatnie; często są też dołączane do zewnętrznych dysków twardych (niektóre dyski posiadają nawet specjalny przycisk powodujący uruchomienie procesu tworzenia kopii bezpieczeństwa).



Rys. 6a. *Tworzenie kopii danych za pomocą programu* ***Kopia zapasowa*** *(Windows 7)*

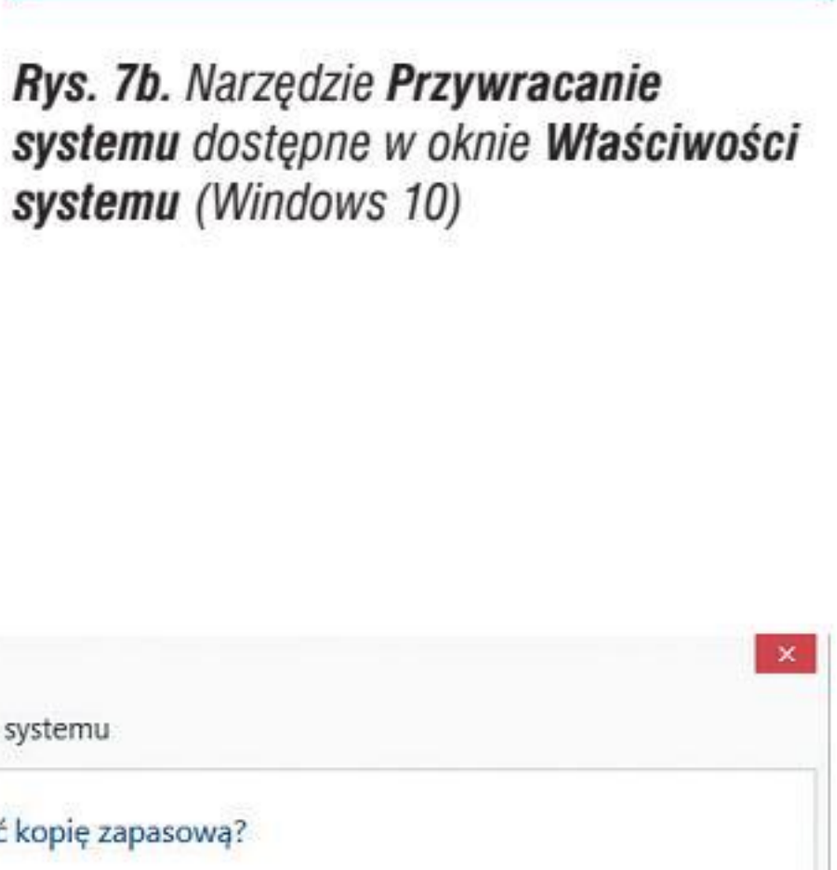


Rys. 6b. *Tworzenie kopii danych za pomocą programu* ***Historia plików*** *(Windows 8, 8.1, 10)*

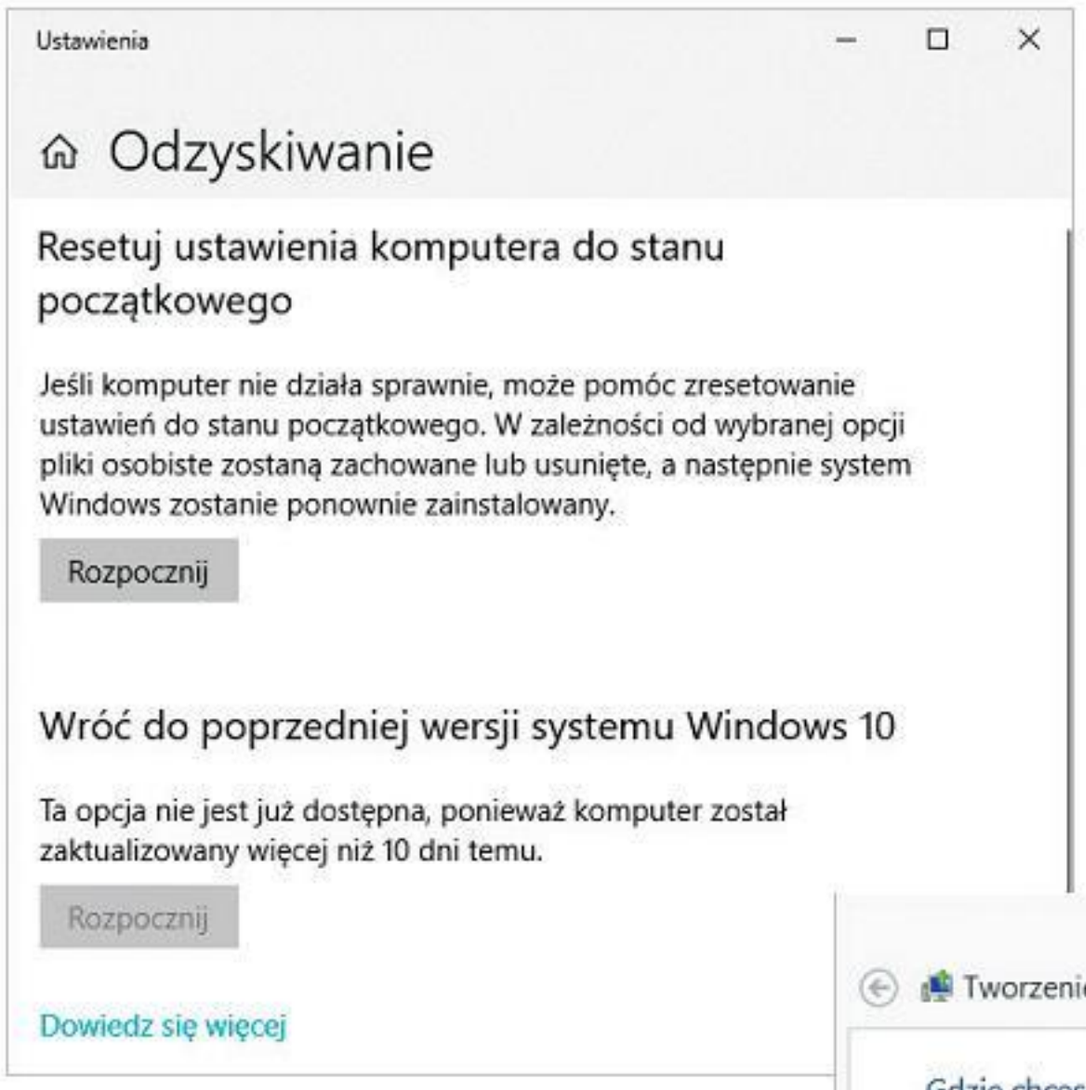
W przypadku drobnych awarii systemu operacyjnego Windows godną polecenia metodą naprawy jest korzystanie z narzędzia **Przywracanie systemu**. W Windows 7 można je znaleźć w menu **Programy (Wszystkie programy)/Akcesoria/Narzędzia systemowe**. Program ten umożliwia zapisanie aktualnego stanu systemu w postaci punktu kontrolnego lub przywrócenie systemu operacyjnego do stanu z określonego dnia (rys. 7a).



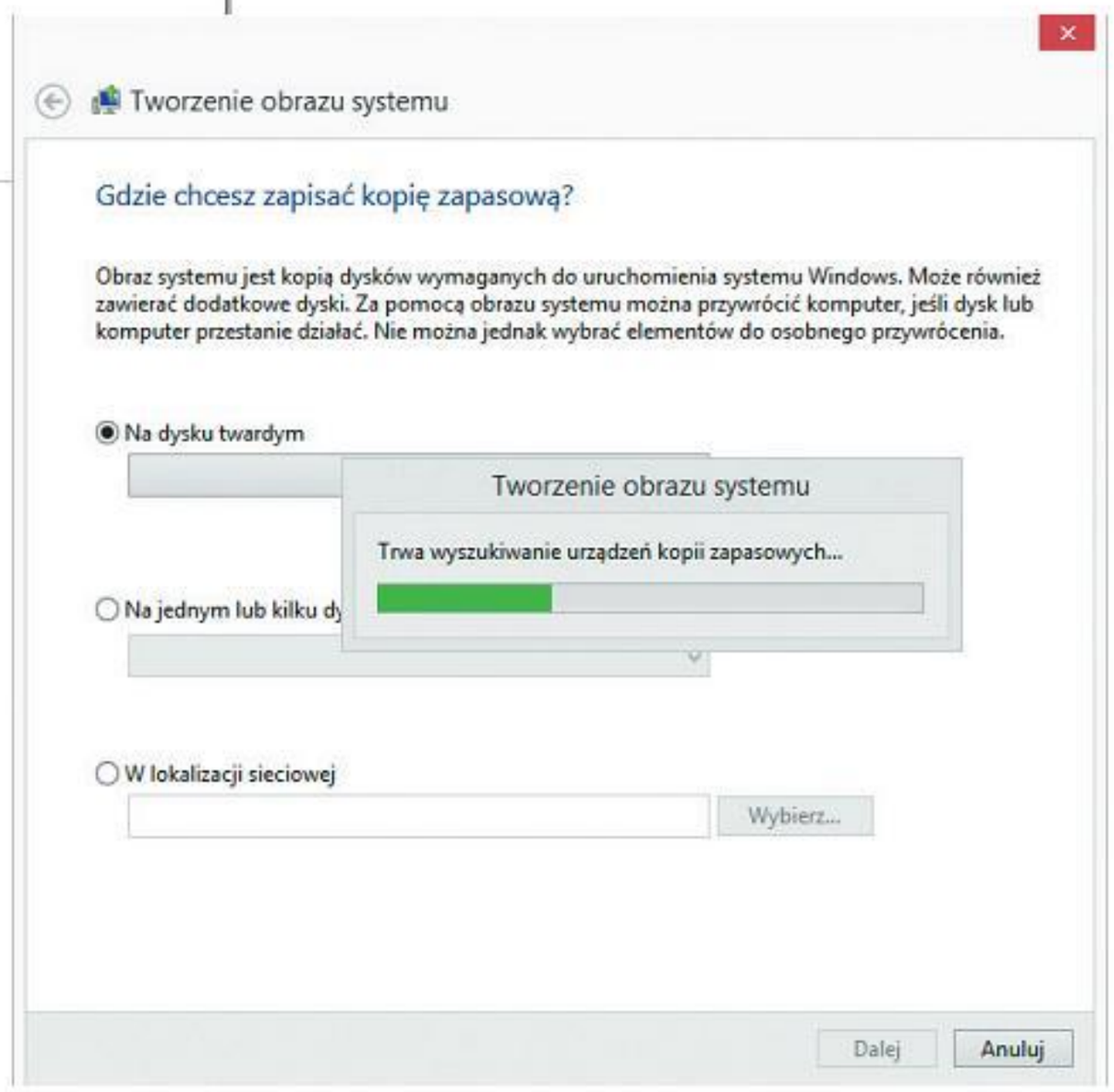
Rys. 7a. *Interfejs programu* ***Przywracanie systemu*** *(Windows 7)*



Rys. 7b. *Narzędzie* ***Przywracanie systemu*** *dostępne w oknie* ***Właściwości systemu*** *(Windows 10)*



Rys. 7c. *Okno resetowania ustawień komputera (Windows 10)*



Rys. 7d. Tworzenie obrazu systemu *(Windows 7, 8, 8.1)*

Aby w systemie Windows 10 wybrać narzędzie **Przywracanie systemu**, należy w oknie **Panel sterowania** kliknąć link **System**, następnie link **Zaawansowane ustawienia systemu**, w otwartym oknie **Właściwości** systemu wybrać zakładkę **Ochrona systemu**, a na koniec kliknąć przycisk **Przywracanie systemu** (rys. 7b).

W ustawieniach systemu Windows 10 (opcja **Odzyskiwanie**) możemy zresetować system, przywracając jego ustawienia domyślne – zostaną zrobione porządki, ale bez usuwania zainstalowanych programów (rys. 7c).

W systemach Windows 8 i 8.1 nie ma narzędzia **Przywracanie systemu**, ale można utworzyć obraz systemu operacyjnego. W tym celu należy w oknie **Historia plików** (rys. 6b) kliknąć link **Kopia zapasowa obrazu systemu** – otworzy się okno **Tworzenie obrazu systemu** (rys. 7d). Dalej należy postępować zgodnie z pojawiającymi się na ekranie komunikatami.

Narzędzie tworzenia obrazu systemu jest również dostępne w systemie Windows 7. W oknie **Kopia zapasowa/Przywracanie** (rys. 6a) należy kliknąć link **Utwórz obraz systemu**.

3.3. Odzyskiwanie danych w przypadku ich utraty spowodowanej czynnikami zewnętrznymi

Metody opisane w punkcie 3.2. mają ograniczone zastosowanie w przypadku utraty danych spowodowanej czynnikami zewnętrznymi. Np. w przypadku pożaru domu jest prawdopodobne, że spłonie zarówno sprzęt komputerowy, jak i znajdujące się w pomieszczeniach zewnętrzne nośniki danych z kopiami zapasowymi. Aby uniknąć takiej sytuacji, warto posiadać dodatkową kopię bezpieczeństwa najważniejszych danych w odległym geograficznie miejscu. Dobrym rozwiązaniem jest korzystanie z usług jednego z internetowych serwisów, umożliwiających tworzenie kopii bezpieczeństwa w chmurze obliczeniowej.

Serwisy tego typu są zwykle bezpłatne dla określonego limitu danych (np. do 2 GB); powyżej tego limitu konieczne jest dokonywanie stałych, miesięcznych opłat za przechowywanie danych.

Ćwiczenie 9. Szukamy informacji o serwisach do tworzenia kopii zapasowych w chmurze

Poszukaj w Internecie informacji o serwisach umożliwiających tworzenie kopii zapasowych w chmurze. Porównaj trzy wybrane serwisy, biorąc pod uwagę ilość miejsca dostępnego za darmo oraz koszty przechowywania większej ilości danych.

Ćwiczenie 10. Tworzymy kopie bezpieczeństwa plików

Sporządź kopię bezpieczeństwa wszystkich plików utworzonych przez ciebie na szkolnym komputerze. Kopię możesz umieścić na przenośnym nośniku danych lub w Internecie.

Warto zapamiętać

- Konta użytkownika zakłada się w celu kontroli dostępu do danych przez użytkowników komputerów pracujących np. w sieci.
- Nieupoważniony dostęp do danych może nastąpić w wyniku działań użytkownika lub osób trzecich.
- Programy antywirusowe to podstawowa metoda ochrony przed złośliwym oprogramowaniem.
- Szyfrowanie danych (plików, folderów, dysku) zapewnia wysoki poziom ich ochrony.
- Mimo że istnieją sposoby odzyskiwania utraconych danych, należy chronić dane, stosując kopie bezpieczeństwa.

Pytania i polecenia

1. W jaki sposób możemy chronić komputer przed nieupoważnionymi osobami?
2. Jakich zasad należy przestrzegać, udostępniając swoje dane w Internecie, m.in. na portalach społecznościowych?
3. W jaki sposób osoby nieuprawnione mogą zdobywać poufne informacje? Podaj trzy przykłady. Omów sposoby zabezpieczania się przed tymi zagrożeniami.
4. W jaki sposób należy włączyć automatyczną aktualizację systemu?
5. Dlaczego ważne jest instalowanie na naszych komputerach oprogramowania antywirusowego i firewalla?
6. W jaki sposób można zaszyfrować istotny plik lub folder znajdujący się na naszym komputerze?
7. Jakie czynniki mogą spowodować utratę danych?
8. W jaki sposób można odzyskać utracone dane?

Zadania

1. Sprawdź, jakie informacje na twój temat znajdują się w Internecie. Wpisz w wyszukiwarkę swoje imię i nazwisko (ewentualnie inne dane identyfikujące, np. miasto zamieszkania) i zobacz, czy w wynikach wyszukiwania znajdziesz informacje o sobie.
2. Utwórz dokument tekstowy, w którym zapiszesz cytat ze swojej ulubionej książki. Zaszyfruj plik, korzystając z odpowiedniej opcji edytora tekstu, i prześlij go (np. za pomocą poczty elektronicznej) koledze lub koleżance. Czy odbiorca będzie mógł odczytać zawartość pliku?
3. Przygotuj prezentację na temat bezpieczeństwa i ochrony danych w komputerach i sieciach komputerowych. Jeśli wykorzystasz teksty z podręcznika, pamiętaj o podaniu źródła.

Dla zainteresowanych

4. Zapoznaj się z dokumentacją programu TrueCrypt / VeraCrypt.
5. Znajdź w Internecie więcej informacji na temat odzyskiwania utraconych danych.

Wokół dokumentów komputerowych

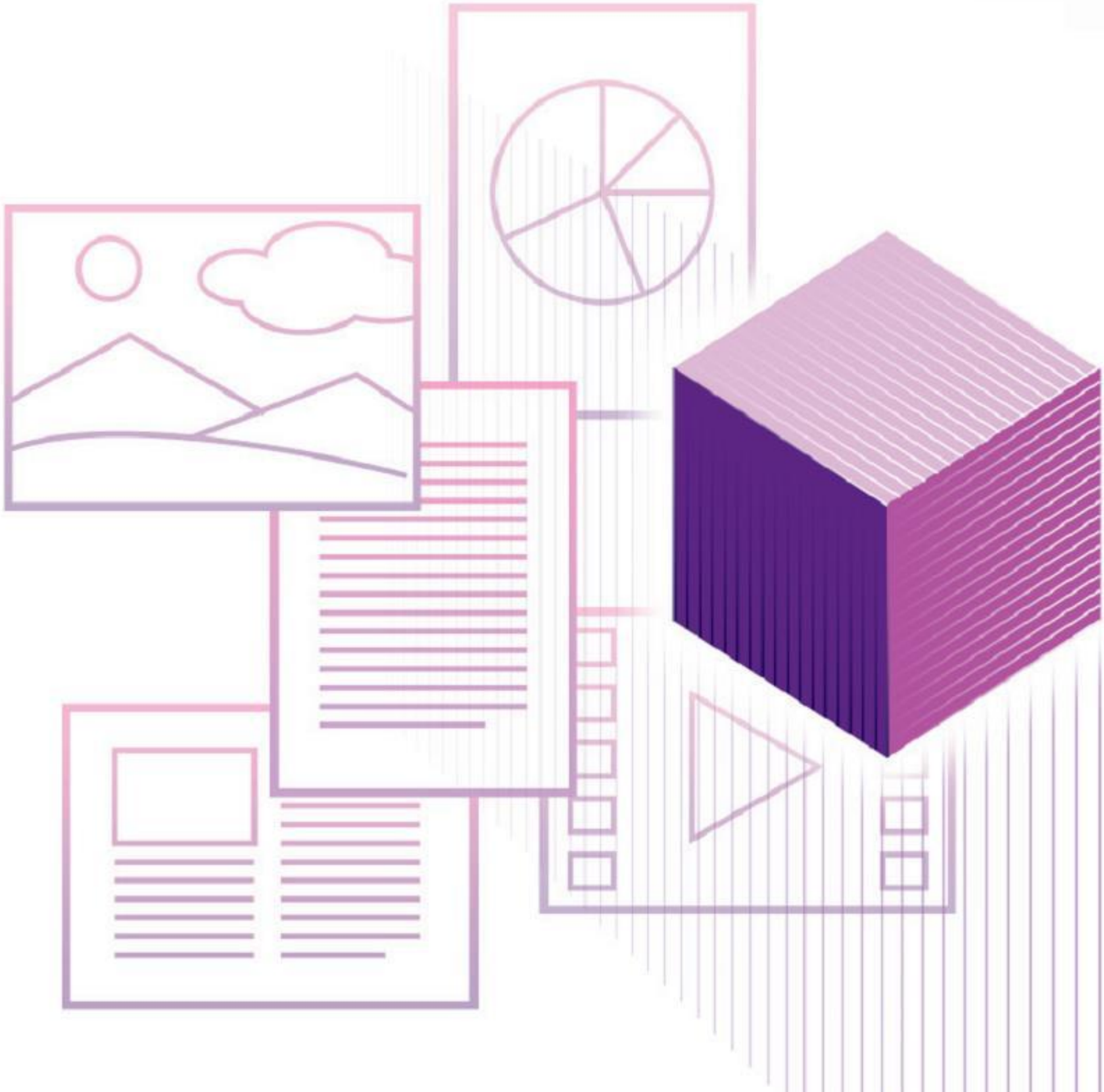
B

Moduł

Temat B1. Formuły, funkcje i wykresy w arkuszu kalkulacyjnym

Temat B2. Filtry oraz tabele i wykresy przestawne w arkuszu kalkulacyjnym

Temat B3. Opracowywanie grafiki wektorowej



Temat B1. Formuły, funkcje i wykresy w arkuszu kalkulacyjnym

1. Tworzenie formuł w arkuszu kalkulacyjnym
2. Formaty danych
3. Formatowanie tabeli arkusza kalkulacyjnego
4. Stosowanie funkcji arkusza kalkulacyjnego
   - 4.1. Wstawianie funkcji
   - 4.2. Funkcja logiczna JEŻELI
   - 4.3. Funkcja statystyczna LICZ.JEŻELI
   - 4.4. Funkcje matematyczne
   - 4.5. Funkcje daty i czasu
5. Przedstawianie danych w postaci wykresu
   - 5.1. Najczęściej używane typy wykresów
   - 5.2. Wykresy funkcji trygonometrycznych i liniowych
6. Przykłady rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego

Warto powtórzyć

1. Jakie znasz zastosowania arkusza kalkulacyjnego?
2. Jak jest zbudowana tabela arkusza kalkulacyjnego?
3. Jakie dane można wprowadzać do komórek arkusza kalkulacyjnego?
4. Jakie znasz własności arkusza kalkulacyjnego?
5. Jakie znasz funkcje arkusza kalkulacyjnego?
6. Jakie znasz typy wykresów?

1. Tworzenie formuł w arkuszu kalkulacyjnym

Arkusz kalkulacyjny jest programem umożliwiającym przedstawianie **danych** (najczęściej liczbowych) w postaci **tabel** i ich przetwarzanie (głównie wykonywanie obliczeń). Jednym z najważniejszych narzędzi arkusza kalkulacyjnego są **funkcje**, pozwalające na łatwe wykonywanie nawet skomplikowanych obliczeń matematycznych, statystycznych czy finansowych. Za pomocą arkusza kalkulacyjnego można także wizualizować dane (przedstawiać na **wykresach**), co ułatwia ich analizę.

Wyrażenia operujące na danych z arkusza (**formuły**) buduje się według ściśle określonych zasad. Formuła rozpoczyna się znakiem =, po którym występują: **dane**, na których wykonywane są obliczenia, **operatory** (np.: +, %) lub **funkcje** z ich **argumentami**.



Danymi w formułach mogą być m.in.: dowolne liczby (stałe), wartości komórek arkusza (zmienne) lub wartości funkcji.

D3

	A	B	C	D	E	F	G	H
1			Frekwencja klas I-III					
2	klasa	liczba uczniów	liczba godzin	liczba godz. ogółem	liczba godz. nieobecności	w tym liczba godz. nieuspr.	frekwencja (% obecności)	liczb godz. nieuspr./na 1 ucznia
3	IA	30	154		622	98		
4	IB	29	148		390	45		
5	IC	28	148		732	76		
6	ID	31	154		290	45		
7	IIA	29	141		555	45		
8	IIB	27	141		890	110		
9	IIC	26	154		489	67		
10	IID	28	154		567	89		
11	IIIA	25	132		444	23		
12	IIIB	26	132		345	45		
13	IIIC	27	136		234	34		
14	IIID	29	132		231	18		

Rys. 1. *Tabela arkusza kalkulacyjnego z fikcyjnymi danymi – ćwiczenie 1.*

Ćwiczenie 1. Tworzymy tabelę w arkuszu kalkulacyjnym

1. W arkuszu kalkulacyjnym utwórz tabelę przedstawioną na rys. 1. Wprowadź fikcyjne dane.
2. Utwórz formuły obliczające: liczbę godzin ogółem, frekwencję i liczbę godzin nieusprawiedliwionych przypadających na jednego ucznia. Wstaw formuły do odpowiednich komórek i, kopiując je odpowiednio do pozostałych komórek, wykonaj obliczenia dla wszystkich klas.
3. Zapisz plik pod nazwą *Frekwencja*.

Formułę można umieścić w komórce o dowolnym adresie.
Ważne są adresy komórek zawarte w formule – muszą dotyczyć danych potrzebnych do obliczenia wyniku.

2. Formaty danych

Arkusz kalkulacyjny pozwala ustalić **format** wyświetlanych danych.
Można go ustalać przed wprowadzeniem danych i po ich wprowadzeniu – dla całej kolumny, wiersza lub dla wybranych, zaznaczonych komórek.

Jeśli ustawiony jest tzw. format **Ogólny**, program automatycznie rozpozna typ danej wprowadzonej do komórki. Teksty wyrównywane są do lewej strony, a liczby do prawej. Liczby umieszczane w komórkach arkusza mają domyślny format **Liczbowy** (w tym formacie możemy określić liczbę cyfr wyświetlanych po przecinku).

Dla danych liczbowych można ustalić również inne formaty:

- **Walutowy** – podobny do **Liczbowego**, z tym że przed liczbą lub po niej dodawany jest symbol waluty.
- **Daty**, **Czasu** – liczby wyświetlają się jako data (np. 2015-11-17) lub jako czas (np. 13:30).
- **Procentowy** – wartości są mnożone przez 100 i wyświetlane ze znakiem %.
- **Ułamkowy** – wyświetla wprowadzoną liczbę w postaci ułamka zwykłego. Ułamek wyświetla się w postaci: *część całkowita*, *spacja*, *licznik/mianownik*, np. 11,25 wyświetli się jako 11 1/4.
- **Naukowy** – liczby wyświetlają się w postaci wykładniczej. Zapis *x,xxE+yy* oznacza iloczyn liczby rzeczywistej *x,xx* (większej od –10 i mniejszej od 10) i liczby 10, podniesionej do potęgi *yy*, np.:
  $105 = 1{,}05 \cdot 10^2 = 1{,}05E+02$;
  $0{,}00000258 = 2{,}58 \cdot 10^{-6} = 2{,}58E-06$.
  Uwaga: Jeżeli kolumna jest za wąska, aby wyświetlić wszystkie cyfry dużej liczby, liczba wyświetli się w postaci wykładniczej. Jeśli jednak szerokość kolumny jest zbyt mała, mogą się wyświetlić znaki ####, przy czym w komórce nadal pamiętana jest właściwa liczba.
- **Specjalny** – dla danych specjalnego typu, np. numerów telefonu, numerów PESEL.
- **Niestandardowy** – umożliwia tworzenie własnych, niestandardowych formatów. Najłatwiej jest wybrać z listy jeden z proponowanych formatów, a następnie zmodyfikować go według potrzeb.

Jeśli wpisany tekst nie mieści się w komórce, będzie wystawał poza nią (aby tego uniknąć, należy w opcjach formatowania zaznaczyć **Zawijaj tekst** lub zwiększyć rozmiar komórki). Liczby natomiast są w takim przypadku zaokrąglane lub przedstawiane w postaci wykładniczej (format **Naukowy**).

Ćwiczenie 2. Zapoznajemy się z formatami danych w arkuszu kalkulacyjnym

Zapoznaj się ze szczegółowym opisem formatów liczbowych stosowanych w arkuszach kalkulacyjnych. Jaki format zastosujesz do wpisania numerów PESEL?

Ćwiczenie 3. Ustalamy formaty danych w arkuszu kalkulacyjnym

Otwórz plik *Frekwencja*, zapisany w ćwiczeniu 1. Dla danych z kolumny *D* ustal format **Liczbowy** bez liczb po przecinku, dla kolumny *G* – **Procentowy**, z dwoma miejscami po przecinku, a dla *H* – **Liczbowy** z jednym miejscem po przecinku. Zapisz plik pod tą samą nazwą.

Wskazówka: Opcję ustalania formatu danych w komórkach tabeli znajdziesz w menu **Format/Komórki**, na karcie **Narzędzia główne** (w grupie **Liczby**) lub w menu kontekstowym komórki (polecenie **Formatuj komórki**). W przypadku zastosowania formatu procentowego, zastanów się, jak zmienić formułę, aby wynik był prawidłowy.

3. Formatowanie tabeli arkusza kalkulacyjnego

Komórki tabeli możemy formatować na wiele sposobów, np. stosować obramowanie i cieniowanie, zmieniać szerokość kolumn i wysokość wierszy. W arkuszu kalkulacyjnym zdefiniowano gotowe style, które umożliwiają formatowanie komórek.

Dane umieszczone w komórkach formatujemy, zmieniając parametry czcionki (te same, co w edytorze tekstu) oraz stosując różne sposoby wyrównywania tekstu w obrębie komórki.

Jeśli poszczególne kolumny lub wiersze nie muszą mieć tych samych rozmiarów, można ustalić ich szerokość i wysokość niezależnie lub skorzystać z opcji **Autodopasowanie** (dla kolumn lub wierszy). Opcje te znajdziemy w menu **Format** lub na karcie **Narzędzia główne** w grupie **Komórki**.

Ćwiczenie 4. Sprawdzamy sposób wyświetlania danych w komórce

Utwórz nowy skoroszyt arkusza kalkulacyjnego. Wpisz do komórki A1 liczbę 58348729008. Liczba zostanie wyświetlona w postaci wykładniczej. Następnie ustaw kursor w tej komórce i kliknij opcję **Autodopasowanie** dla kolumny. Co się zmieniło?

Ćwiczenie 5. Formatujemy komórki tabeli arkusza kalkulacyjnego

1. Otwórz plik *Frekwencja*, zapisany w ćwiczeniu 3. Sformatuj komórki, dodając m.in. zacieniowania i obramowania, podobnie jak pokazano na rys. 1.
2. Zapisz plik pod tą samą nazwą.

W celu zwiększenia czytelności tabeli można zastosować **formatowanie warunkowe**. Taki sposób formatowania, poprzez wyróżnianie określonych komórek, ułatwia analizę umieszczonych w nich wartości, a także porównywanie i ocenianie tendencji.

Komórki możemy wyróżnić kolorystycznie lub stosować zestawy ikon (nie wszystkie wersje programu proponują takie zestawy). Możemy na przykład zastosować inny kolor zacieniowania komórki dla wartości najwyższych, a inny dla najniższych lub odróżnić kolorystycznie wartości ujemne od nieujemnych.

Aby zastosować formatowanie warunkowe:

- zaznacz odpowiednie komórki,
- wybierz opcję **Formatowanie warunkowe** (w menu **Format** lub na karcie **Narzędzia główne**),
- ustal odpowiednie reguły formatowania.

Ćwiczenie 6. Stosujemy formatowanie warunkowe

1. Otwórz plik *Frekwencja*, zapisany w ćwiczeniu 5. Stosując formatowanie warunkowe, wyróżnij wybranym kolorem frekwencję większą niż średnia i liczbę godzin nieusprawiedliwionych przypadających na jednego ucznia większą niż średnia.
2. Zapisz plik pod tą samą nazwą.

Wskazówka: Sprawdź, czy w warunkach dla komórek występuje polecenie **Powyżej średniej**. Nie musisz wtedy obliczać średniej.

	A	B	C	D	E
1	**Lp.**	**Nazwisko**	**Imię**	**Egzamin 1**	**Egzamin 2**
2	1	Zając	Tomasz	44	48
3	2	Nowak	Agnieszka	26	20
4	3	Wrona	Bolesław	49	37
5	4	Kalarus	Krzysztof	30	33
6	5	Kminek	Franciszka	43	46
7	6	Krajewski	Kacper	39	50
8	7	Szymczak	Barbara	40	42
9	8	Grabowski	Jan	26	40
10	9	Janik	Ewa	49	26
11	10	Małyga	Jakub	30	49
12	11	Krawczyk	Maria	43	30
13	12	Stopa	Kasia	33	26
14	13	Kaczka	Stefan	45	49
15	14	Wysocki	Maciej	49	30
16	15	Wieczorek	Zofia	38	43
17	16	Kotlarska	Anna	27	20
18	17	Piechota	Radosław	32	45
19	18	Leszczyński	Tadeusz	39	40
20	19	Sołtys	Beata	29	33
21	20	Kogut	Marian	35	27

Rys. 2. *Przykład formatowania warunkowego w tabeli arkusza kalkulacyjnego z fikcyjnymi danymi. Komórki z wynikami wyższymi niż 40 punktów są zacieniowane na różowo, a z niższymi niż 40 punktów – na żółto. Wynik równy 40 pozostał niewyróżniony*

Ćwiczenie 7. Stosujemy formatowanie warunkowe

1. W arkuszu kalkulacyjnym utwórz tabelę przedstawioną na rysunku 2. Wprowadź fikcyjne dane.
2. Stosując formatowanie warunkowe, wyróżnij wybranym kolorem wyniki z każdego egzaminu wyższe niż 40 punktów, a innym niższe niż 40 punktów.
3. Zapisz plik pod nazwą *Wyniki*.

Wskazówka: W warunkach dla komórek ustaw inny kolor dla **Większe niż** 40 i inny dla **Mniejsze niż** 40.

4. Stosowanie funkcji arkusza kalkulacyjnego

4.1. Wstawianie funkcji

W arkuszu kalkulacyjnym występuje kilka kategorii funkcji, które można wykorzystywać w formułach. Są to m.in.: funkcje matematyczne, logiczne, statystyczne, finansowe, tekstowe oraz daty i czasu.

Większość funkcji arkusza kalkulacyjnego wymaga podawania argumentów, na podstawie których jest obliczana wartość funkcji. Najczęściej argumentami są: odwołania do komórek, liczby, rzadziej tekst lub daty. Argumentami mogą być także wartości innej funkcji lub wyrażenia – w takim przypadku najpierw obliczana jest wartość funkcji lub wyrażenia i dopiero ona zostaje użyta jako argument. Jeżeli argumentów jest kilka, do ich oddzielenia używa się średnika.

Przy wstawianiu do komórki niektórych funkcji (np. SUMA, ŚREDNIA, MIN, MAX) arkusz kalkulacyjny proponuje zakres danych, które będą użyte w danej formule. Podany zakres można zmodyfikować, np. zmieniając adresy komórek lub typ adresowania na mieszane bądź bezwzględne.

Aby użyć w formule funkcji, można:
- wybrać funkcję spośród funkcji dostępnych w programie (klikając przycisk reprezentujący daną funkcję lub wybierając z listy)

lub
- wpisać do komórki odpowiednią formułę zawierającą nazwę wybranej funkcji.

W przypadku, gdy nazwę funkcji wprowadzamy samodzielnie, należy pamiętać o wszystkich koniecznych nawiasach i znakach oraz o sposobie i kolejności zapisu argumentów funkcji. Wygodniej jest skorzystać z okna wpisywania argumentów, które otwiera się po wybraniu danej funkcji.

Opcję wstawiania funkcji znajdziemy w menu **Wstaw** lub na karcie **Formuły**. Można też skorzystać z przycisku f_x, znajdującego się po lewej stronie paska formuły.

Ćwiczenie 8. Stosujemy funkcję ŚREDNIA

1. Otwórz plik *Frekwencja*, zapisany w ćwiczeniu 6. Korzystając z funkcji ŚREDNIA, oblicz średnią frekwencję i średnią liczbę godzin nieusprawiedliwionych przypadających na jednego ucznia. Wyniki umieść odpowiednio w komórkach *G15* i *H15*. Przy okazji sprawdzisz, czy ćwiczenie 6. zostało dobrze wykonane przez program.
2. Zapisz plik pod tą samą nazwą.

4.2. Funkcja logiczna JEŻELI

Funkcja **JEŻELI** podaje jedną z dwóch możliwych wartości, w zależności od tego, czy podany warunek jest spełniony (PRAWDA), czy nie (FAŁSZ). Podstawowa postać funkcji: JEŻELI(*test_logiczny;wartość_jeżeli_prawda;wartość_jeżeli_fałsz*).

Za pomocą funkcji JEŻELI można przeprowadzać testy logiczne na wartościach i formułach. Do budowania warunków używa się operatorów porównań: = (równe), > (większe niż) oraz >= (większe lub równe), < (mniejsze niż) oraz <= (mniejsze lub równe), <> (nierówne, czyli różne).

Argumentem *wartość_jeżeli_prawda* lub *wartość_jeżeli_fałsz* funkcji JEŻELI może być dowolna wartość, m.in. będąca wynikiem kolejnej funkcji JEŻELI.

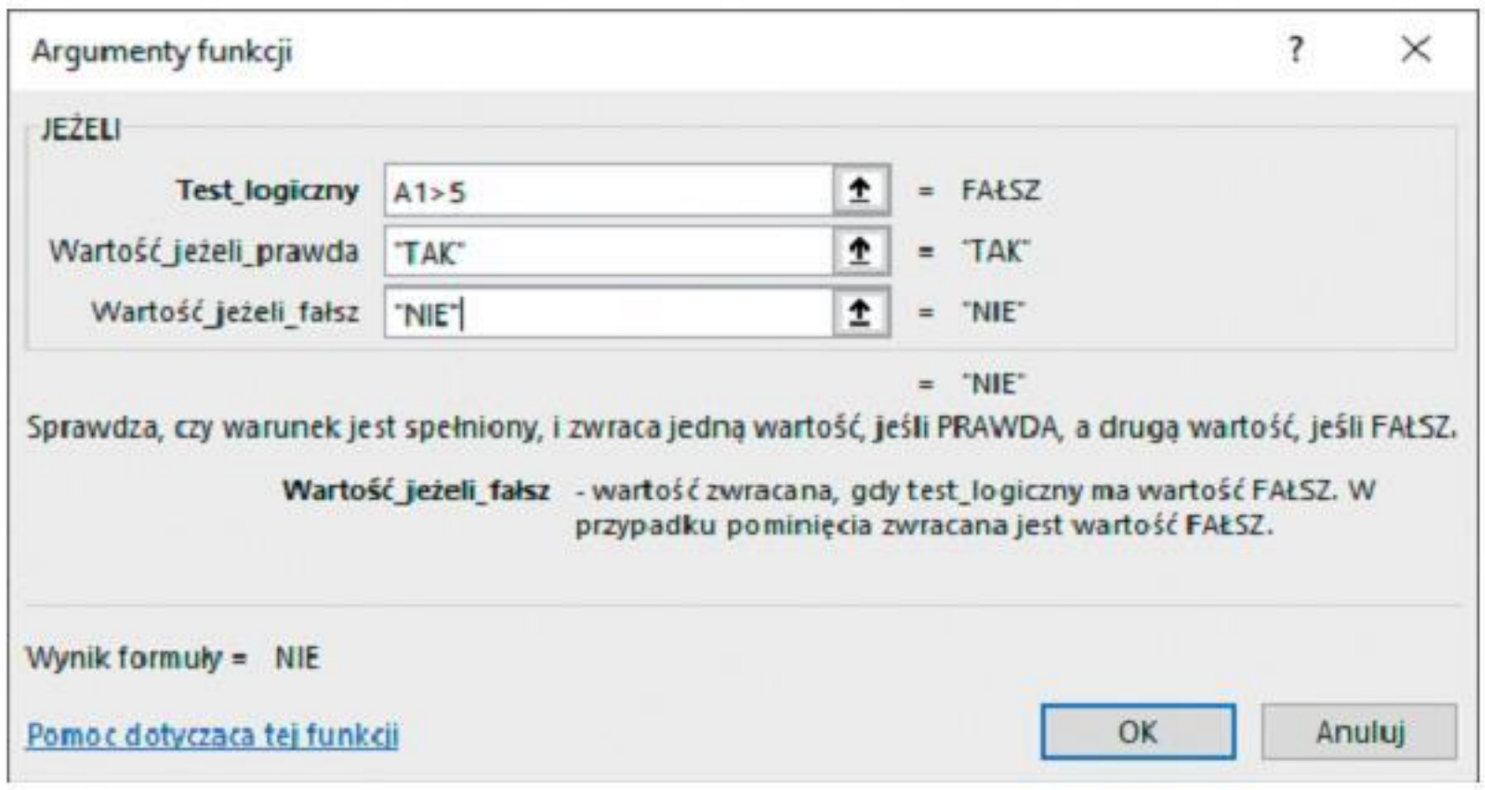
Przykład 1. Stosowanie funkcji JEŻELI

Argumentami funkcji JEŻELI mogą być: tekst, adres komórki, formuła, w tym formuła zawierająca funkcje.
Wartością formuły =JEŻELI(A1>5;"tak";"nie") jest tekst *tak*, jeśli liczba w komórce *A1* jest większa od 5. W przeciwnym przypadku wartością formuły jest tekst *nie*.
Wartością formuły =JEŻELI(B5=0;-25;25) jest liczba -25, jeśli liczba w komórce *B5* jest równa 0. W przeciwnym przypadku wartością formuły jest liczba 25.

Formuła =JEŻELI(B3<>0;SUMA(C2:C10)/B3;"dzielenie przez zero") oblicza iloraz sumy liczb z zakresu *C2:C10* przez liczbę z komórki *B3*, jeżeli liczba ta jest różna od zera. W przeciwnym przypadku wartością formuły jest tekst *dzielenie przez zero*.

Ćwiczenie 9. Stosujemy funkcję JEŻELI

1. Otwórz plik *Wyniki*, zapisany w ćwiczeniu 7. W kolumnie *F* umieść funkcję sumującą punkty uzyskane z obu egzaminów, a w kolumnie *G* – funkcję logiczną JEŻELI, która spowoduje wpisanie wyniku egzaminu: *ZDAŁ*, jeżeli dany uczeń uzyskał wystarczającą sumę punktów z obu egzaminów (np. więcej niż 80 punktów), *NIE ZDAŁ*, jeśli uczeń nie uzyskał wystarczającej liczby punktów.
2. Zapisz plik pod tą samą nazwą.



Rys. 3. *Przykładowe okno określania argumentów funkcji JEŻELI*

4.3. Funkcja statystyczna LICZ.JEŻELI

Funkcja **LICZ.JEŻELI**(zakres;kryterium) oblicza, w ilu komórkach z wybranego zakresu znajdują się wartości spełniające dane kryterium.

Kryterium określa, które komórki będą zliczane. Może być ono podane w postaci liczby, wyrażenia logicznego lub tekstu.

Przykład 2. Stosowanie funkcji LICZ.JEŻELI

Jeśli w komórkach *A1:A30* znajdują się kolejne liczby od 1 do 30, w wyniku formuły =LICZ.JEŻELI(A1:A30;"<=10") otrzymamy liczbę 10 (ponieważ wartości w dziesięciu komórkach są mniejsze lub równe 10).
Jeśli w komórkach *B1:B50* znajdują się napisy *tak* lub *nie*, formuła =LICZ.JEŻELI(B1:B50;"tak") obliczy liczbę komórek, w których umieszczono napis *tak*.
Jeśli w komórkach *C1:C21* znajdują się liczby, formuła =LICZ.JEŻELI(C1:C21;6) obliczy liczbę komórek zawierających liczbę 6.

Ćwiczenie 10. Stosujemy funkcję LICZ.JEŻELI

1. Otwórz plik *Frekwencja*, zapisany w ćwiczeniu 8.
2. Oblicz, ile jest klas, które mają frekwencję większą od wartości średniej, a ile jest klas z niższą frekwencją. Te same obliczenia wykonaj dla liczby godzin nieusprawiedliwionych przypadających na jednego ucznia. Wyniki umieść odpowiednio w komórkach *G16* i *H16* oraz *G17* i *H17*, dla których ustaw format **Liczbowy** (bez miejsc po przecinku).
3. Zapisz plik pod tą samą nazwą.

4.4. Funkcje matematyczne

Funkcja **MODUŁ.LICZBY**(*liczba*) oblicza wartość bezwzględną danej liczby (czyli dla liczby dodatniej wynikiem jest ta sama liczba, a dla liczby ujemnej – liczba przeciwna).

Funkcja **ZAOKR**(*liczba;liczba_cyfr*) zaokrągla liczbę do podanej *liczby_cyfr* po przecinku.

Funkcja **PI**() podaje wartość liczby π z maksymalną dokładnością arkusza (15 miejsc po przecinku).

Przykład 3. Stosowanie funkcji MODUŁ.LICZBY i ZAOKR

Wartością formuły =MODUŁ.LICZBY(-2,5) jest liczba 2,5.
Wartością formuły =MODUŁ.LICZBY(300) jest liczba 300.
Wartością formuły =ZAOKR(9,8765;2) jest liczba 9,88.
Wartością formuły =ZAOKR(87,3548647;5) jest liczba 87,35486.

Ćwiczenie 11. Stosujemy funkcję PI() i ZAOKR

1. Korzystając z funkcji PI(), przygotuj arkusz, w którym na podstawie podanego promienia obliczane będą pole i obwód koła. Wartości promienia wpisz do komórek *A2:A25*.
2. Zmodyfikuj formuły tak, aby wartości pola i obwodu były podawane z dokładnością do dwóch miejsc po przecinku. Sformatuj odpowiednio tabelę, dodaj nagłówki kolumn i obramowania komórek.
3. Zapisz plik pod nazwą *Pole i obwód*.

Funkcja **MOD**(*liczba;dzielnik*) oblicza resztę z dzielenia liczby *liczba* przez *dzielnik*.

Przykład 4. Stosowanie funkcji MOD

Wartością formuły =MOD(15;3) jest liczba 0, a formuły =MOD(12;7) – liczba 5.

Ćwiczenie 12. Stosujemy funkcję MOD

1. Utwórz nową tabelę arkusza kalkulacyjnego. Do kolumn *A* i *B* wpisz po 20 liczb, przy czym liczby z kolumny *A* powinny być większe od liczb z kolumny *B*.
2. W kolumnie *C* utwórz formułę sprawdzającą podzielność liczb z kolumny *A* przez sąsiadujące z nimi liczby z kolumny *B* i wyświetlającą odpowiednie komunikaty: *Liczba z kolumny A dzieli się przez liczbę z kolumny B* lub *Liczba z kolumny A nie dzieli się przez liczbę z kolumny B*.
3. Sformatuj odpowiednio tabelę, dodaj nagłówki kolumn i obramowania komórek.
4. Zapisz plik pod nazwą *Podzielność*.

Wskazówka: W przypadku, gdy reszta z dzielenia dwóch liczb jest równa 0, pierwsza liczba (dzielna) dzieli się przez drugą liczbę (dzielnik). Należy skorzystać z funkcji JEŻELI i zagnieździć w niej funkcję MOD. W formule należy podać adresy odpowiednich komórek.

Funkcja **LOS**() generuje liczbę losową większą od zera lub równą zeru i mniejszą od jeden (wartość funkcji zmienia się po każdym przeliczeniu arkusza, np. po zmianie zawartości jednej z komórek).

Funkcja **ZAOKR.DO.CAŁK**(*liczba*) zaokrągla podaną liczbę w dół, do najbliższej liczby całkowitej (obcina część ułamkową).

Przykład 5. Stosowanie funkcji LOS i ZAOKR.DO.CAŁK

Chcemy wygenerować losowo liczby z przedziału ⟨1; 6⟩, które będą określały liczbę oczek, jaka wypadnie w stu rzutach sześcienną kostką do gry.

Po umieszczeniu w stu wierszach kolumny *A* formuły =LOS() otrzymamy liczby losowe z przedziału ⟨0; 1⟩ – rys. 4.

Jeśli wynik pomnożymy przez 6, otrzymamy liczby z przedziału ⟨0,6). Aby otrzymać wartości całkowite, musimy zastosować funkcję ZAOKR.DO.CAŁK, której argumentem będzie: LOS()*6.

Gdy do stu wierszy kolumny *B* wpiszemy formułę: =ZAOKR.DO.CAŁK(LOS()*6), to jakie otrzymamy wyniki?

Dlaczego ostatecznie formuła powinna mieć postać: =ZAOKR.DO.CAŁK(LOS()*6)+1?

B1

	A	B
1	0,972763397	4
2	0,280848089	2
3	0,968247435	2
4	0,419572623	5
5	0,416676114	6
6	0,803079991	4
7	0,038395371	3
8	0,316451723	4
9	0,999999851	3
10	0,776844256	1
11	0,309097258	2
12	0,044090303	3
13	0,599357763	3

Rys. 4. *Przykład zastosowania funkcji LOS i ZAOKR.DO.CAŁK. W kolumnie* B *wyświetlone są przykładowe wyniki rzutu kostką do gry*

Aby wylosować nowy zestaw liczb, należy nacisnąć klawisz **F9.**

Ćwiczenie 13. Stosujemy funkcję LOS i ZAOKR.DO.CAŁK

1. Uczniowie z każdej klasy w fikcyjnej szkole pełnią dyżury. Kolejność była zawsze ustalana według zapisu w dzienniku, ale postanowiono to zmienić na wybór losowy.
2. Utwórz w arkuszu kalkulacyjnym tabelę pokazaną na rys. 5.
3. W kolumnie *C* umieść formułę generującą liczby losowe z przedziału ⟨1; *n*⟩ (*n* – liczba uczniów w danej klasie), wyznaczającą liczbę z listy porządkowej uczniów w poszczególnych klasach.
4. Zapisz plik pod nazwą *Dyżury*.

C2

	A	B	C
1	KLASA	LICZBA UCZNIÓW W KLASIE	WYLOSOWANY NUMER PORZĄDKOWY DYŻURNEGO
2	1a	32	
3	1b	31	
4	1c	30	
5	1d	33	
6	2a	34	
7	2b	31	
8	2c	28	
9	2d	32	
10	3a	31	
11	3b	29	
12	3c	32	
13	3d	30	

Rys. 5. *Tabela z fikcyjnymi danymi – ćwiczenie 13.*

4.5. Funkcje daty i czasu

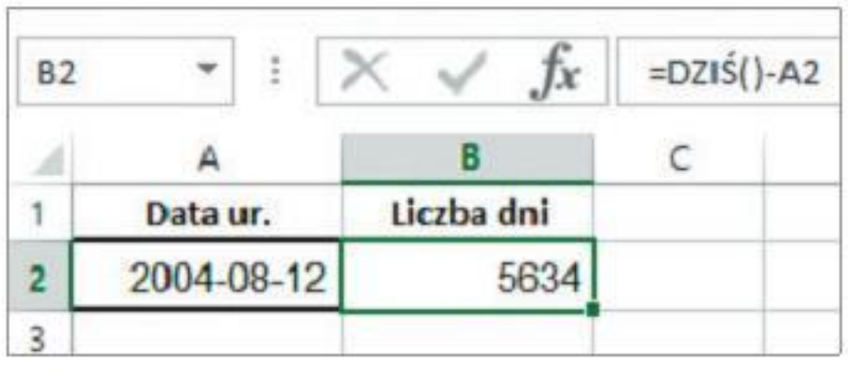
Do obliczeń, w których konieczne jest wyznaczenie czasu (liczby lat, dni, godzin itp.), jaki upłynął między dwoma terminami, przydatne są funkcje należące do kategorii daty i czasu.

Aby wstawić do komórki bieżącą datę,
należy nacisnąć kombinację klawiszy **Ctrl + ;**.
Aby wstawić do komórki bieżącą godzinę,
należy nacisnąć kombinację klawiszy **Ctrl + Shift + ;**.

Jeśli data lub godzina wstawione do komórki mają być aktualizowane, np. przy każdorazowym przeliczeniu arkusza, do ich wstawienia należy użyć funkcji **DZIŚ** lub **TERAZ**. Wartością tych funkcji są specjalne liczby, oznaczające odpowiednio bieżącą datę lub datę i czas. Jeśli przed wprowadzeniem funkcji w komórce ustawiony był format **Ogólny**, to wynik funkcji DZIŚ zostanie wyświetlony w postaci daty, a wynik funkcji TERAZ – w postaci daty i godziny. Ponieważ data i godzina są pobierane z zegara systemowego, należy zwrócić uwagę, czy w komputerze, za pomocą którego wykonywane są obliczenia, te wartości są aktualne. Zapisywanie daty i czasu w postaci liczb umożliwia wykonywanie na nich operacji arytmetycznych – np. odejmowania w celu ustalenia liczby dni pomiędzy dwiema datami.

Przykład 6. Stosowanie funkcji DZIŚ

Aby obliczyć, ile dni upłynęło od dnia twoich narodzin, w arkuszu kalkulacyjnym utwórz tabelę przedstawioną na rys. 6. W komórce *A2* wpisz datę swoich urodzin w formacie *rrrr-mm-dd*, gdzie *rrrr* oznacza rok, *mm* – miesiąc, *dd* – dzień.
W komórce *B2* wpisz formułę =DZIŚ()-A2 i naciśnij klawisz **ENTER**.
Zmień formatowanie komórki na ogólne.



Rys. 6. *Przykład zastosowania funkcji DZIŚ*

Ćwiczenie 14. Stosujemy funkcję DZIŚ

1. Korzystając z funkcji daty, oblicz w arkuszu kalkulacyjnym:
   a. ile dni trwała II wojna światowa,
   b. ile dni upłynęło od zakończenia II wojny światowej.
2. Zapisz plik pod nazwą *Wojna*.

Wskazówka: Komórki, w których będą przeprowadzane obliczenia, powinny posiadać format ogólny lub liczbowy.

5. Przedstawianie danych w postaci wykresu

5.1. Najczęściej używane typy wykresów

Arkusz kalkulacyjny oferuje kilkanaście typów wykresów. Mają one różne zastosowania, zależnie od typu danych i relacji, jakie mają przedstawiać. Wygląd poszczególnych typów wykresów można zobaczyć w oknie podglądu kreatora wykresów.

Najczęściej używane typy wykresów

- **Kolumnowy** – obrazuje zmiany wartości. Umożliwia również porównywanie różnych wartości.
- **Słupkowy** – odmiana wykresu kolumnowego.
- **Liniowy** – stosowany do przedstawiania tendencji wzrostowych lub spadkowych (np. wzrost produkcji, spadek zatrudnienia).
- **Kołowy** – pokazuje zależności między poszczególnymi częściami a całością.
- **XY (punktowy)** – wyświetla serie danych w postaci punktów w układzie współrzędnych. Stosowany jest np. do rysowania wykresów funkcji.
- **Pierścieniowy** – odmiana wykresu kołowego. Umożliwia przedstawienie kilku serii danych.
- **Warstwowy** – uwidacznia, jak zmiana wartości przebiegała w czasie. Wyświetla sumę poszczególnych wartości, a także zależności pomiędzy nimi.
- **Stożkowy**, **cylindryczny**, **ostrosłupowy** – wykresy te dodają efekty trójwymiarowe do wykresów słupkowych i kolumnowych.

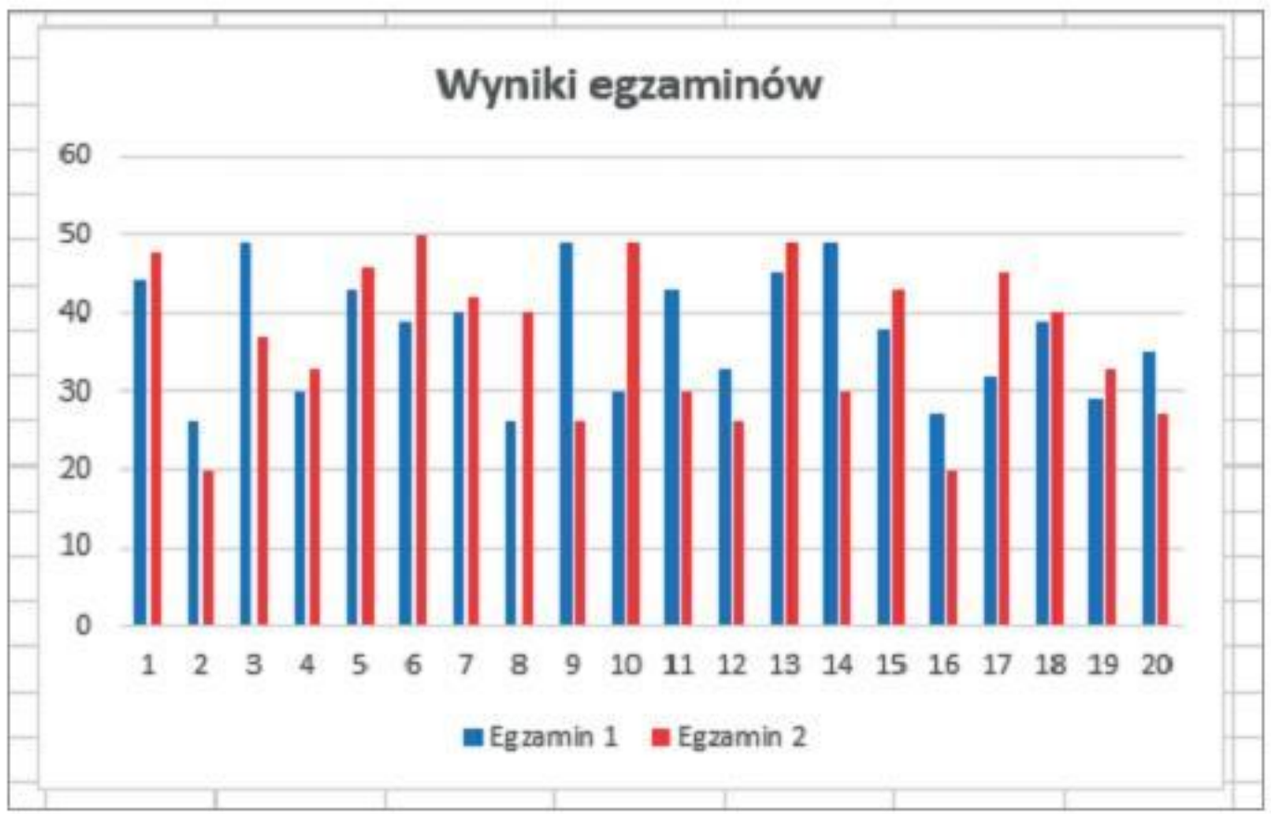
Ćwiczenie 15. Dostosowujemy typ wykresu do danych

Jakiego typu wykresu użyjesz do przedstawienia:

a. wyników pomiarów temperatury powietrza o określonych godzinach?
b. zmiany wartości dochodów firmy w kolejnych latach?
c. wielkości produkcji krzeseł różnych firm, jak również sumarycznej wielkości produkcji i zmian wartości w czasie?

Ćwiczenie 16. Tworzymy wykres w arkuszu kalkulacyjnym

1. Otwórz plik *Wyniki*, zapisany w ćwiczeniu 9. Utwórz wykres porównujący wyniki z obydwu egzaminów dla poszczególnych uczniów. Na wykresie powinny być umieszczone: legenda, tytuł, etykiety osi kategorii (X) (rys. 7.).
2. Zapisz plik pod tą samą nazwą.



Rys. 7. *Przykładowy wykres kolumnowy dla dwóch serii danych*

Wykres można też utworzyć, zaznaczając serie danych i naciskając klawisz **F11**. Wykres zostanie umieszczony w tym samym skoroszycie, w osobnym arkuszu, którego zakładka pojawi się na dole okna programu. Metoda ta sprawdza się jednak tylko w przypadku nieskomplikowanych wykresów. Tworząc złożone wykresy, i tak trzeba zmienić wiele ustawień.

Aby utworzyć wykres dla serii danych, które nie leżą w sąsiadujących kolumnach, należy zaznaczyć dane, trzymając wciśnięty klawisz **Ctrl**. Można również najpierw wykonać wykres dla jednej serii, a każdą następną zaznaczyć, skopiować do **Schowka** (**Ctrl+C**), uaktywnić wykres i wkleić do niego (**Ctrl+V**) zaznaczoną serię danych.

5.2. Wykresy funkcji trygonometrycznych i liniowych

Przykład 7. Rysowanie wykresu funkcji sinus w przedziale od 0 do 360 stopni

W arkuszu kalkulacyjnym argumenty funkcji trygonometrycznych podaje się w radianach, a nie w stopniach.

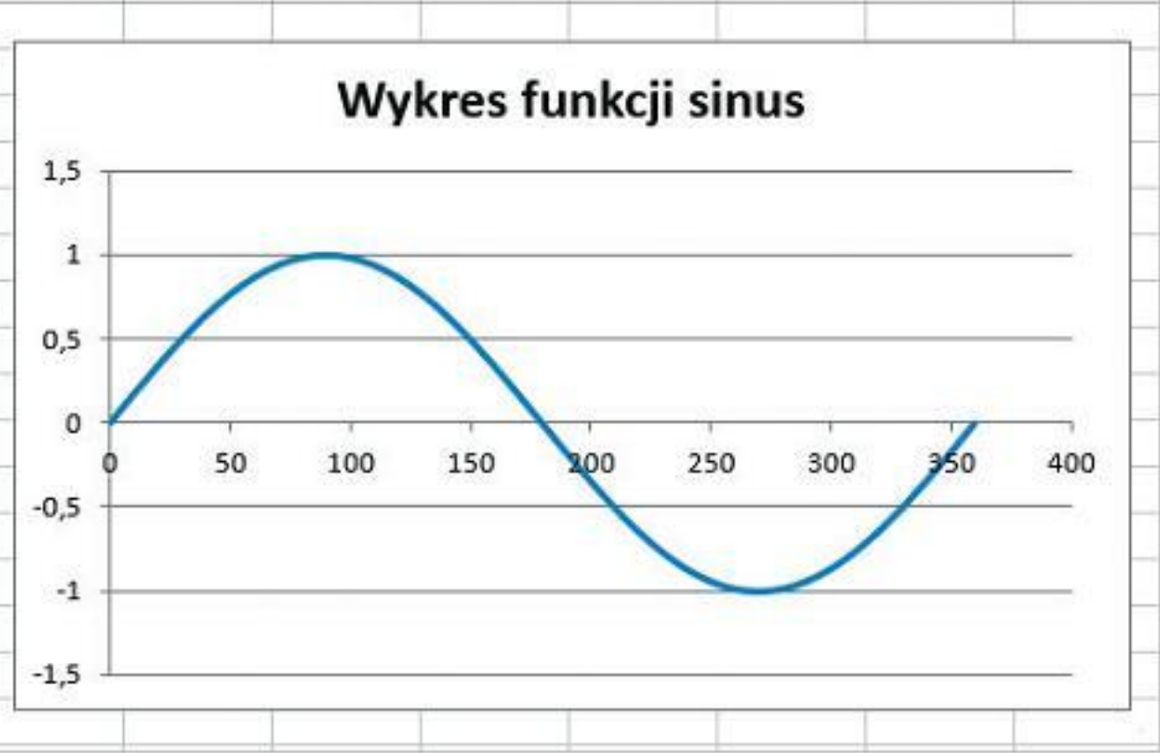
Chcemy narysować wykres funkcji w przedziale ⟨0°; 360°⟩.

1. W nowym arkuszu, w kolumnie *A* (zaczynając od komórki *A2*), wpisz argumenty funkcji w zadanych odstępach, np. od 0° do 360°, co 10° (najlepiej skorzystaj z możliwości automatycznego wpisywania wartości różniących się o pewną stałą – porada 7., temat B2).
2. W kolumnie *B* wprowadź formuły przeliczające stopnie na radiany, np. =A2*PI()/180).
3. W kolumnie *C* wprowadź formuły obliczające wartości funkcji dla zadanych parametrów, np. =SIN(B2). Wartości w kolumnach *B* i *C* można zaokrąglić, np. do dwóch miejsc po przecinku.
4. Zaznacz kolumny *A* i *C* oraz wybierz opcję tworzenia wykresów. Jako typ wybierz wykres XY (punktowy). Dane z kolumny *A* posłużą jako opis osi X.
5. Usuń z wykresu legendę (lub odznacz odpowiednią opcję w oknie tworzenia wykresu) – nie jest tu potrzebna.
6. Umieść tytuł wykresu (rys. 8.).

Uwaga: Opisaną wyżej metodą można narysować wykres każdej funkcji trygonometrycznej, odpowiednio zmieniając formułę z kolumny *C* i dostosowując wartości argumentów w kolumnie *A* (nie wszystkie funkcje trygonometryczne są określone w całym zbiorze liczb rzeczywistych).

B2 | =A2*PI()/180

	A	B	C
1	stopnie	radiany	sinus
2	0	0	0
3	10	0,174533	0,173648178
4	20	0,349066	0,342020143
5	30	0,523599	0,5
6	40	0,698132	0,64278761
7	50	0,872665	0,766044443
8	60	1,047198	0,866025404
9	70	1,22173	0,939692621
10	80	1,396263	0,984807753
11	90	1,570796	1
12	100	1,745329	0,984807753
13	110	1,919862	0,939692621
14	120	2,094395	0,866025404
15	130	2,268928	0,766044443
16	140	2,443461	0,64278761



Rys. 8. *Przykład wykresu* ***XY*** *(****punktowego****) – wykres funkcji sinus*

Ćwiczenie 17. Tworzymy wykres funkcji trygonometrycznej

1. Korzystając z przykładu 7., wykonaj za pomocą arkusza kalkulacyjnego wykres funkcji: $f(x) = 2 \sin x$ (w przedziale od 0 do 720 stopni).
2. Zapisz plik pod nazwą *Funkcja sinus*.

Do rysowania wykresów funkcji liniowej $y = ax + b$ najlepiej zastosować typ wykresu **XY (punktowy)**. Należy wybrać rodzaj wykresu, w którym punkty danych są połączone liniami. Wartości argumentów x najlepiej umieścić w jednej kolumnie (lub jednym wierszu), a odpowiadające im wartości funkcji – w kolejnej kolumnie (wierszu). Wówczas od razu widzimy, jaka jest wartość funkcji y dla danego argumentu x, oraz możemy sprawnie zaznaczyć zakres komórek potrzebny do utworzenia wykresu.

Ćwiczenie 18. Tworzymy wykres funkcji liniowej

1. W arkuszu kalkulacyjnym przygotuj odpowiednią tabelę z danymi i utwórz wykres funkcji liniowej: $y = -7x - 3$.
2. Zapisz plik pod nazwą *Funkcja liniowa*.

6. Przykłady rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego

Myślenie komputacyjne związane jest z rozwiązywaniem problemów z różnych dziedzin z wykorzystaniem metod wywodzących się z informatyki.

Należy sformułować problem w taki sposób, aby do jego rozwiązania było możliwe zastosowanie komputera. Dane powinny być logicznie zorganizowane, aby było możliwe wyciąganie wniosków z analizy ich powiązań. Reprezentację danych można przedstawić w postaci modelu lub symulacji.

Powinniśmy tak zaprojektować sposób rozwiązania, aby był jak najbardziej efektywny i optymalnie wykorzystywał możliwości komputera.

Ważną cechą rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego jest to, aby doświadczenie nabyte przy rozwiązywaniu jednego problemu mogło zostać wykorzystane przy rozwiązywaniu innych problemów.

Pokażemy sposób rozwiązywania wybranych problemów, uwzględniając podstawowe etapy myślenia komputacyjnego:

1. określenie problemu,
2. określenie modeli i pojęć,
3. znalezienie rozwiązania,
4. zaprogramowanie i testowanie rozwiązania.

Problem 1. Skrzyżowanie ulic

Etap 1. Określenie problemu

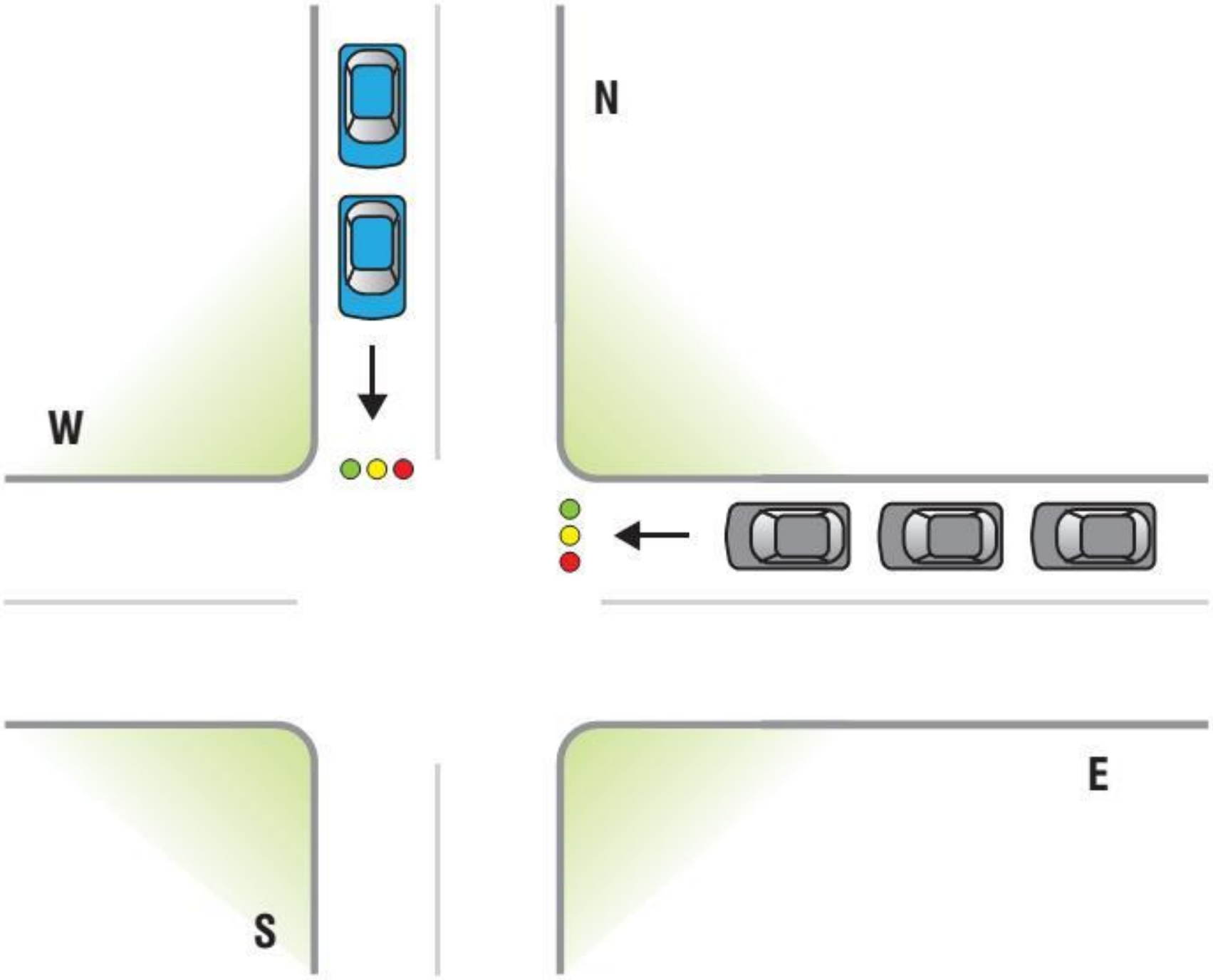
Chcemy zbadać ruch na skrzyżowaniu ulic i znaleźć optymalne ustawienie zielonych świateł, aby jak najmniej samochodów czekało na przejazd przez skrzyżowanie.

Dane do zadania musisz zgromadzić samodzielnie, badając rzeczywistą sytuację na drodze (w temacie podano fikcyjne dane). W tym celu udaj się na wybrane skrzyżowanie i policz, ile samochodów przejeżdża w określonym kierunku w cyklu zmiany świateł. Obserwuj przynajmniej piętnaście pełnych cykli. Poza tym sprawdź, jaki jest

czas przejazdu przez skrzyżowanie jednego samochodu i ile trwa jeden cykl zmiany świateł. Rezultatem rozwiązania tego problemu będzie informacja, przy jakich parametrach cyklu zmiany świateł możliwe jest, aby obydwa kierunki były przejezdne.

Etap 2. Określenie modeli i pojęć

Dla uproszczenia modelu rzeczywistej sytuacji na skrzyżowaniu ulic będziemy badać tylko ruch z północy na południe i ze wschodu na zachód (jak pokazano na rysunku 9.). Przyjmujemy, że samochody jadą tylko na wprost.



Rys. 9. *Schematyczny rysunek przedstawiający przykładowe skrzyżowanie*

Model jest uproszczony, ponieważ wykonanie złożonego modelu, w którym uwzględnilibyśmy wszystkie warianty ruchu samochodów na skrzyżowaniu i zmiany wszystkich świateł byłoby skomplikowane i trudne do realizacji. Po sprawdzeniu poprawności działania prostego modelu można go rozbudowywać.

Przyjmiemy następujące oznaczenia kierunków:

- NS – z północy na południe (azymut 180°),
- EW – ze wschodu na zachód (azymut 270°).

W modelu będziemy posługiwać się następującymi pojęciami:

- *cykl zmiany świateł* – okres od włączenia światła zielonego na kierunku wschodnio-zachodnim do ponownego włączenia zielonego światła na tym kierunku,
- *zielone światło na kierunku NS* – czas wyświetlania zielonego światła w jednym cyklu na kierunku północ-południe,
- *zielone światło na kierunku EW* – czas wyświetlania zielonego światła w jednym cyklu na kierunku wschód-zachód,

- *efektywny czas cyklu* – suma czasu wyświetlania zielonych świateł w obydwu kierunkach (z pominięciem przerwy na zmianę świateł),
- *liczba samochodów z kierunku NS* – liczba samochodów, które przyjeżdżają z kierunku północnego podczas trwania jednego cyklu zmiany świateł,
- *liczba samochodów z kierunku EW* – liczba samochodów, które przyjeżdżają z kierunku wschodniego podczas trwania jednego cyklu zmiany świateł,
- *czas przejazdu przez skrzyżowanie* – czas przejazdu przez skrzyżowanie jednego samochodu, rozumiany jako parametr przepustowości skrzyżowania.

Etap 3. Znalezienie rozwiązania

Podczas badania (obserwowania) rzeczywistej sytuacji powinniśmy zaobserwować i zebrać następujące informacje:

- *cykl zmiany świateł* – w naszym przykładzie przyjmiemy, że cały cykl zmiany świateł trwa 60 sekund, a *efektywny czas cyklu* to 50 sekund,
- *liczba samochodów z kierunku NS* – liczba samochodów, które przyjechały z kierunku północnego podczas jednego cyklu. Będziemy potrzebować piętnastu takich wartości (w piętnastu cyklach). Dane zbieramy, licząc nowo pojawiające się samochody od momentu rozpoczęcia cyklu. Jeśli na przykład 3 samochody nie zdążyły przejechać przez światła i pozostały z poprzedniego cyklu, nie liczymy ich już w następnym cyklu,
- *liczba samochodów z kierunku EW* – liczba samochodów, które przyjechały z kierunku wschodniego podczas jednego cyklu. Będziemy również potrzebować piętnastu takich wartości. Dane zbieramy analogicznie i w tym samym czasie, co dla kierunku NS,
- *czas przejazdu przez skrzyżowanie* – wyznaczymy, zliczając, ile samochodów oczekujących na przejazd przez skrzyżowanie zdąży przez nie przejechać w określonym czasie od zapalenia zielonego światła, a następnie dzieląc ten czas przez liczbę samochodów. Na przykład: jeśli w czasie 30 sekund od zapalenia zielonego światła przez skrzyżowanie przejedzie 6 samochodów, to *czas przejazdu przez skrzyżowanie* wynosi 5 sekund.

Po zebraniu i spisaniu wszystkich informacji należy zastanowić się nad sposobem rozwiązania oraz wybrać narzędzie (program komputerowy), które umożliwi nam zgromadzenie wszystkich danych i ich odpowiednie przetworzenie (wykonanie potrzebnych obliczeń).

Odpowiednim narzędziem jest arkusz kalkulacyjny, w którym możemy zgromadzić wiele danych (umieszczając je w komórkach arkusza), wykonywać na nich różne obliczenia (stosując funkcje arkusza) i na tej podstawie utworzyć odpowiedni model rozwiązania problemu.

Podczas badania rzeczywistego modelu spisaliśmy tylko liczby samochodów nadjeżdżających z każdego kierunku w piętnastu cyklach zmiany świateł. Jest to za mało danych, aby badać model. Jednak spisane dane mogą posłużyć nam do symulacji ruchu na tym skrzyżowaniu. Korzystając z nich, możemy losować różne układy tych danych i w ten sposób symulować różne kombinacje liczb samochodów, które przybyły na skrzyżowanie z kierunku północnego i wschodniego.

Zebrane dane empiryczne wprowadzimy do tabeli arkusza kalkulacyjnego (rys. 10.).

A3 | 1

	A	B	C
1	Dane empiryczne		
2	numer cyklu zmiany świateł	liczba samochodów z kierunku NS	liczba samochodów z kierunku EW
3	1	4	3
4	2	6	2
5	3	8	2
6	4	4	3
7	5	6	4
8	6	9	1
9	7	7	0
10	8	6	2
11	9	6	3
12	10	7	2
13	11	6	3
14	12	7	6
15	13	8	3
16	14	9	2
17	15	11	1
18			

Rys. 10. *Przykładowe (fikcyjne) dane zebrane w czasie badania ruchu na skrzyżowaniu*

Aby wylosować inny układ liczb z kolumny *B* (samochodów jadących z północy na południe), potrzebujemy funkcji, która poda wartość z komórki wybranej z zakresu *B3:B17* dla wylosowanego numeru cyklu (od 1 do 15). Podobnie, aby wylosować inny układ liczb z kolumny *C* (samochodów jadących ze wschodu na zachód), potrzebujemy funkcji, która poda wartość z komórki wybranej z zakresu *C3:C17* dla wylosowanego numeru cyklu (od 1 do 15).

Aby wylosować liczbę całkowitą z przedziału od 1 do 15, zastosujemy funkcję LOS.ZAKR.

LOS.ZAKR(*dół*;*góra*) zwraca losową liczbę całkowitą z podanego przedziału, gdzie *dół* to liczba najmniejsza, a *góra* to liczba największa w przedziale.

Na przykład LOS.ZAKR(1;15) wyświetli losową liczbę całkowitą z przedziału ⟨1, 15⟩.

Natomiast, aby dla wylosowanego numeru cyklu wybierać liczby np. z komórek od *B3* do *B17*, możemy zastosować funkcję WYSZUKAJ.PIONOWO.

WYSZUKAJ.PIONOWO(*szukana_wartość*;*tabela_tablica*;*nr_indeksu_kolumny*; *przeszukiwany_zakres*) wyszukuje wartość w pierwszej kolumnie od lewej podanego zakresu *tabela_tablica* i zwraca wartość z tego samego wiersza w kolumnie *nr_indeksu_kolumny*: przybliżoną (1/ PRAWDA) lub dokładną (0/ FAŁSZ).

Musimy jeszcze policzyć, ile samochodów zostało przed skrzyżowaniem po zakończonym cyklu świateł – oddzielnie z kierunków północ-południe i wschód-zachód. W tym celu należy od liczby samochodów *liczba samochodów z kierunku NS* odjąć liczbę samochodów, która zdążyła przejechać przez skrzyżowanie w tym kierunku. A tę ostatnią obliczymy, dzieląc czas wyświetlania zielonego światła w jednym cyklu na kierunku północ-południe przez czas przejazdu przez skrzyżowanie jednego samochodu (*zielone światło na kierunku NS* przez *czas przejazdu przez skrzyżowanie*).

Etap 4. Zaprogramowanie i testowanie rozwiązania

Rozwiązanie przedstawiamy w arkuszu kalkulacyjnym. Jeśli dobrze przemyśleliśmy sposób rozwiązania, to pozostaje nam już tylko zapisać i wstawić odpowiednie formuły. Na rysunku 11. pokazano przykładowy projekt tabeli do rozwiązania zadania.

Wykorzystamy funkcję WYSZUKAJ.PIONOWO i w komórce *E3* umieścimy formułę: =WYSZUKAJ.PIONOWO(LOS.ZAKR(1;15);$A$3:$B$17;2;0), która wyszuka wartość w kolumnie *A* w komórkach z *A3:A17* i zwróci dokładną wartość z tego samego wiersza w kolumnie drugiej, czyli *B*. Jeśli wylosowaną liczbą będzie 12, to funkcja zwróci zawartość komórki *B14*, czyli liczbę 7.

Aby obliczyć, ile samochodów pozostało po cyklu, należy od liczby samochodów, które nadjechały, odjąć liczbę samochodów, które zdążyły przejechać przez skrzyżowanie. W każdym kolejnym cyklu do liczby samochodów, która najechała, dodajemy liczbę samochodów, która została z poprzedniego cyklu.

Musimy zadbać, aby w formule obliczającej, ile samochodów nie opuściło w danym cyklu skrzyżowania, nie pojawiły się ujemne liczby samochodów. W tym celu można zastosować funkcję MAX.

W komórce *F3* możemy umieścić formułę:
=MAX(0; E3-$J$4/$J$3), która zwróci zero dla ujemnego wyniku obliczeń.

Odpowiednio w komórce *F4* umieścimy formułę:
=MAX(0;F3+E4-$J$4/$J$3), która uwzględni samochody, które pozostały z poprzedniego cyklu. Formułę należy skopiować do pozostałych komórek w kolumnie *F* (od *F5* do *F17*).

E3 =WYSZUKAJ.PIONOWO(LOS.ZAKR(1;15);$A$3:$B$17;2;0)

	A	B	C	D	E	F	G	H	I	J	K
1	Dane empiryczne			Symulacja danych - losowanie liczby samochodów							
2	numer cyklu zmiany świateł	liczba samochodów z kierunku NS	liczba samochodów z kierunku EW	numer cyklu zmiany świateł	liczba samochodów z kierunku NS	liczba samochodów, która zostaje po cyklu	liczba samochodów z kierunku EW	liczba samochodów, która zostaje po cyklu	Parametry cyklu zmiany świateł		
3	1	4	3	1	7	2	2	0	czas przejazdu przez skrzyżowanie	5	s
4	2	6	2	2	9	6	3	0	zielone światło na kierunku NS	**25**	s
5	3	8	2	3	9	10	2	0	zielone światło na kierunku EW	25	s
6	4	4	3	4	9	14	3	0	efektywny czas cyklu zmiany świateł	50	s
7	5	6	4	5	7	16	2	0			
8	6	9	1	6	6	17	2	0			
9	7	7	0	7	4	16	2	0			
10	8	6	2	8	9	20	2	0			
11	9	6	3	9	6	21	6	1			
12	10	7	2	10	9	25	2	0			
13	11	6	3	11	6	26	0	0			
14	12	7	6	12	7	28	3	0			
15	13	8	3	13	4	27	6	1			
16	14	9	2	14	7	29	0	0			
17	15	11	1	15	4	28	3	0			

Rys. 11. *Przykładowy projekt tabeli arkusza kalkulacyjnego – symulacja ruchu samochodów na skrzyżowaniu*

Tabelę z parametrami cyklu zmiany świateł uzupełniamy danymi wynikającymi z obserwacji. W naszym przykładzie przyjmujemy, że *czas przejazdu przez skrzyżowanie* = 5 s, *efektywny czas cyklu* = 50 s, *zielone światło na kierunku EW* = 25 s, a *zielone światło na kierunku NS* = 25 s.

Naciskając klawisz **F9**, możemy ponownie losować liczby samochodów w obydwu kierunkach i obserwować symulację.

W naszym przykładzie obserwujemy, że na kierunku NS liczba samochodów oczekujących na przejazd zwiększa się (tworzy się korek), podczas gdy kierunek EW jest przejezdny. Żeby poszukać rozwiązania tej sytuacji, będziemy wprowadzać z klawiatury inne wartości parametru *zielone światło na kierunku NS* (np. zwiększać o 5 sekund) do momentu, aż liczby samochodów pozostających po zakończeniu cyklu w obydwu kierunkach będą najmniejsze. Naciskając klawisz **F9**, możemy wielokrotnie sprawdzać rozwiązanie w różnych losowanych kombinacjach. Należy pamiętać, aby w komórce *J5* dodać formułę =J6-J4, która będzie automatycznie obliczać *czas zielonego światła na kierunku EW*. Efekt symulacji widzimy na rysunku 12.

H4 =MAX(0;H3+G4-$J$5/$J$3)

	A	B	C	D	E	F	G	H	I	J	K
1	Dane empiryczne			Symulacja danych - losowanie liczby samochodów							
2	numer cyklu zmiany świateł	liczba samochodów z kierunku NS	liczba samochodów z kierunku EW	numer cyklu zmiany świateł	liczba samochodów z kierunku NS	liczba samochodów, która zostaje po cyklu	liczba samochodów z kierunku EW	liczba samochodów, która zostaje po cyklu	Parametry cyklu zmiany świateł		
3	1	4	3	1	6	0	3	0	czas przejazdu przez skrzyżowanie	5	s
4	2	6	2	2	8	1	2	0	zielone światło na kierunku NS	**35**	s
5	3	8	2	3	4	0	4	1	zielone światło na kierunku EW	15	s
6	4	4	3	4	6	0	1	0	efektywny czas cyklu zmiany świateł	50	s
7	5	6	4	5	4	0	3	0			
8	6	9	1	6	6	0	2	0			
9	7	7	0	7	7	0	2	0			
10	8	6	2	8	8	1	2	0			
11	9	6	3	9	9	3	3	0			
12	10	7	2	10	7	3	2	0			
13	11	6	3	11	8	4	4	1			
14	12	7	6	12	6	3	1	0			
15	13	8	3	13	4	0	3	0			
16	14	9	2	14	7	0	3	0			
17	15	11	1	15	9	2	2	0			
18											

Rys. 12. *Symulacja ruchu samochodów na skrzyżowaniu – ustalenie optymalnego czasu trwania zielonego światła*

Ćwiczenie 19. Realizujemy rozwiązanie problemu „Skrzyżowanie ulic" w arkuszu kalkulacyjnym

1. W arkuszu kalkulacyjnym utwórz tabelę, umieść dane i odpowiednie formuły, korzystając z opisanych wskazówek.
2. Przetestuj rozwiązanie, przyjmując kolejno liczbę *zielone światło na kierunku NS* ze zbioru {5, 10, 15, 20, 25, 30, 35, 40, 45}. Wylosuj kilkakrotnie inny zestaw liczb w kolumnach *E* i *G*, naciskając klawisz **F9**. Co zauważasz? Skomentuj otrzymywane rezultaty dla poszczególnych danych. Dla jakich danych wyniki są najbardziej optymalne, czyli jak powinny być według ciebie ustawione zielone światła na kierunkach północ-południe i wschód-zachód?
3. Zapisz plik pod nazwą *Skrzyżowanie*.

Ćwiczenie 20. Rozbudowujemy model

1. Zmodyfikuj model tak, żeby można było sprawdzać wpływ czasu trwania cyklu zmiany świateł na przejezdność skrzyżowania.
2. Ustal minimalny czas trwania cyklu zmiany świateł, przy którym skrzyżowanie jest przejezdne
3. Zapisz plik pod nazwą *Skrzyżowanie_modyfikacja*.

Problem 2. Sprzedawca gazet

Etap 1. Określenie problemu

Sprzedawca gazet zamawia codziennie gazety u dystrybutora, za każdym razem określając, ile gazet chce zamówić na dany dzień. Kupuje gazety w paczkach, płacąc za jedną paczkę gazet cenę $c1 = 100$ zł, a sprzedaje ją po cenie $c2 = 180$ zł. Jednak liczba osób, które każdego dnia kupują gazety, jest losowa i zdarza się, że nie wszystkie gazety zostaną sprzedane. Sprzedawca może odsprzedać gazety dystrybutorowi, ale tylko w pełnych paczkach, w cenie $c3 = 50$ zł za paczkę. Na potrzeby tego modelu zakładamy, że jeśli sprzedawca rozpocznie sprzedaż z danej paczki, to sprzeda z niej wszystkie gazety.

Sprzedawca notował dane sprzedaży i wie, ile paczek gazet sprzedał w ciągu ostatnich dziesięciu dni. Jego zapisy przedstawia tabela 1.

Chcemy pomóc sprzedawcy znaleźć taką liczbę paczek gazet, aby sprzedał ich jak najwięcej, a jak najmniej oddawał z powrotem do dystrybutora.

Liczba sprzedanych gazet w ciągu 10 dni										
Dzień	1	2	3	4	5	6	7	8	9	10
Liczba sprzedanych paczek gazet	12	13	11	12	13	13	12	15	13	14

Tabela 1. *Dane dotyczące sprzedaży gazet w ostatnich dziesięciu dniach*

Rozwiązaniem problemu będzie określenie, ile paczek gazet powinien dziennie zamówić sprzedawca, żeby zoptymalizować swoje zarobki.

Etap 2. Określenie modeli i pojęć

Ponieważ celem zadania jest znalezienie liczby paczek, którą trzeba zamówić, aby osiągnąć największy zysk, w pierwszej kolejności powinniśmy się zastanowić, jakie mamy możliwości.

Znamy dane notowane przez sprzedawcę, więc możemy założyć, że podobne wielkości będą dotyczyć sprzedaży w kolejnych dziesięciu dniach. Z danych zawartych w tabeli 1. wynika, że liczby paczek gazet sprzedanych w poszczególnych dniach należą do przedziału $\langle 11; 15\rangle$. Przyjmijmy zatem, że będziemy brać te wielkości pod uwagę przy rozwiązywaniu problemu. W pierwszym kroku przeanalizujemy zysk dla każdej możliwej kombinacji liczby paczek gazet zamówionych u dystrybutora i sprzedanych klientom. W tym celu utworzymy tabelę (tabela 2.).

		popyt – liczba paczek, które chcą kupić klienci				
		11	12	13	14	15
podaż – liczba paczek gazet zamówionych u dystrybutora	11					
	12					
	13					
	14					
	15					

Tabela 2. *Przykładowa tabela do obliczenia zysku dla różnych wariantów podaży i popytu*

Zastosowane oznaczenia:

- *c1* – cena, po której sprzedawca kupuje gazety u dystrybutora,
- *c2* – cena, po której sprzedawane są gazety klientom,
- *c3* – cena, po której można odsprzedać gazety dystrybutorowi,
- *popyt* – liczba paczek, które chcą kupić klienci,
- *podaż* – liczba paczek zamówionych u dystrybutora,
- *zysk jednostkowy* – zysk z jednej sprzedanej paczki gazet obliczony według wzoru: *zysk = c2 – c1,*
- *strata jednostkowa* – strata z jednej paczki gazet obliczona według wzoru: *strata = c1 – c3.*

Etap 3. Znalezienie rozwiązania

Aby rozpocząć rozwiązywanie tego problemu, należy zastanowić się, jakie mamy dane i jakie zależności zachodzą między nimi.

Znając ceny zakupu i sprzedaży oraz wartości popytu i podaży, możemy obliczyć dzienny zysk sprzedawcy (tabela 3.).

Zysk sprzedawcy obliczamy, mnożąc sprzedaną liczbę paczek przez *zysk jednostkowy* i odejmując stratę (iloczyn niesprzedanej liczby paczek oraz *straty jednostkowej*), jeśli taka występuje.

Chcemy teraz utworzyć wzór, dzięki któremu obliczymy zysk sprzedawcy dla różnych wielkości popytu i podaży.

Musimy jednak pamiętać, że nie jest możliwa sytuacja, w której sprzedawca sprzeda więcej gazet, niż zamówił. Dlatego przy obliczaniu zysku musimy przyjąć **mniejszą wartość** z wartości *podaży* i *popytu*.

Na przykład: Jeśli sprzedawca zamówił 14 paczek, a sprzedał 12, to zysk obliczymy, biorąc mniejszą z tych wartości, czyli 12. Natomiast pozostałe 2 paczki musiał odsprzedać i poniósł stratę w wysokości 100 zł, czyli zysk sprzedawcy wyniósł tego dnia $12 \cdot (180 - 100)$ zł $- (14 - 12) \cdot (100 - 50)$ zł $= 860$ zł (tabela 3.).

Jeśli natomiast sprzedawca zamówił 12 paczek, a klienci chcieli kupić 15 paczek, to i tak sprzedał tylko 12, czyli jego zysk wyniósł $12 \cdot (180 - 100)$ zł $= 960$ zł (tabela 3.).

Zysk sprzedawcy = 12 · 80zł = 960zł

Zysk sprzedawcy = 12 · 80zł - (14-12) · 50zł = 860zł

		popyt – liczba paczek, które chcą kupić klienci				
		11	12	13	14	15
podaż – liczba paczek gazet zamówionych u dystrybutora	11	880,00 zł	880,00 zł	880,00 zł	880,00 zł	880,00 zł
	12	830,00 zł	960,00 zł	960,00 zł	960,00 zł	960,00 zł
	13	780,00 zł	910,00 zł	1 040,00 zł	1 040,00 zł	1 040,00 zł
	14	730,00 zł	860,00 zł	990,00 zł	1 120,00 zł	1 120,00 zł
	15	680,00 zł	810,00 zł	940,00 zł	1 070,00 zł	1 200,00 zł

Tabela 3. *Wyliczenia dziennego zysku sprzedawcy dla wybranych wartości podaży i popytu*

Zatem w przypadku, gdy wszystkie gazety zostały sprzedane, czyli *podaż* ≤ *popyt*, jako wartość straty należy przyjąć zero.

Wzór na policzenie zysku sprzedawcy można przedstawić następująco:

zysk sprzedawcy =

= MIN(podaż; popyt) · zysk jednostkowy – MAX(0; podaż – popyt) · strata jednostkowa.

Sama informacja dotycząca zysku dla poszczególnych wartości popytu i podaży nie pozwoli nam jednak na określenie, jakie zamówienie złożyć w przyszłości. Największy zysk osiągniemy przy największej sprzedaży (15 paczek), ale niestety nie możemy zagwarantować takiej sytuacji.

Potwierdzają to dane z tabeli 1. Sprzedaż 15 paczek zdarzyła się tylko raz w ostatnich dziesięciu dniach, co oznacza, że jest ona mało prawdopodobna. Dokładniej, prawdopodobieństwo wystąpienia takiej sprzedaży w przyszłości wynosi 1/10. Natomiast sprzedaż 13 paczek wystąpiła 4 razy, co daje nam prawdopodobieństwo 4/10.

Możemy obliczyć prawdopodobieństwo wystąpienia prognozowanych wartości popytu:

liczba paczek	11	12	13	14	15
prawdopodobieństwo wystąpienia	0,1	0,3	0,4	0,1	0,1

Tabela 4. *Prawdopodobieństwo wystąpienia prognozowanych wartości popytu*

Na tej podstawie możemy obliczyć oczekiwany zysk w przyszłości, mnożąc zysk dla poszczególnych wartości sprzedaży oraz prawdopodobieństwo wystąpienia takiej sprzedaży. Największa wartość oczekiwanego zysku wskaże liczbę paczek gazet najbardziej korzystną do zamawiania.

Etap 4. Zaprogramowanie i testowanie rozwiązania

Narzędziem, które może nam pomóc w obliczeniach, jest arkusz kalkulacyjny. Do komórek tabeli 2. należy wpisać formuły, które obliczą zysk dla odpowiednich wartości z kolumn i wierszy. Następnie każdą z tych wartości należy pomnożyć przez wartość prawdopodobieństwa wystąpienia danego popytu i zsumować oczekiwany zysk dla danej liczby zamówionych paczek (rys. 13.).

W pierwszej kolejności należy wpisać wartości *c1*, *c2* i *c3* do odpowiednich komórek arkusza oraz wyliczyć *zysk jednostkowy* i *stratę jednostkową*. Następnie trzeba utworzyć tabelę na wzór tabeli 3. oraz dodać do niej wiersz określający prawdopodobieństwo wystąpienia prognozowanych wartości popytu.

Do obliczenia oczekiwanego dziennego zysku wykorzystamy wzór:

zysk sprzedawcy =

= MIN(podaż; popyt) · zysk jednostkowy – MAX(0; podaż – popyt) · strata jednostkowa.

W arkuszu kalkulacyjnym do obliczeń możemy wykorzystać funkcje MIN i MAX.

Następnie należy zysk sprzedawcy pomnożyć przez wartość prawdopodobieństwa wystąpienia danej wartości sprzedaży. W tym celu w komórce *C9* należy wpisać formułę:

=(MIN(C$8;$B9)*$C$2-MAX(0;$B9-C$8)*$C$3)*C$7

i skopiować ją do odpowiednich komórek w kolumnach *C:G*.

C9 =(MIN(C$8;$B9)*$C$2-MAX(0;$B9-C$8)*$C$3)*C$7

	A	B	C	D	E	F	G	H
1								
2	180,00 zł	c2	80,00 zł	zysk jednostkowy				
3	100,00 zł	c1	50,00 zł	strata jednostkowa				
4	50,00 zł	c3						
5								
6			popyt - liczba paczek, które chcą kupić klienci					
7	prawdopodobieństwo		0,1	0,2	0,4	0,2	0,1	Oczekiwany zysk
8			11	12	13	14	15	dzienny [SUMA]
9	podaż - liczba paczek gazet zamówionych u dystrybutora	11	88,00 zł	176,00 zł	352,00 zł	176,00 zł	88,00 zł	880,00 zł
10		12	83,00 zł	192,00 zł	384,00 zł	192,00 zł	96,00 zł	947,00 zł
11		13	78,00 zł	182,00 zł	416,00 zł	208,00 zł	104,00 zł	988,00 zł
12		14	73,00 zł	172,00 zł	396,00 zł	224,00 zł	112,00 zł	977,00 zł
13		15	68,00 zł	162,00 zł	376,00 zł	214,00 zł	120,00 zł	940,00 zł
14								

Rys. 13. *Rozwiązanie problemu w arkuszu kalkulacyjnym*

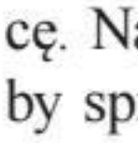
Na rysunku 13. widzimy, że największy oczekiwany zysk dzienny (988 zł) występuje dla podaży wynoszącej 13 paczek gazet. Takie zamówienie sprzedawca powinien złożyć u dystrybutora.

Ćwiczenie 21. Realizujemy rozwiązanie problemu „Sprzedawcy gazet" w arkuszu kalkulacyjnym

1. W arkuszu kalkulacyjnym utwórz tabelę, umieść dane i odpowiednie formuły, korzystając z opisanych wskazówek.

2. Przetestuj rozwiązanie, zmieniając dane wejściowe na inne – według własnego pomysłu. Co zauważasz? Kiedy sprzedawca ma największy zysk (przy jakiej wielkości podaży i popytu)? Skomentuj otrzymywane rezultaty dla nowych danych.
3. Zapisz plik pod nazwą *Sprzedawca*.

Model oraz sposób rozwiązania problemu sprzedawcy gazet można wykorzystać w przypadku handlowania innymi towarami, a także do planowania produkcji.



Ćwiczenie 22. Korzystamy ze zdobytych doświadczeń do rozwiązania podobnego problemu

1. Korzystając z analizy i rozwiązania problemu „Sprzedawcy gazet", określ własny, podobny problem i przygotuj jego rozwiązanie w arkuszu kalkulacyjnym.
2. Zapisz plik pod nazwą określającą problem.

Testowanie rozwiązania problemu „Sprzedawcy gazet" z wykorzystaniem rozwiązania problemu „Skrzyżowanie ulic"

Sposobem na przetestowanie naszego rozwiązania problemu „Sprzedawcy gazet" jest symulacja liczby sprzedanych gazet. Możemy tu wykorzystać te same narzędzia i sposób rozumowania, co w przypadku problemu „Skrzyżowanie ulic".

Jako dane empiryczne należy wtedy przyjąć dane zanotowane przez sprzedawcę. Następnie należy wykonać symulację zysku ze sprzedaży w zależności od liczby sprzedanych paczek gazet (rys. 14.). Wartością szukaną jest liczba paczek gazet, którą trzeba zamówić u dystrybutora, więc będzie to wartość wpisywana przez nas z klawiatury.

Zysk sprzedawcy obliczamy, wykorzystując wzór zawierający funkcje MIN i MAX, podając jako *popyt* wylosowaną wartość z odpowiedniej komórki w kolumnie *E*, a jako *podaż* – wartość wpisaną z klawiatury do tabeli z danymi.

E3 =WYSZUKAJ.PIONOWO(LOS.ZAKR(1;10);$A$3:$B$12;2;0)

	A	B	C	D	E	F	G	H	I	J	K
1	Dane empiryczne			Symulacja				Dane			
2	Dzień	Liczba sprzedanych paczek gazet		Dzień	Liczba sprzedanych paczek gazet	Zysk sprzedawcy		podaż - liczba paczek zamówionych w hurtowni	13		
3	1	12		1	13	1040		c1	100	zł	
4	2	13		2	12	910		c2	180	zł	
5	3	11		3	12	910		c3	50	zł	
6	4	12		4	13	1040		zysk jednostkowy	80	zł	
7	5	13		5	13	1040		strata jednostkowa	50	zł	
8	6	13		6	12	910					
9	7	12		7	13	1040					
10	8	15		8	15	1040					
11	9	13		9	12	910					
12	10	14		10	13	1040					
13											
14					ŚREDNIA	988					
15											

Rys. 14. *Przykładowa symulacja dla podaży 13 paczek gazet – średni zysk dzienny wynosi 988 zł*

F3 | =MIN(E3;$I$2)*$I$6-MAX(0;$I$2-E3)*$I$7

	A	B	C	D	E	F	G	H	I	J
1	Dane empiryczne			Symulacja				Dane		
2	Dzień	Liczba sprzedanych paczek gazet		Dzień	Liczba sprzedanych paczek gazet	Zysk sprzedawcy		podaż - liczba paczek zamówionych w hurtowni	15	
3	1	12		1	12	810		c1	100	zł
4	2	13		2	12	810		c2	180	zł
5	3	11		3	15	1200		c3	50	zł
6	4	12		4	12	810		zysk jednostkowy	80	zł
7	5	13		5	13	940		strata jednostkowa	50	zł
8	6	13		6	13	940				
9	7	12		7	13	940				
10	8	15		8	15	1200				
11	9	13		9	12	810				
12	10	14		10	13	940				
13										
14					ŚREDNIA	940				

Rys. 15. *Przykładowa symulacja dla podaży 15 paczek gazet – średni zysk dzienny wynosi 940 zł*

Ćwiczenie 23. Testujemy rozwiązanie „Sprzedawcy gazet", korzystając z rozwiązania problemu „Skrzyżowanie ulic"

1. Korzystając z rozwiązania problemu „Skrzyżowanie ulic", przygotuj symulację liczby sprzedanych gazet.
2. Zmieniając liczbę paczek zamówionych u dystrybutora, a następnie naciskając klawisz **F9**, obserwuj, jak zmienia się symulowana wartość zysku sprzedawcy z dziesięciu dni (rys. 15.).
3. Zapisz plik pod nazwą *Sprzedawca_symulacja*.

Problem 3. Klimat

Etap 1. Określenie problemu

Chcemy sprawdzić, czy w stolicy naszego województwa w ciągu dwudziestu ostatnich lat pojawiły się zauważalne zmiany średnich temperatur powietrza w poszczególnych miesiącach i latach, które mogą być związane ze zjawiskiem tzw. ocieplania klimatu.

Danych do rozwiązania problemu będziemy szukać w Internecie.

W rezultacie rozwiązania tego problemu mamy otrzymać wykres (wykresy) pokazujący, jaki jest trend zmian temperatury w ostatnich dwudziestu latach.

Etap 2. Określenie modeli i pojęć

W modelu uwzględnimy wartości średnich temperatur w każdym z dwunastu miesięcy w latach od 2000 do 2019. Będziemy posługiwać się następującymi pojęciami:

- *średnia temperatura miesięczna* – oznaczająca średnią temperaturę powietrza w danym miesiącu,
- *średnia temperatura roczna* – oznaczająca średnią temperaturę powietrza w danym roku.

Etap 3. Znalezienie rozwiązania

Zebrane dane najlepiej umieścić w tabeli. Dodatkowo dane mamy przedstawić na wykresie, który poddamy analizie. Najlepszym narzędziem do rozwiązania problemu będzie arkusz kalkulacyjny, który umożliwia nie tylko gromadzenie danych, ale i przedstawianie ich na wykresie oraz wszechstronną analizę.

Po wprowadzeniu danych należy wykonać wykres (wykresy), dobierając odpowiedni typ wykresu (wykresów), umożliwiający przedstawienie tendencji wzrostowych lub spadkowych oraz linii trendu.

Etap 4. Zaprogramowanie i testowanie rozwiązania

Zaczniemy od wprowadzenia danych do tabeli arkusza kalkulacyjnego. Na rysunku 16. pokazano projekt tabeli z jednym wierszem danych – *średnich temperatur* dla Wrocławia w roku 2000. Do obliczenia rocznej średniej temperatury użyjemy funkcji ŚREDNIA.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Średnie temperatury miesięczne i roczne od 2000 do 2019 roku dla Wrocławia**													
2	**rok**	**średnia temperatura miesięczna**												**średnia temperatura roczna**
3		**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	**IX**	**X**	**XI**	**XII**	
4	2000	0,4	3,8	5,1	12,1	15,6	17,9	16,5	19	13,4	12,4	6,8	2,4	10,5

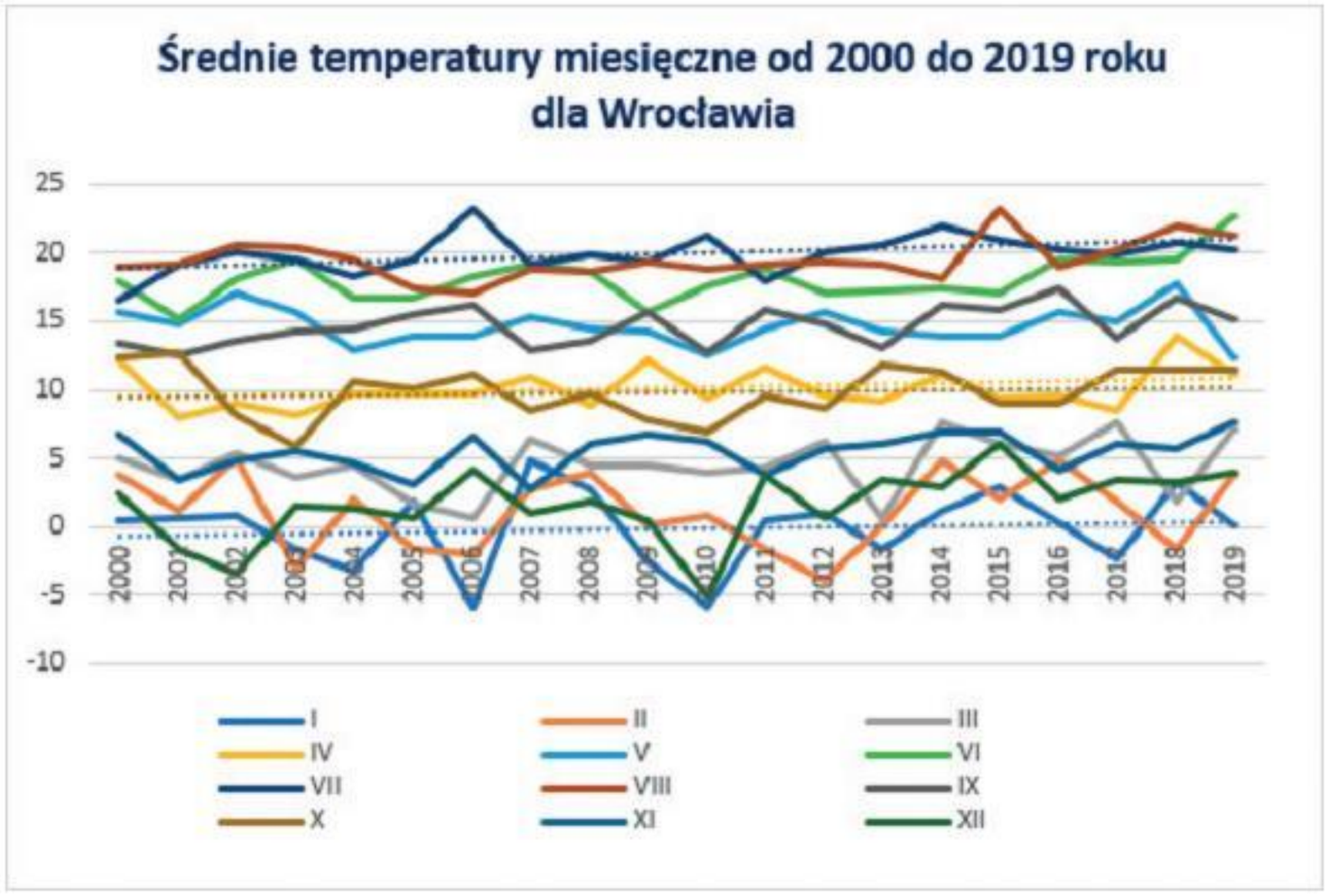
Rys. 16. *Przykładowe średnie temperatury miesięczne dla Wrocławia w roku 2000*

Ćwiczenie 24. Realizujemy rozwiązanie problemu „Klimat" w arkuszu kalkulacyjnym

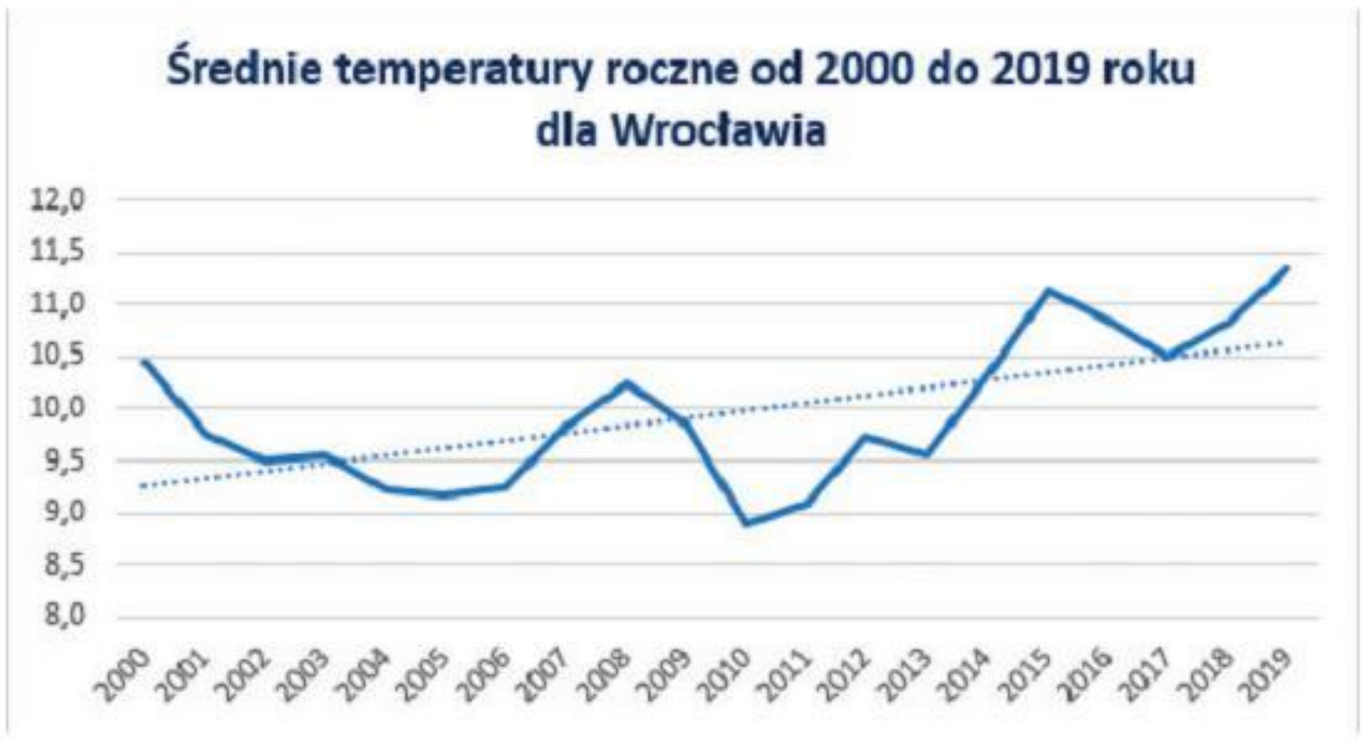
1. Znajdź w Internecie odpowiednie dane dla stolicy twojego województwa z ostatnich dwudziestu lat.
2. W arkuszu kalkulacyjnym utwórz tabelę podobną do pokazanej na rysunku 16. i umieść w niej dane dla wszystkich miesięcy i kolejnych lat.
3. Dodaj również obliczenia średniej temperatury rocznej dla każdego roku.
4. Zapisz plik pod nazwą *Temperatury*.

Aby zobaczyć, jakie są tendencje zmian średniej temperatury na przełomie dwudziestu lat, przygotujemy dwa wykresy: wykres *średnich temperatur miesięcznych* od 2000 do 2019 roku i wykres *średnich temperatur rocznych* od 2000 do 2019 roku.

Aby sprawdzić, jaki jest trend, można skorzystać z możliwości dodawania linii trendu dostępnej w arkuszu kalkulacyjnym (z menu kontekstowego linii wykresu należy wybrać polecenie dodawania linii trendu). Przykłady wykresów pokazano na rysunkach 17. i 18. Linię trendu dodano dla miesięcy: styczeń, kwiecień, lipiec i październik.



Rys. 17. *Przykładowy wykres średnich temperatur miesięcznych dla Wrocławia od 2000 do 2019 roku*



Rys. 18. *Przykładowy wykres średnich rocznych temperatur dla Wrocławia od 2000 do 2019 roku*

Ćwiczenie 25. Przygotowujemy i analizujemy wykresy

1. Dla danych zgromadzonych w tabeli arkusza kalkulacyjnego w ćwiczeniu 24. przygotuj wykresy podobne do pokazanych na rysunkach 17. i 18.
2. Dodaj odpowiednio linie trendu. Jaki zauważasz trend zmian temperatury w swoim mieście wojewódzkim w ostatnich dwudziestu latach? Podaj swoje wnioski.
3. Dodatkowo znajdź minimalną i maksymalną średnią roczną temperaturę. Podaj lata, w których one wystąpiły.
4. Zapisz plik pod tą samą nazwą.

Wskazówka: Jeśli chcesz analizować dane tylko dla wybranego miesiąca (wybranych miesięcy), na wykresie z rysunku 17. możesz ukryć niektóre kolumny tabeli arkusza kalkulacyjnego dla wybranego miesiąca (wybranych miesięcy).

Ćwiczenie 26. Korzystamy z rozwiązania problemu do rozwiązywania innego podobnego problemu

1. Zastosuj rozwiązanie problemu „Klimat" do sprawdzenia tendencji średnich miesięcznych i rocznych opadów deszczu w swoim mieście wojewódzkim w ostatnich dwudziestu latach. Znajdź odpowiednie dane w Internecie.
2. Jaki zauważasz trend zmian opadów deszczu w swoim mieście wojewódzkim w ostatnich dwudziestu latach? Podaj swoje wnioski.
3. Zapisz plik pod nazwą *Opady*.

Warto zapamiętać

- Arkusz kalkulacyjny umożliwia wykonywanie obliczeń i przedstawianie danych oraz wyników obliczeń w postaci wykresów.
- Obliczenia w arkuszu kalkulacyjnym przeprowadzamy, tworząc formuły. Danymi w formułach mogą być m.in.: dowolne liczby (stałe), wartości komórek arkusza (zmienne) lub wartości funkcji.
- W arkuszu kalkulacyjnym można ustalić wybrany format wyświetlanych danych, m.in.: **Liczbowy**, **Procentowy**, **Daty** i **Czasu**.
- Arkusz kalkulacyjny umożliwia formatowanie komórek i danych umieszczonych w komórkach. W celu ułatwienia analizy danych można wyróżnić komórki poprzez zastosowanie formatowania warunkowego.
- W arkuszu kalkulacyjnym występuje kilka kategorii funkcji, które można wykorzystywać w formułach, m.in.: funkcje matematyczne, logiczne, statystyczne oraz daty i czasu.
- Przed utworzeniem wykresu trzeba wybrać jego typ, który w optymalny sposób będzie prezentował dane.

Pytania i polecenia

1. Do czego służy arkusz kalkulacyjny?
2. W jaki sposób buduje się formuły w arkuszu kalkulacyjnym?
3. Jakie dane można umieszczać w komórkach arkusza kalkulacyjnego?
4. Jakie znasz formaty danych liczbowych? W jaki sposób można zmienić format danych?
5. Jakie znasz możliwości formatowania tabeli? Na czym polega formatowanie warunkowe?
6. W jaki sposób można użyć w formule funkcji? Omów na wybranym przykładzie.
7. Do czego służy funkcja JEŻELI? Podaj przykład jej zastosowania.
8. Omów na przykładzie zastosowanie funkcji LICZ.JEŻELI.
9. Jakiej funkcji użyjesz do obliczenia reszty z dzielenia liczby całkowitej *a* przez liczbę całkowitą *b*, a jakiej do obliczenia wartości bezwzględnej liczby rzeczywistej *c*? Podaj przykłady.
10. Do czego służy funkcja LOS? Podaj przykład jej zastosowania.
11. Omów na przykładzie, w jaki sposób można obliczyć liczbę dni dzielących od siebie dwie daty.

12. Omów podstawowe typy wykresów oraz ich zastosowania. Od czego zależy użycie danego typu wykresu?
13. W jaki sposób można narysować wykres funkcji sinus za pomocą arkusza kalkulacyjnego?
14. W jaki sposób można narysować wykres funkcji liniowej?

Zadania

1. Znajdź w Internecie informacje na temat powierzchni i liczby ludności każdego województwa w Polsce. Utwórz tabelę, w której umieścisz zebrane dane, oraz formułę obliczającą gęstość zaludnienia w każdym województwie. Przedstaw wyniki obliczeń na wykresie (dobierz typ wykresu, uzasadnij wybór). Zapisz plik pod nazwą *Gęstość zaludnienia*.
2. W arkuszu kalkulacyjnym utwórz tabelę zawierającą tabliczkę mnożenia liczb od 1 do 25. Zapisz plik pod nazwą *Tabliczka mnożenia.*
3. Utwórz tabelę, w której zgromadzisz dane o liczbie ludności w krajach Unii Europejskiej w minionym roku. Skorzystaj z Internetu lub innych źródeł. Korzystając z dostępnych funkcji arkusza kalkulacyjnego, oblicz łączną i średnią liczbę ludności, podaj też najmniejszą i największą liczbę ludności. Zapisz plik pod nazwą *Ludność UE*.
4. Otwórz plik *Ludność UE,* zapisany w zadaniu 3. Stosując formatowanie warunkowe, odróżnij kolorystycznie komórkę z najwyższą liczbą ludności od komórki z najniższą liczbą ludności. Zapisz plik pod tą samą nazwą.
5. Przygotuj w arkuszu kalkulacyjnym tabelę przedstawioną na rys. 19. Zaktualizuj dane, korzystając z Internetu lub innych źródeł. Dodaj formułę obliczającą sumaryczną powierzchnię użytkowanych gruntów. Utwórz wykres, który zobrazuje udział poszczególnych sposobów użytkowania gruntów w stosunku do całości. Zapisz plik pod nazwą *Grunty*.

Polska	Grunty orne	Łąki i pastwiska	Lasy	Pozostałe grunty i wody
w procentach	46,6	12,8	28,1	12,5
w tys. km	145,72	40,03	87,87	39,09

Rys. 19. *Tabela – zadanie 5.*

6. Zaprojektuj arkusz pozwalający śledzić miesięczne wydatki z domowego budżetu. Wydatki podziel na kategorie, np. żywność, rachunki itp. Zapisz plik pod nazwą *Budżet domowy*.
7. W pliku *Wyniki* zapisanym w ćwiczeniu 16. w odpowiednich komórkach umieść formuły obliczające, ilu uczniów zdało egzamin, a ilu uczniów go nie zdało. Skorzystaj z odpowiedniej funkcji arkusza. Zadbaj o umieszczenie opisu zawartości komórek. Zapisz plik pod tą samą nazwą.
8. Korzystając z odpowiedniej funkcji arkusza kalkulacyjnego, podaj swój wiek: w dniach, godzinach, minutach i sekundach.
9. W arkuszu kalkulacyjnym przygotuj tabelę i utwórz wykres:
   a. funkcji liniowej $y = -2x - 2,5$;
   b. funkcji kwadratowej $y = x^2 + 2x + 1$;
   c. funkcji liniowej $y = ax + b$ dla dowolnych współczynników a i b;
   d. funkcji tangens.

Dla zainteresowanych

10. Podatek dochodowy od osób fizycznych za 2019 rok płacono według następującej skali:

Podstawa obliczenia podatku w złotych		Podatek wynosi
dochód ponad	dochód do	
	85 528	17,75 % minus kwota zmniejszająca podatek
85 528		15 181 zł 22 gr + 32 % nadwyżki ponad 85 528 zł minus kwota zmniejszająca podatek
1 000 000		15 181 zł 22 gr + 32 % nadwyżki ponad 85 528 zł minus kwota zmniejszająca podatek plus dodatkowe 4% od nadwyżki ponad 1 mln zł podstawy

Utwórz arkusz pozwalający na obliczenie podatku. Korzystając z Internetu, zaktualizuj dane. Dokonaj zaokrąglenia kwoty podatku do pełnych złotych. Jaki podatek należałoby zapłacić od następujących rocznych kwot dochodów: 15 000 zł, 25 000 zł, 60 000 zł, 79 000 zł, 100 000 zł, 2 000 000 zł (przy założeniu, że kwota zmniejszająca podatek jest równa 0)?

Wskazówka: Skorzystaj z możliwości zagnieżdżania funkcji JEŻELI.

11. Rozwiąż równanie $x^2 + 2x + 1 = 0$, korzystając z narzędzi udostępnianych przez arkusz kalkulacyjny.
12. Zaprojektuj tabelę do wprowadzania własnych ocen śródrocznych i końcowych ze wszystkich przedmiotów przez cztery lata (osiem semestrów). Utwórz formuły obliczające:
    a. średnie ocen na koniec każdego semestru i na koniec każdego roku szkolnego;
    b. minimalną i maksymalną średnią;
    c. liczby poszczególnych ocen w każdym roku szkolnym i na koniec czterech lat nauki, np. liczba piątek w I klasie, II klasie, III klasie, IV klasie, łączna liczba piątek (po czterech latach), liczba czwórek itd. Ustal odpowiednie formaty danych i sformatuj tabelę tak, aby była czytelna.

Temat B2. Filtry oraz tabele i wykresy przestawne w arkuszu kalkulacyjnym

1. Korzystanie z filtrów
2. Tabele i wykresy przestawne
3. Stosowanie filtru w tabeli przestawnej
4. Praktyczny poradnik

Warto powtórzyć

1. Jakie dane można umieszczać w komórkach arkusza kalkulacyjnego?
2. W jaki sposób buduje się formuły w arkuszu kalkulacyjnym?
3. Jakie są podstawowe typy wykresów?
4. Kiedy stosuje się wykres kolumnowy?

1. Korzystanie z filtrów

W tabeli ocen (rys. 1.) chcemy wyświetlić nazwiska uczniów, którzy z fizyki i chemii otrzymali co najmniej piątki. Ręczne wykonanie takiej operacji przy dużej liczbie uczniów byłoby bardzo pracochłonne. Jak to zrobić w arkuszu kalkulacyjnym?

Program Microsoft Excel oferuje funkcję, która ułatwi wykonanie tego typu zadania – **filtry**. Przy stosowaniu opcji filtrowania ważne jest, aby kolumny w tabeli miały nagłówki, np. *nazwisko*, *imię*, *ulica*, *przedmiot*, *liczba sztuk*.

Przykład 1. Stosowanie filtrów

1. Zaznacz dowolną komórkę tabeli, np. *B5*.
2. Z menu (lub na karcie) **Dane** wybierz opcję filtrowania (**Filtr/Autofiltr** lub **Filtruj**). Obok tytułów kolumn pojawią się przyciski ze strzałkami (rys. 1.).
3. Kliknięcie którejkolwiek ze strzałek powoduje rozwinięcie listy z możliwościami wyboru różnych kryteriów filtrowania oraz alfabetycznej listy wszystkich unikalnych wartości występujących w wybranej kolumnie tabeli. Wybierając jedną z wartości, ograniczymy wyświetlanie do wierszy zawierających tę wartość w aktualnej kolumnie. Na przykład, jeżeli chcemy wyświetlić tylko nazwiska uczniów, którzy mają szóstkę z matematyki, z rozwijalnej listy obok tytułu kolumny z ocenami z matematyki wybieramy 6.
4. W naszym przykładzie skorzystamy z możliwości ustawiania niestandardowych filtrów (opcja **Filtry liczb** lub **Inne**, **Niestandardowe**). Dla kolumn *Fizyka* i *Chemia* w filtrze niestandardowym należy ustalić warunek **Większe niż lub równe 5**. Po ustaleniu kryteriów wiersze, które ich nie spełniają, są ukrywane.

B5 | Krzysztof

	A	B	C	D	E	F	G	H	I	J
1	Nazwisk	Imię	J. pols	Matematy	Fizyka	Biolog	Chemi	Geograf	J. angiel	Średn
2	Zając	Tomasz	6	5	6	5	5	5	6	5,43
3	Nowak	Agnieszka	4	3	5	6	4	4	5	4,43
4	Wrona	Bolesław	4	3	2	4	3	2	4	3,14
5	Kalarus	Krzysztof	3	2	4	5	4	4	2	3,43
6	Kminek	Franciszka	4	5	2	6	2	6	5	4,29
7	Krajewski	Kacper	6	5	5	4	5	2	4	4,43
8	Szymczak	Barbara	2	6	2	5	3	4	4	3,71
9	Grabowski	Jan	5	3	3	4	5	4	3	3,86
10	Janik	Ewa	5	4	5	4	4	5	5	4,57
11	Małyga	Jakub	3	2	2	6	3	4	2	3,14
12	Krawczyk	Maria	5	5	6	6	6	4	5	5,29
13	Stopa	Kasia	5	4	5	5	4	2	4	4,14
14	Kaczka	Stefan	2	6	2	3	2	2	2	2,71
15	Wysocki	Maciej	6	2	4	5	3	5	5	4,29
16	Wieczorek	Zofia	6	5	5	4	6	3	4	4,71
17	Kotlarska	Anna	4	3	4	6	2	4	6	4,14
18	Piechota	Radosław	5	3	4	4	4	6	3	4,14
19	Leszczyński	Tadeusz	4	4	5	5	3	5	4	4,29
20	Sołtys	Beata	3	2	4	6	4	5	5	4,14
21	Kogut	Marian	3	4	2	5	2	6	3	3,57
22										

Rys. 1. *Stosowanie filtrów – fragment przykładowej tabeli arkusza kalkulacyjnego z fikcyjnymi danymi*

Uwagi:

- Aby ponownie wyświetlić wszystkie wiersze, należy przy polach, dla których ustawiono filtry, zaznaczyć opcję wyświetlenia wszystkich wierszy lub opcję wyczyszczenia filtru.
- Aby wyłączyć widok filtru, należy wybrać tę samą opcję, która włącza ten widok (punkt 2.).

Ćwiczenie 1. Filtrujemy dane zapisane w tabeli arkusza kalkulacyjnego

1. Utwórz tabelę (podobną do pokazanej na rys. 1.) zawierającą fikcyjne nazwiska uczniów, ich oceny z poszczególnych przedmiotów oraz średnie tych ocen.
2. Zapisz plik pod nazwą *Uczniowie*.
3. Korzystając z przykładu 1., wyświetl nazwiska uczniów, którzy mają:
   a. z fizyki i chemii oceny wyższe lub równe 5,
   b. z matematyki ocenę najwyższą,
   c. średnią ocen większą lub równą 4,00,
   d. średnią ocen większą od 4,00, a mniejszą od 5,00.
   Uwaga: Przed wykonaniem kolejnego podpunktu (b-d) wyświetl ponownie wszystkie wiersze.
4. Nie zapisuj zmian.

Ćwiczenie 2. Filtrujemy dane zapisane w tabeli arkusza kalkulacyjnego

Otwórz plik *Uczniowie* zapisany w ćwiczeniu 1. Zastosuj filtr wyświetlający wiersze:
a. zawierające dane chłopców; zapisz plik pod nazwą *Uczniowie_chłopcy*,
b. zawierające dane dziewcząt; zapisz plik pod nazwą *Uczniowie_dziewczęta*.

Wskazówka: Imiona dziewcząt podane w ćwiczeniu kończą się literą a. Z rozwijanej listy znajdującej się obok pola *Imię* należy wybrać odpowiednio filtrowanie niestandardowe i ustawić kryterium **nie kończy się na *a*** lub **kończy się na *a***.

2. Tabele i wykresy przestawne

Do bardziej kompleksowej analizy (zwłaszcza dużej liczby) danych, wykonywania różnych (często skomplikowanych) obliczeń i podsumowań oraz ich wyświetlania, możemy wykorzystać **tabele przestawne**.

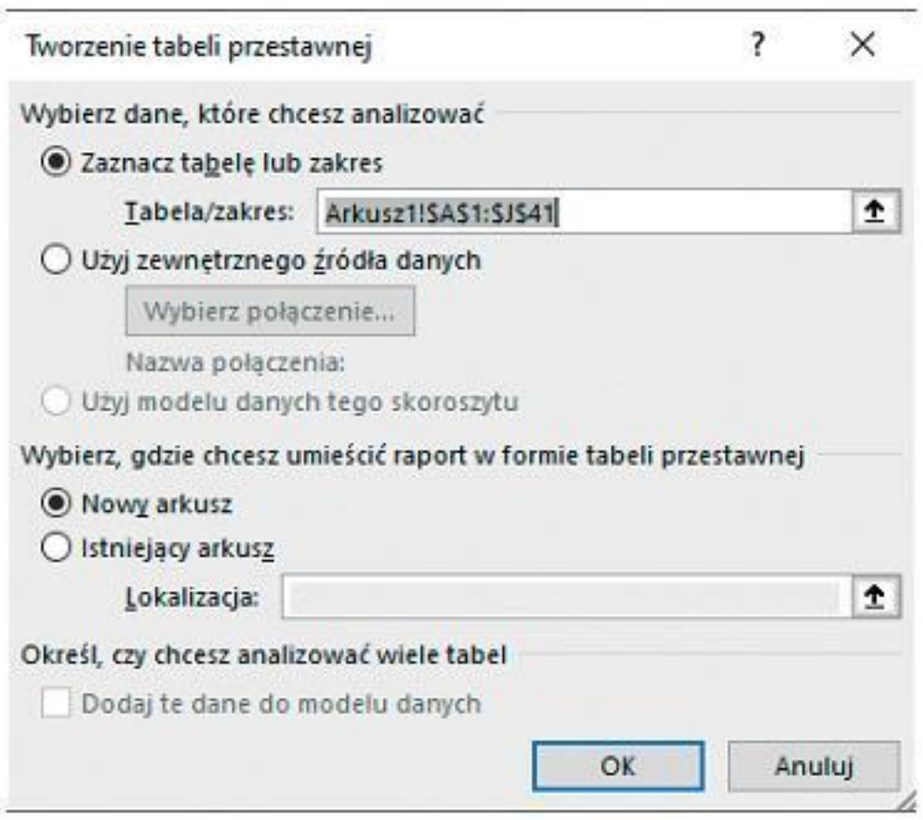
Aby lepiej zrozumieć ideę tabel przestawnych, zaczniemy od prostego przykładu. Skorzystamy z tabeli przygotowanej w ćwiczeniu 1.

Aby skorzystać z możliwości tabel przestawnych, musimy wcześniej odpowiednio przygotować tabelę arkusza kalkulacyjnego. Podobnie jak w przypadku filtrów, wszystkie kolumny muszą mieć nagłówki umieszczone w komórce nad kolumną danych. W przypadku tabel przestawnych nagłówki powinny mieć różne nazwy.

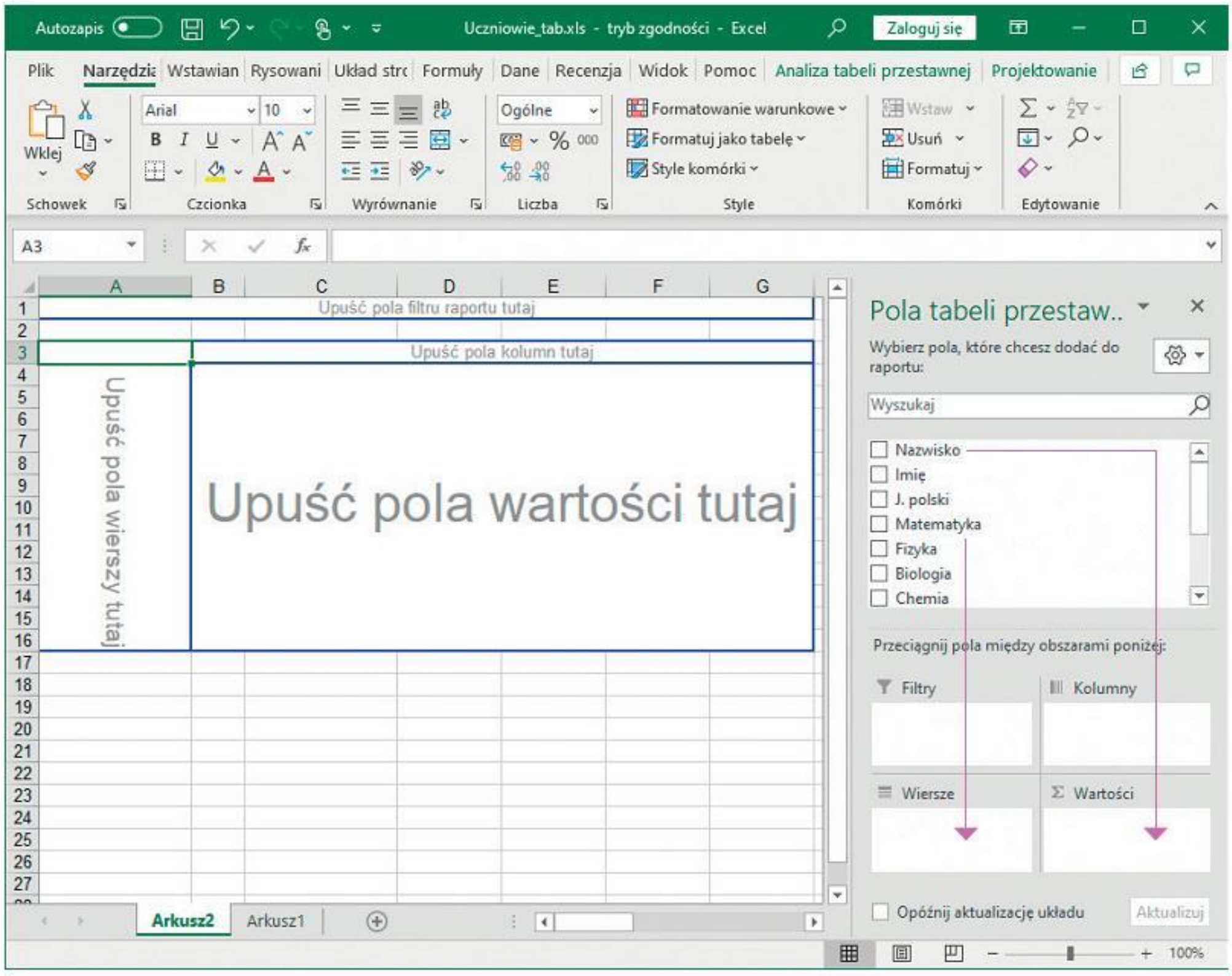
Przykład 2. Tworzenie tabeli przestawnej

Chcemy wiedzieć, ilu uczniów otrzymało poszczególne oceny (od dopuszczającej do celującej) z matematyki.

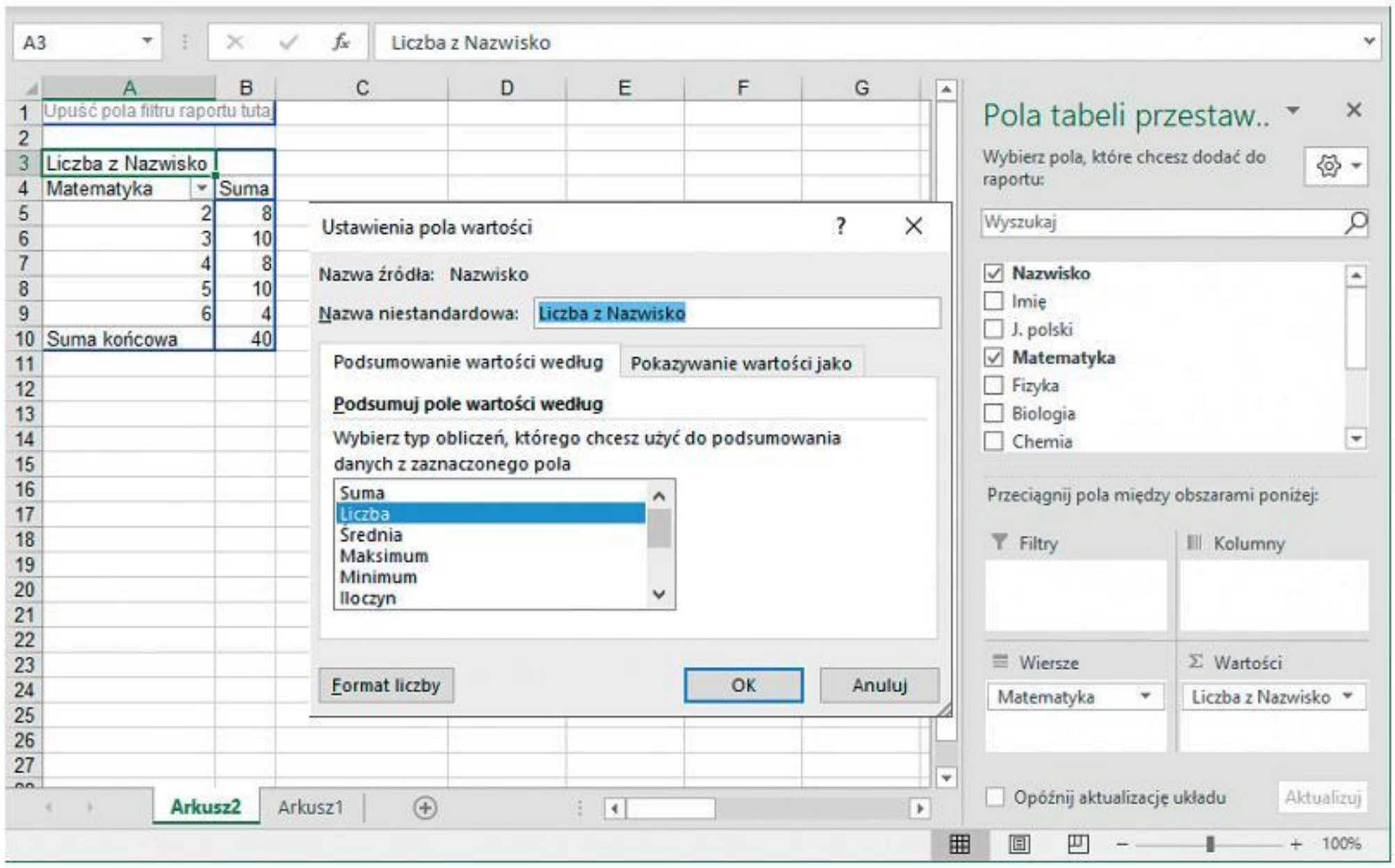
1. Zaznacz tabelę lub zakres danych (łącznie z nagłówkami), które chcesz analizować.
2. Z menu **Wstawianie** wybierz opcję **Tabela przestawna**.
3. W oknie **Tworzenie tabeli przestawnej** sprawdź zakres danych w polu **Zaznacz tabelę lub zakres** (rys. 2a), wybierz miejsce umieszczenia tabeli przestawnej (**Nowy arkusz**) i kliknij **OK**.
4. W nowym arkuszu po prawej stronie pojawi się panel **Pola tabeli przestawnej** z listą pól tabeli, które wybraliśmy do analizy. Przeciągnij odpowiednio pola, w naszym przypadku: *Matematyka* do obszaru **Wiersz** (jeśli dane mają być umieszczone w wierszach) i *Nazwisko* do obszaru **Wartości** (rys. 2b).
5. Kliknij pole *Nazwisko* i wybierz z listy rozwijanej polecenie **Ustawienia pola wartości** typ obliczeń *Liczba*. W tabeli pojawi się informacja o liczbie osób, które otrzymały poszczególne oceny (rys. 2c).



Rys. 2a. *Okno* ***Tworzenie tabeli przestawnej***



Rys. 2b. *Tworzenie tabeli przestawnej w nowym arkuszu – wybieranie pól*



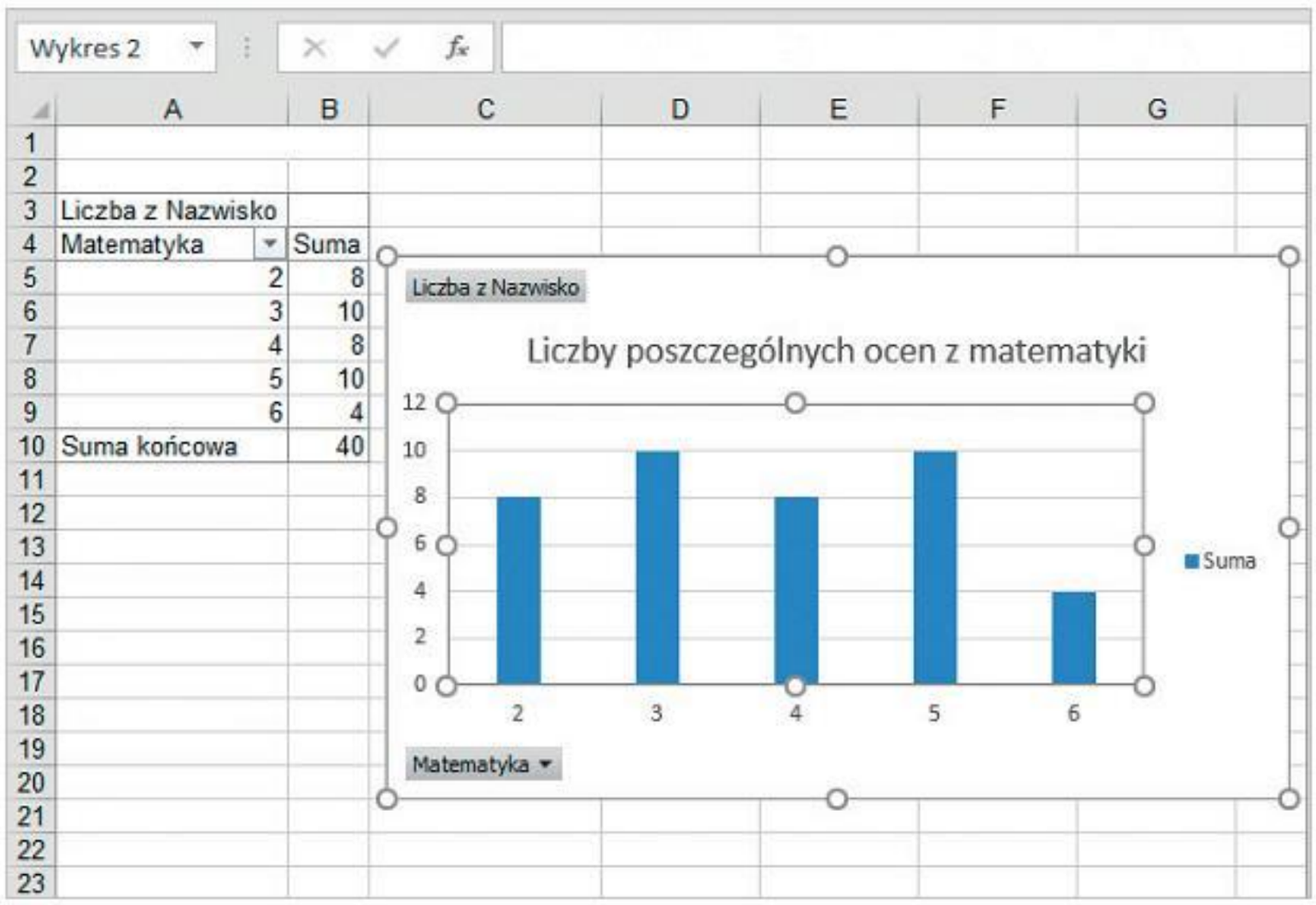
Rys. 2c. *Utworzona tabela przestawna – okno* ***Ustawienia pola wartości***

Podczas tworzenia tabel przestawnych nie zmienia się tabela źródłowa (układ danych i wartości komórek).

Ćwiczenie. 3. Tworzymy tabelę przestawną

1. Otwórz plik *Uczniowie* zapisany w ćwiczeniu 1.
2. Korzystając z przykładu 2., utwórz tabelę przestawną, przedstawiającą liczbę uczniów, którzy otrzymali poszczególne oceny (2, 3, 4, 5 i 6) z matematyki.
3. Zapisz plik pod nazwą *Uczniowie_tab*.

Aby utworzyć wykres przestawny, należy kliknąć dowolną komórkę tabeli i na karcie **Analiza tabeli przestawnej** wybrać polecenie **Wykres przestawny**.



Rys. 3. *Tabela i wykres przestawny – ćwiczenie 4.*

Ćwiczenie. 4. Tworzymy wykres przestawny

1. Otwórz plik *Uczniowie* zapisany w ćwiczeniu 3.
2. Utwórz wykres przestawny (kolumnowy) przedstawiający liczbę uczniów, którzy otrzymali poszczególne oceny z matematyki. Zmień tytuł wykresu na „Liczba poszczególnych ocen z matematyki”.
3. Zapisz plik pod tą samą nazwą.

Jeśli zmienimy dane w tabeli źródłowej, należy odświeżyć widok (**Analiza tabeli przestawnej/ Odśwież**) – wówczas dane w tabelach przestawnych również się odpowiednio zmienią.

Ćwiczenie. 5. Tworzymy tabelę i wykres przestawny

1. Otwórz plik *Uczniowie_tab* zapisany w ćwiczeniu 3.
2. W nowym arkuszu utwórz tabelę i wykres przestawny, przedstawiający liczbę uczniów, którzy otrzymali poszczególne oceny z języka polskiego.
3. Zapisz plik pod tą samą nazwą.

3. Stosowanie filtru w tabeli przestawnej

Tabela na rysunku 4. zawiera liczby godzin przepracowanych przez pracowników pewnej firmy w ciągu dwóch tygodni. Chcemy dla każdego pracownika obliczyć płacę w każdym tygodniu oraz mieć możliwość porównywania płacy w pierwszym i drugim tygodniu. Jak to zrobić w arkuszu kalkulacyjnym?

Ćwiczenie. 6. Przygotowujemy tabelę w arkuszu kalkulacyjnym

1. W arkuszu kalkulacyjnym utwórz tabelę pokazaną na rysunku 4. zawierającą fikcyjne dane.
2. Dla każdego pracownika oblicz sumę godzin i płacę w każdym tygodniu. Wykonaj obliczenia dla dwóch tygodni, wpisując tylko dwie formuły i kopiując je do pozostałych komórek.
3. Zapisz plik pod nazwą *Pracownicy*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		stawka za goodzinę	8,0													
3	LP	Pracownik	pon-01	wt-02	śr-03	czw-04	piąt-05	suma godzin - tydzień 1	płaca - tydzień 1	pon-08	wt-09	śr-10	czw-11	piąt-12	suma godzin - tydzień 2	płaca - tydzień 2
4	1	*Pracownik1*	7,5	8,0	9,5	6,0	8,0			7,5	8,5	9,5	5,0	8,0		
5	2	*Pracownik2*	8,0	7,5	8,0	9,0	8,0			6,5	8,5	9,5	6,0	8,0		
6	3	*Pracownik3*	8,5	8,5	8,5	8,0	6,5			8,0	6,5	9,0	9,5	7,5		
7	4	*Pracownik4*	9,5	6,5	7,0	8,0	9,0			9,0	9,0	8,5	7,5	7,5		
8	5	*Pracownik5*	9,0	7,5	7,5	8,0	8,5			8,5	8,5	8,0	7,5	7,5		
9	6	*Pracownik6*	8,5	8,0	9,5	7,5	5,0			8,0	7,5	7,5	8,0	8,0		
10	7	*Pracownik7*	9,5	9,5	10,0	9,5	6,6			8,5	8,0	9,5	9,5	6,0		
11	8	*Pracownik8*	10,0	6,3	8,5	9,0	5,0			8,5	8,0	8,0	7,5	8,0		
12	9	*Pracownik9*	8,5	8,5	8,5	8,5	7,8			8,0	8,0	7,5	7,5	8,0		
13	10	*Pracownik10*	7,5	7,5	6,5	6,0	9,5			5,5	5,5	9,5	7,5	6,0		
14	11	*Pracownik11*	7,5	7,5	6,5	8,0	9,5			8,0	5,5	6,5	7,5	8,0		
15	12	*Pracownik12*	8,0	8,0	6,0	9,0	10,0			8,5	7,5	6,0	7,0	8,0		
16	13	*Pracownik13*	5,5	5,5	8,5	9,0	8,5			7,9	10,0	10,0	10,0	6,3		
17	14	*Pracownik14*	7,2	8,0	6,8	9,0	8,5			9,5	8,5	8,5	8,5	8,0		
18	15	*Pracownik15*	7,5	7,5	7,9	9,0	8,1			7,9	8,0	9,0	8,0	9,0		
19	16	*Pracownik16*	9,5	9,5	9,5	9,0	7,5			8,0	8,0	9,5	9,5	9,5		
20	17	*Pracownik17*	9,0	8,0	9,0	8,2	7,5			7,9	7,5	9,0	7,0	8,0		
21	18	*Pracownik18*	9,0	8,0	9,0	8,0	8,5			8,0	8,5	8,0	8,0	8,0		
22	19	*Pracownik19*	6,5	8,0	9,0	9,0	6,5			8,0	8,0	7,9	8,2	8,0		
23	20	*Pracownik20*	8,5	7,5	9,0	8,0	8,0			8,0	7,5	7,5	7,5	8,5		

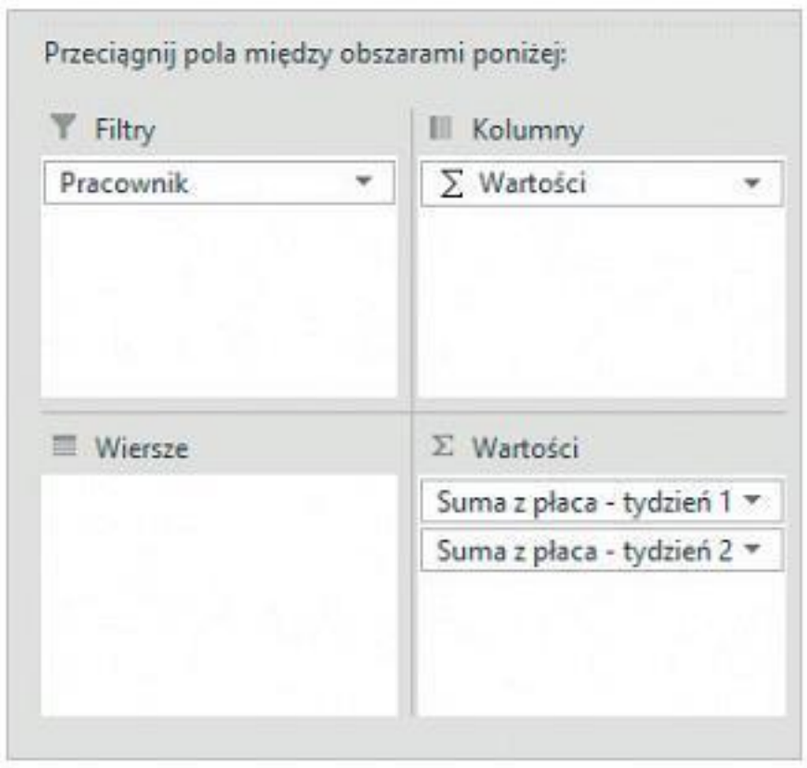
Arkusz1 | zestawienie dla menedżera | wykres | Arkusz3 ...

Rys. 4. *Tabela arkusza kalkulacyjnego z fikcyjnymi danymi – ćwiczenie 6.*

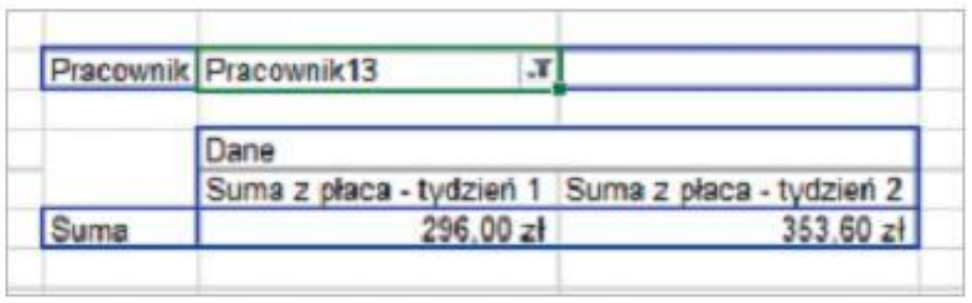
Przykład 3. Stosowanie filtru w tabeli przestawnej

Naszym zadaniem jest wykonanie indywidualnych zestawień płacowych dla każdego pracownika. Aby przedstawić dane w podziale na tygodnie oraz mieć możliwość szybkiego wyboru pracownika, utworzymy tabelę przestawną i zastosujemy w niej filtrowanie.

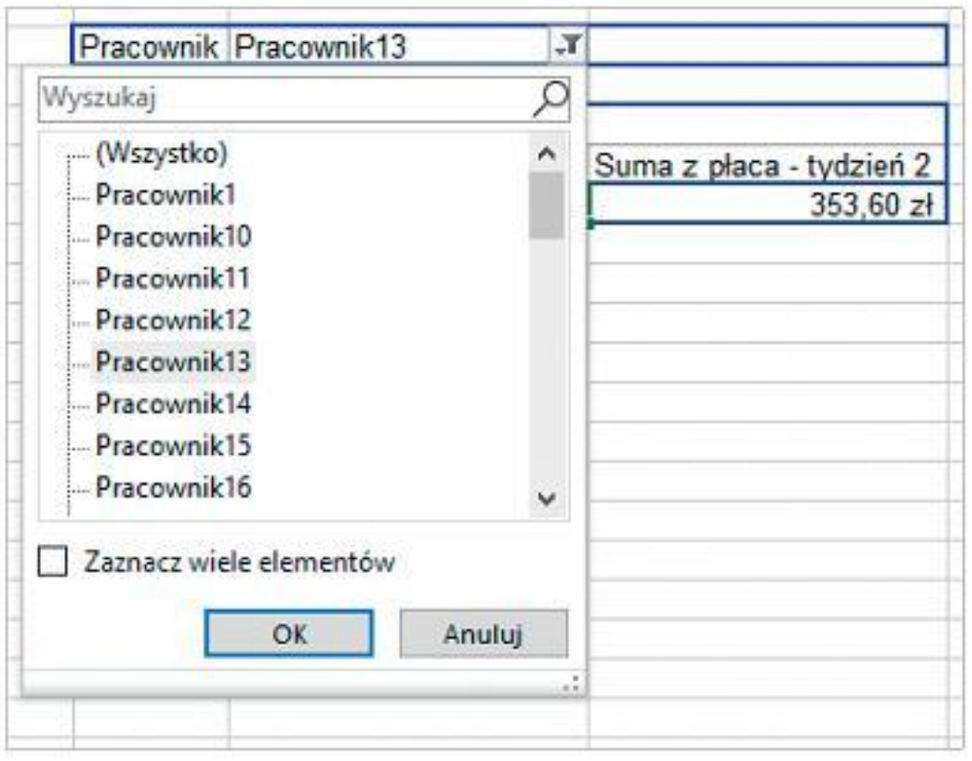
1. Zaznacz cały zakres danych, który będziemy analizować wraz z nagłówkami oraz wstaw tabelę przestawną (przykład 2.).
2. W panelu **Pola tabeli przestawnej** wybierz pola *płaca – tydzień 1* oraz *płaca – tydzień 2* i umieść je w obszarze **Wartości** (rys. 5a).
3. W **Ustawieniach pola wartości** zastosuj *Suma* jako typ obliczeń.
4. Wybierz pole *Pracownik* i umieść je w obszarze *Filtry* (rys. 5a).



Rys. 5a. *Widok* ***Pola tabeli przestawnej*** *– przykład 3.*



Rys. 5b. *Zastosowanie filtru dla pola* ***Pracownik***



Rys. 5c. *Wybór elementu z rozwijanej listy* ***Filtru*** *w tabeli przestawnej*

Ćwiczenie. 7. Stosujemy filtr w tabeli przestawnej

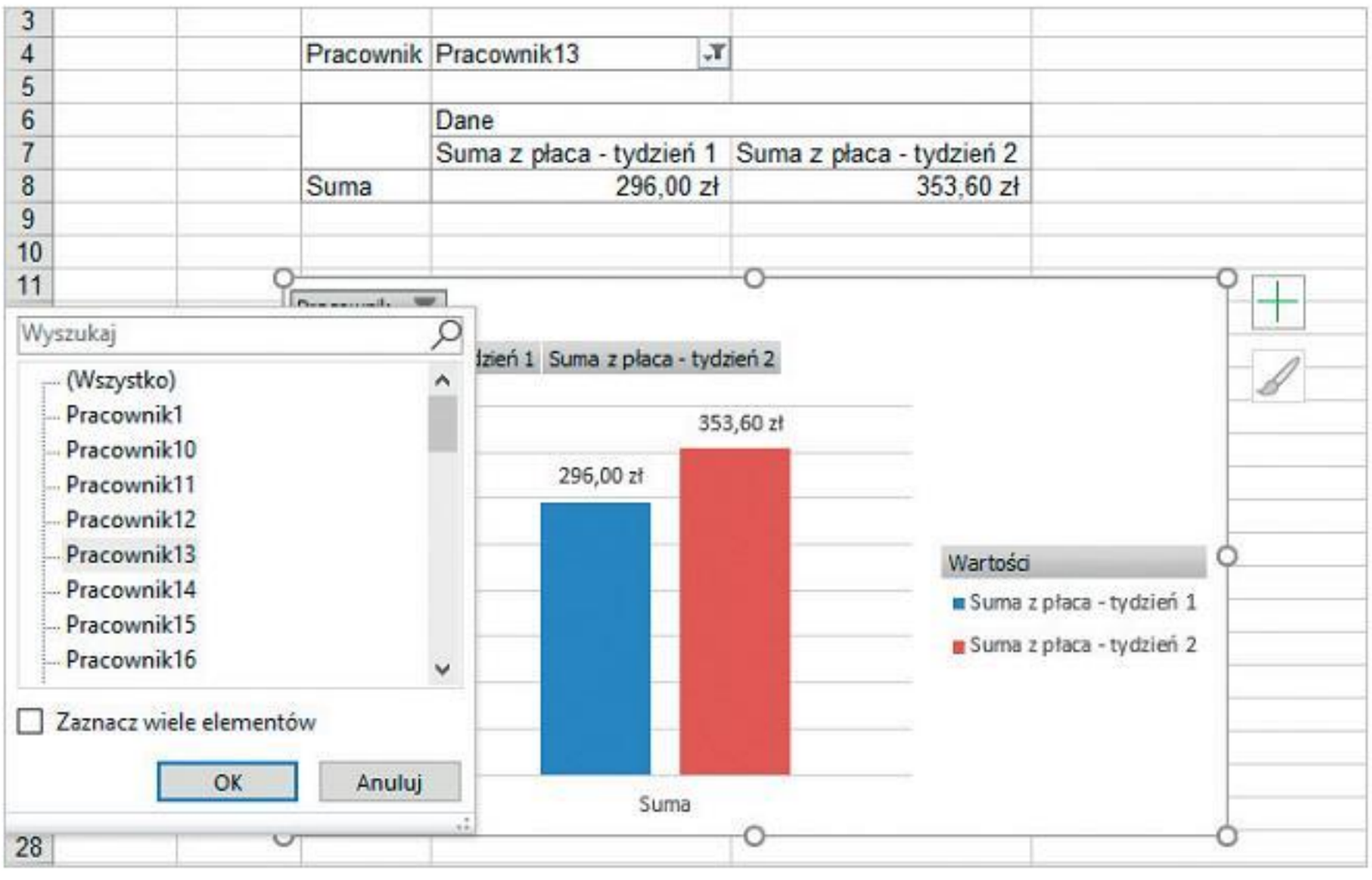
1. Otwórz plik *Pracownicy* zapisany w ćwiczeniu 6.
2. W nowym arkuszu utwórz tabelę przestawną, gdzie wartościami będą sumy wynagrodzenia za każdy tydzień pracy pracowników. Zastosuj filtr dla pola *Pracownik.*
3. Zapisz plik pod nazwą *Pracownicy _tab*.

Przykład 4. Wstawianie wykresu do tabeli przestawnej z filtrem

1. Kliknij w obszarze utworzonej przez siebie tabeli przestawnej.
2. Wybierz opcję **Analiza tabeli przestawnej/ Wykres przestawny**, a następnie – wykres kolumnowy.
3. Dodaj etykiety danych oraz sformatuj oś pionową, aby pokazywała wartości od 0 (rys. 6a).
4. Po naciśnięciu przycisku na wykresie można wybrać pracownika, dla którego zostanie utworzony wykres (rys. 6b).



Rys. 6a. *Wykres przestawny – przykład 4.*



Rys. 6b. *Wybór elementu z rozwijanej listy* ***Filtru*** *w wykresie przestawnym*

Ćwiczenie. 8. Tworzymy wykres przestawny

1. Otwórz plik *Pracownicy_tab* zapisany w ćwiczeniu 7.
2. Utwórz wykres przestawny do tabeli przestawnej.
3. Zapisz plik pod tą samą nazwą.

4. Praktyczny poradnik

Porada 1. Otwieranie dokumentów zawierających makra

Makra utworzone w arkuszu kalkulacyjnym zapisywane są w pliku wraz z danymi arkusza. Podczas otwierania takiego pliku pojawia się ostrzeżenie przed potencjalnym zagrożeniem.

Makra mogą zawierać wirusy niebezpieczne dla naszego komputera. Z tego powodu, otwierając pliki pochodzące od innych osób, powinniśmy zawsze wybierać opcję wyłączającą makra. Przed otwarciem w arkuszu kalkulacyjnym pliku zawierającego makra i pochodzącego od innej osoby powinniśmy sprawdzić (za pomocą programu antywirusowego), czy nie zawiera on wirusów. Dopiero potem możemy ponownie otworzyć dokument i włączyć makra.

Porada 2. Zasady adresowania w arkuszu kalkulacyjnym

Praca w arkuszu kalkulacyjnym opiera się na **adresowaniu względnym**. Po skopiowaniu formuły adresy komórek zawarte w formule zmieniają się względem komórki zawierającej tę formułę.

Jeśli chcemy odwołać się do określonej komórki, której adres nie będzie się zmieniał podczas kopiowania formuły, stosujemy **adresowanie bezwzględne.** Do adresu komórki występującej w danej formule dodajemy znaki $ zarówno przed symbolem kolumny (blokując w ten sposób odwołanie do kolumny), jak i przed numerem wiersza (blokując w ten sposób odwołanie do wiersza), np. *$B$2*.

Jeśli w adresie komórki, wraz ze zmianą położenia formuły zawierającej ten adres, nie może zmieniać się symbol kolumny lub numer wiersza, stosujemy **adresowanie mieszane**. W celu zastosowania adresu mieszanego do adresu komórki występującej w danej formule dodajemy znak $ przed numerem wiersza (blokując w ten sposób odwołanie do wiersza, np. *B$2*) lub przed literą kolumny (blokując w ten sposób odwołanie do kolumny, np. *$B2*).

Porada 3. Praca z dużymi arkuszami

Czasami obszar roboczy arkusza może nie mieścić się na ekranie. Podczas przewijania z ekranu mogą zniknąć nazwy kolumn (albo nazwy wierszy), co utrudnia orientację. Aby tego uniknąć, można zablokować przewijanie pewnej części arkusza. W tym celu należy skorzystać z opcji **Zablokuj okienka** (w menu **Okno** lub na karcie **Widok**). Zablokowane zostaną komórki położone na lewo i w górę od aktualnie zaznaczonej. Operację tę można cofnąć, korzystając z polecenia **Odblokuj okienka**.

Porada 4. Praca w kilku arkuszach

Jeśli chcemy, aby takie same dane i obliczenia były wprowadzone równocześnie na kilku arkuszach w jednym skoroszycie, należy, przytrzymując klawisz **Ctrl**, kliknąć zakładki tych arkuszy (w dolnej części okna). Jeśli mają to być kolejne arkusze, do ich zaznaczenia można wykorzystać klawisz **Shift**.

Porada 5. Korzystanie z wbudowanych stylów do formatowania tabeli

Chcielibyśmy ozdobić arkusz, ale nie mamy czasu na żmudne ustalanie poszczególnych elementów formatowania. W takim przypadku najprościej jest skorzystać z gotowych stylów proponowanych przez program, wybierając je w opcjach **Format/ Autoformatowanie** lub **Style komórki** na karcie **Narzędzia główne**.

Należy zaznaczyć zakres komórek, który ma zostać sformatowany, a następnie w otwartym oknie wybrać jeden z proponowanych stylów (rys. 7.).



Rys. 7. *Przykładowe okna wyboru stylów formatowania w programie Microsoft Excel*

Porada 6. Sortowanie list

Dane w tabelach można sortować. Po zaznaczeniu danych (np. wierszy) należy skorzystać z opcji **Dane/Sortuj**.

Przy sortowaniu numerowanych list nie należy zaznaczać kolumn zawierających numery porządkowe, by nie zostały one posortowane razem z listą, co zwykle jest niepotrzebne.

Ćwiczenie 9. Sortujemy listę uczniów

1. Otwórz plik *Uczniowie*, zapisany w ćwiczeniu 1. Posortuj listę uczniów alfabetycznie.
2. Zapisz plik pod tą samą nazwą.

Uwaga: Przed sortowaniem wyświetl ponownie wszystkie wiersze tabeli.

Porada 7. Numerowanie list

Program Microsoft Excel może dokończyć za nas tworzenie regularnych list, np. z kolejnymi numerami, nazwami miesięcy, nazwami dni tygodnia itp.

Aby wstawić nazwy kolejnych dni tygodnia, należy:

- do komórki *B1* wpisać „poniedziałek”, do komórki *B2* – „wtorek”,
- zaznaczyć komórki *B1* i *B2*,
- uchwycić mały czarny prostokąt, znajdujący się w prawym dolnym rogu zaznaczenia i przeciągnąć go w dół o potrzebną liczbę komórek.

Uwaga: W ten sam sposób wprowadzamy nazwy miesięcy, kolejne liczby itp.

Ćwiczenie 10. Tworzymy numerowane listy

W arkuszu kalkulacyjnym utwórz w 30 wierszach, w kolumnach *A:E* listy numerowane: *A* – kolejne liczby naturalne, *B* – nazwy kolejnych dni tygodnia, *C* – nazwy kolejnych miesięcy, *D* – liczby parzyste, *E* – liczby nieparzyste.

Porada 8. Zaznaczanie danych w wierszu lub kolumnie

Aby w wierszu lub kolumnie zaznaczyć komórki z danymi, należy, trzymając wciśnięty klawisz **Shift**, kliknąć dwukrotnie odpowiednio prawą lub dolną krawędź pierwszej komórki zawierającej dane. Zostaną zaznaczone komórki do pierwszej pustej.

Aby zaznaczyć całą kolumnę lub wiersz, należy kliknąć nagłówek kolumny lub wiersza. Aby zaznaczyć całą kolumnę, można również umieścić kursor w wybranej komórce danej kolumny i nacisnąć klawisze **Ctrl+spacja**. Aby zaznaczyć wiersz arkusza, można również nacisnąć klawisze **Shift+spacja**. Aby zaznaczyć cały obszar arkusza, należy kliknąć szare puste pole w lewym górnym rogu arkusza.

Porada 9. Drukowanie arkusza

Jeśli chcemy wydrukować dokument utworzony w arkuszu kalkulacyjnym, musimy zdecydować, jak powinna być zorientowana strona: poziomo czy pionowo. Należy ustalić marginesy strony: górny, dolny, prawy i lewy. Przed rozpoczęciem drukowania zawsze należy włączyć podgląd wydruku i sprawdzić, czy wszystko wygląda poprawnie. Bardzo przydatna jest funkcja **Podgląd podziału stron** – dzięki niej można łatwo zmieniać granice podziału stron, klikając je prawym przyciskiem myszy i przeciągając. W oknie podglądu wydruku można skorygować marginesy stron, wyśrodkować tabelę na stronie, dopasować dokument do strony, a także ustalić treść nagłówka i stopki. Większość z tych operacji wykonujemy, korzystając z opcji **Plik/Ustawienia strony** (lub na karcie **Układ strony**).

Warto zapamiętać

- W arkuszu kalkulacyjnym można stosować filtrowanie, które umożliwia wyszukanie danych według określonych kryteriów.
- Tabele przestawne służą do bardziej kompleksowej analizy danych, m.in. wykonywania podsumowań, obliczeń.
- Aby utworzyć tabelę przestawną, należy odpowiednio przygotować tabelę arkusza kalkulacyjnego, m.in. każda kolumna tabeli powinna mieć inny nagłówek.
- Na podstawie tabeli przestawnej można wykonać wykres przestawny.
- W tabeli przestawnej możemy stosować filtrowanie danych.

Pytania i polecenia

1. Do czego służą filtry? Jakie mają zastosowanie w praktyce?
2. Czym jest tabela przestawna? W jakim celu się ją stosuje?

3. Jak powinna być przygotowana tabela arkusza kalkulacyjnego, aby można było utworzyć tabelę przestawną?
4. Wyjaśnij na przykładzie sposób tworzenia tabeli przestawnej.
5. Wyjaśnij na przykładzie sposób tworzenia wykresu przestawnego.
6. Przedstaw trzy wybrane porady dotyczące pracy w arkuszu kalkulacyjnym.

Zadania

1. Otwórz plik *Uczniowie,* zapisany w ćwiczeniu 1. Stosując filtrowanie, wyświetl nazwiska i imiona uczniów, którzy mają:
   a. z języka polskiego i angielskiego oceny wyższe od 4,
   b. z fizyki mają ocenę dopuszczającą,
   c. mają średnią ocen większą od 3,00, ale mniejszą od 4,50.
2. Znajdź w Internecie wykaz lektur szkolnych dla liceum i technikum. Zgromadź dane (nazwisko i imię autora oraz tytuł) w tabeli arkusza kalkulacyjnego. Jeśli na podstawie danej książki został zrealizowany film, umieść w oddzielnej kolumnie słowo „Film", a w następnej – rok powstania filmu. Korzystając z filtrów:
   a. uporządkuj dane według nazwiska autora,
   b. ogranicz wyświetlanie danych do książek, na podstawie których zrealizowano film,
   c. ogranicz wyświetlanie danych do filmów zrealizowanych przed rokiem 2000.

   Zapisz plik pod nazwą *Lektury*.
3. Otwórz plik *Uczniowie_tab* zapisany w ćwiczeniu 3. W nowym arkuszu utwórz tabelę przestawną, przedstawiającą liczbę uczniów, którzy otrzymali oceny 2 i 6 z fizyki. Zapisz plik pod tą samą nazwą.
4. Otwórz plik *Pracownicy* zapisany w ćwiczeniu 6. Stosując funkcję JEŻELI, w kolejnej kolumnie wypisz decyzję o przyznaniu premii: jeśli łączna liczba przepracowanych godzin jest większa lub równa 80 – napis *TAK*, jeśli mniejsza – napis *NIE*. Zastosuj obramowanie komórek oraz umieść nagłówek z napisem PREMIA. Zapisz plik pod tą samą nazwą.
5. Otwórz plik *Pracownicy* zapisany w ćwiczeniu 6. W nowym arkuszu utwórz tabelę przestawną, w której przygotujesz zestawienie zawierające średnią liczbę godzin przepracowaną w pierwszym tygodniu przez wszystkich pracowników oraz najwyższą i najniższą pensję wypłaconą w pierwszym tygodniu. Zapisz plik pod nazwą *Zestawienie*.

Dla zainteresowanych

6. Masz do dyspozycji pewną sumę pieniędzy (np. 10 tys. zł). Aby mieć możliwość skorzystania z nich w razie potrzeby, wybierasz lokatę z miesięczną kapitalizacją odsetek. Na takiej lokacie odsetki od kwoty lokaty są co miesiąc obliczane i dopisywane do niej (każdego miesiąca odsetki są naliczane od coraz większej kwoty). Zakładając, że oprocentowanie lokaty jest stałe (np. 5,89%), oblicz, jaką kwotą będziesz dysponować po dwóch latach oszczędzania. Przedstaw na wykresie wzrost swoich oszczędności.

   Uwaga: Banki podają oprocentowanie w skali rocznej, natomiast odsetki są obliczane co miesiąc; zwróć uwagę na sformułowanie „po dwóch latach"; jakiego rodzaju adresowania należy użyć?
7. Otwórz plik *Pracownicy* zapisany w ćwiczeniu 6. Zastanów się, jakie dane chcesz przedstawić z wykorzystaniem tabeli i wykresu przestawnego. Wykonaj je według własnego pomysłu.

Temat B3. Opracowywanie grafiki wektorowej

1. Stosowanie narzędzi edytora tekstu do tworzenia rysunków w grafice wektorowej
   - 1.1. Korzystanie z kształtów
   - 1.1. Edytowanie punktów w kształtach
2. Przykładowy edytor grafiki wektorowej – program Inkscape
3. Tworzenie obrazu
   - 3.1. Rysowanie figur
   - 3.2. Rysowanie ścieżek
   - 3.3. Rysowanie przestrzenne
4. Przekształcenia
5. Warstwy

Warto powtórzyć

1. Czym charakteryzuje się grafika wektorowa? Czym różni się od grafiki rastrowej?
2. Wymień popularne formaty zapisu grafiki wektorowej.
3. Na czym polega praca z warstwami w programie GIMP?
4. Jakim przekształceniom poddawaliśmy obrazy i ich fragmenty?

1. Stosowanie narzędzi edytora tekstu do tworzenia rysunków w grafice wektorowej

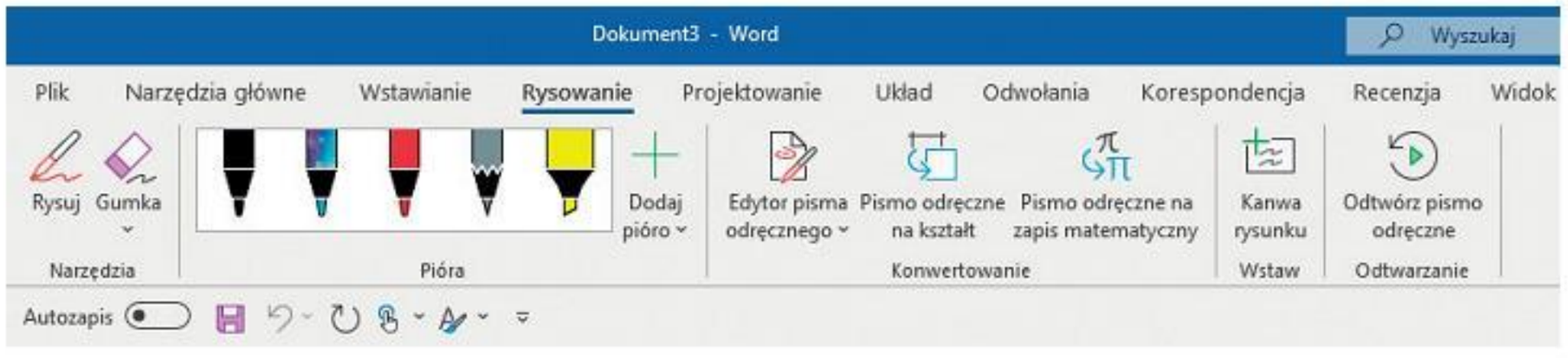
W **grafice wektorowej** obraz jest zdefiniowany matematycznie jako zbiór obiektów geometrycznych: odcinków, okręgów, wielokątów, krzywych. Każdemu obiektowi są przypisane atrybuty (kolor linii, grubość linii, wypełnienie, przezroczystość), ma on również określone wymiary i położenie względem innych obiektów.

1.1. Korzystanie z kształtów

Do wykonywania prostych projektów w grafice wektorowej można wykorzystać narzędzia rysowania programu Microsoft Word. Inne programy do edycji grafiki wektorowej oparte są na podobnych zasadach, lecz posiadają więcej możliwości pracy z obrazem.

W programie Microsoft Word mamy możliwość m.in.: wstawiania kształtów (w tym linii łamanych i krzywych, rysowania odcinków, prostokątów), rysowania elips, brył, wstawiania pól tekstowych, w których można umieścić tekst.

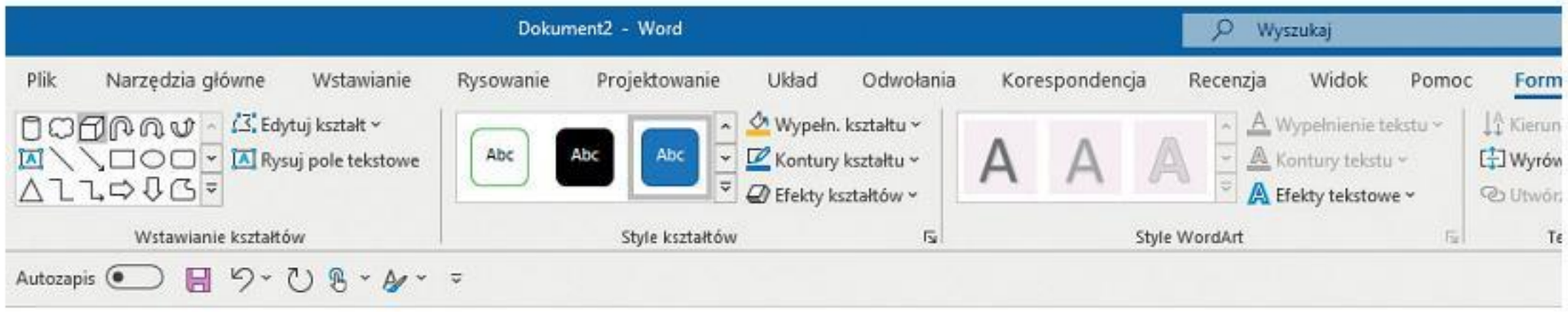
Do tworzenia prostych rysunków możemy wykorzystać standardowe narzędzia rysunkowe programu Microsoft Word, dostępne na karcie **Rysowanie** (rys. 1a).



Rys. 1a. *Narzędzia rysowania (Microsoft Word)*

Narzędzia rysunkowe odsłonią się po wybraniu konkretnego kształtu w opcji **Kształty** na karcie **Wstawianie** (rys. 1b).

Po narysowaniu obiektu mamy możliwość zmieniania jego atrybutów, takich jak: kolor wypełnienia (możemy ustalić kolor jednolity, przejście pomiędzy kolorami, tzw. gradient, wypełnienie teksturą, deseniem lub innym obrazem), kolor linii (obramowania), kolor tekstu, grubość linii (obramowania), styl linii (np.: ciągła, przerywana, kropkowana), sposób zakończenia linii (np. strzałkami), kolor i kształt cienia obiektu, rodzaj zastosowanego efektu trójwymiarowego.



Rys. 1b. *Narzędzia formatowania kształtu (Microsoft Word)*

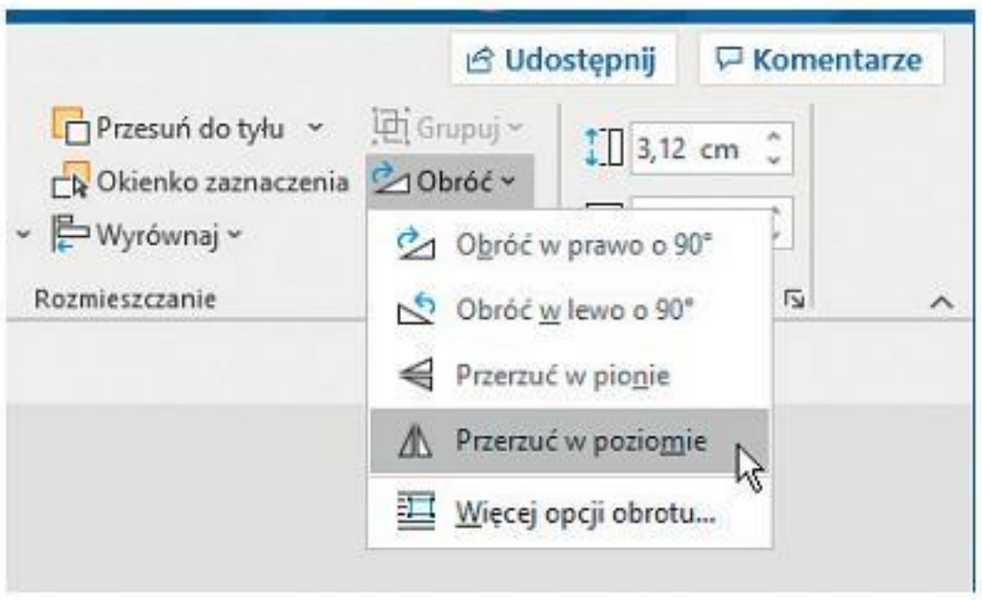
Narysowane obiekty możemy poddawać różnym **przekształceniom**. Najważniejsze z nich to:

- przesuwanie (zmiana położenia),
- zmiana rozmiarów (skalowanie),
- obracanie,
- wykonywanie odbić lustrzanych obiektu w poziomie lub w pionie (przerzucanie w poziomie lub pionie).

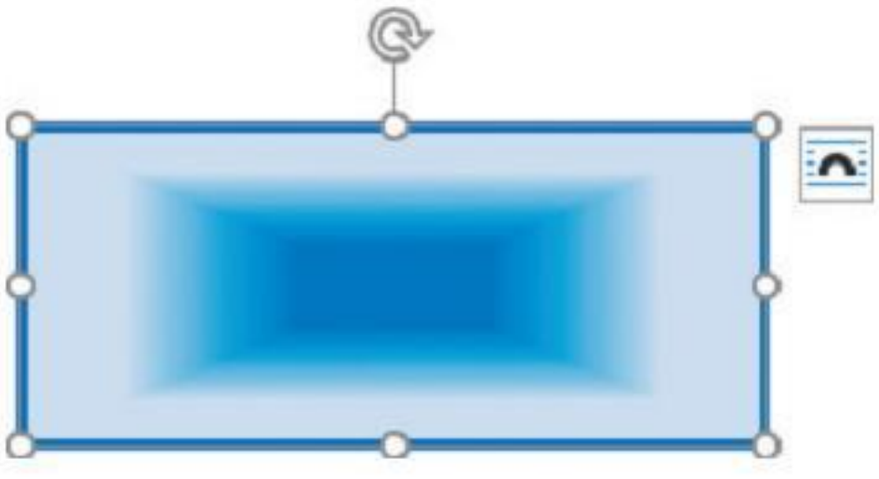
Wszystkie operacje wykonujemy po uaktywnieniu obiektu. Chwytając wyróżnione punkty i je przeciągając, zmieniamy położenie i rozmiar.

Aby obrócić obiekt, możemy uchwycić za **uchwyt obracania kształtu** i przeciągnąć w odpowiednim kierunku.

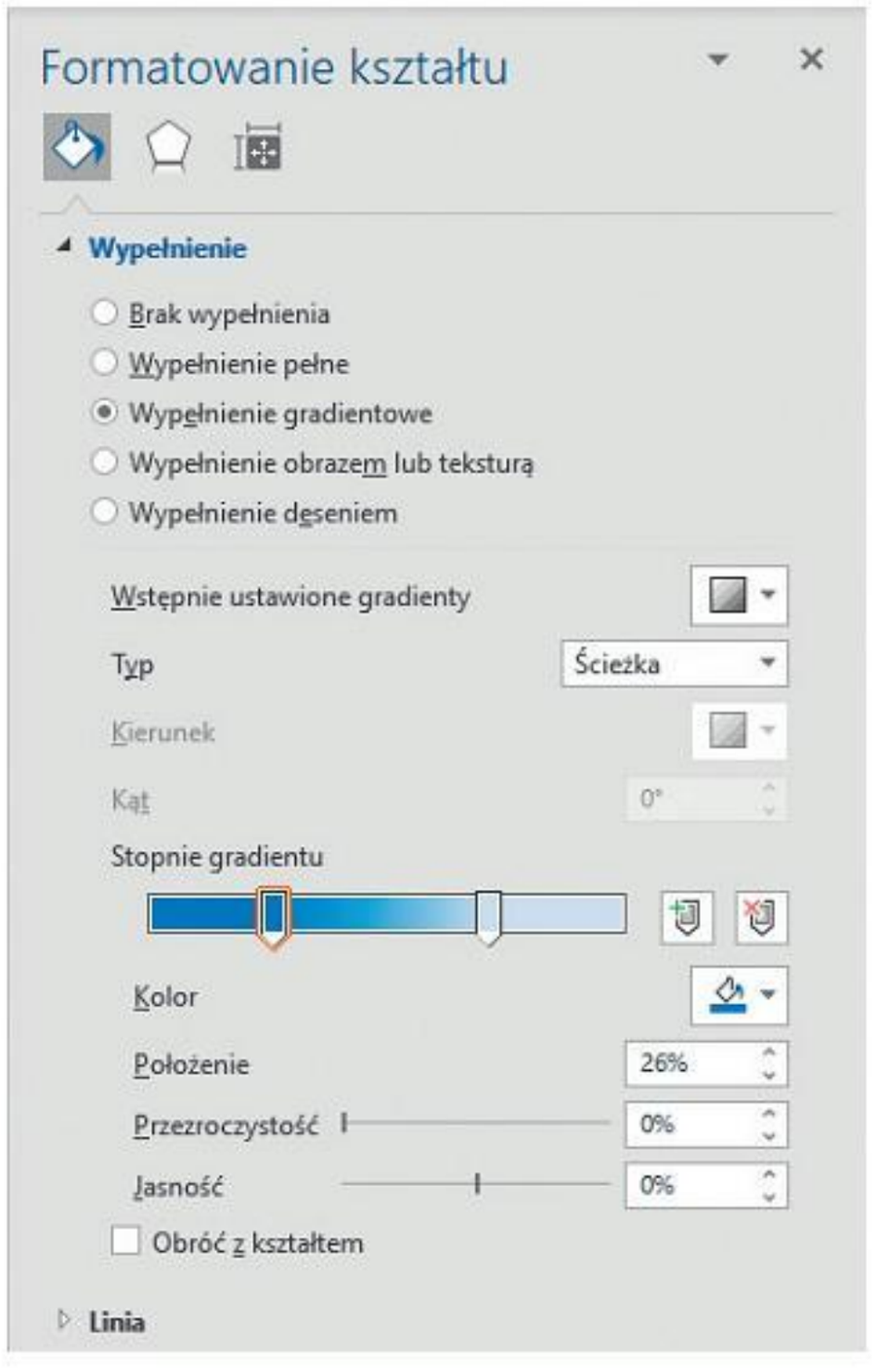
Aby wykonać lustrzane odbicie obiektu, należy na karcie **Formatowanie kształtu** (w grupie **Rozmieszczanie**) wybrać opcję **Obrót**, a w niej konkretne polecenie (rys. 2.).



Rys. 2. *Rozwinięta lista wyboru opcji* ***Obrót***



Rys. 3a. *Prostokąt wypełniony gradientem* ***Formatowanie kształtu*** *– ćwiczenie 1.*



Rys. 3b. *Panel* ***Formatowanie kształtu***

Ćwiczenie 1. Tworzymy rysunek z kształtów i wypełniamy go gradientem

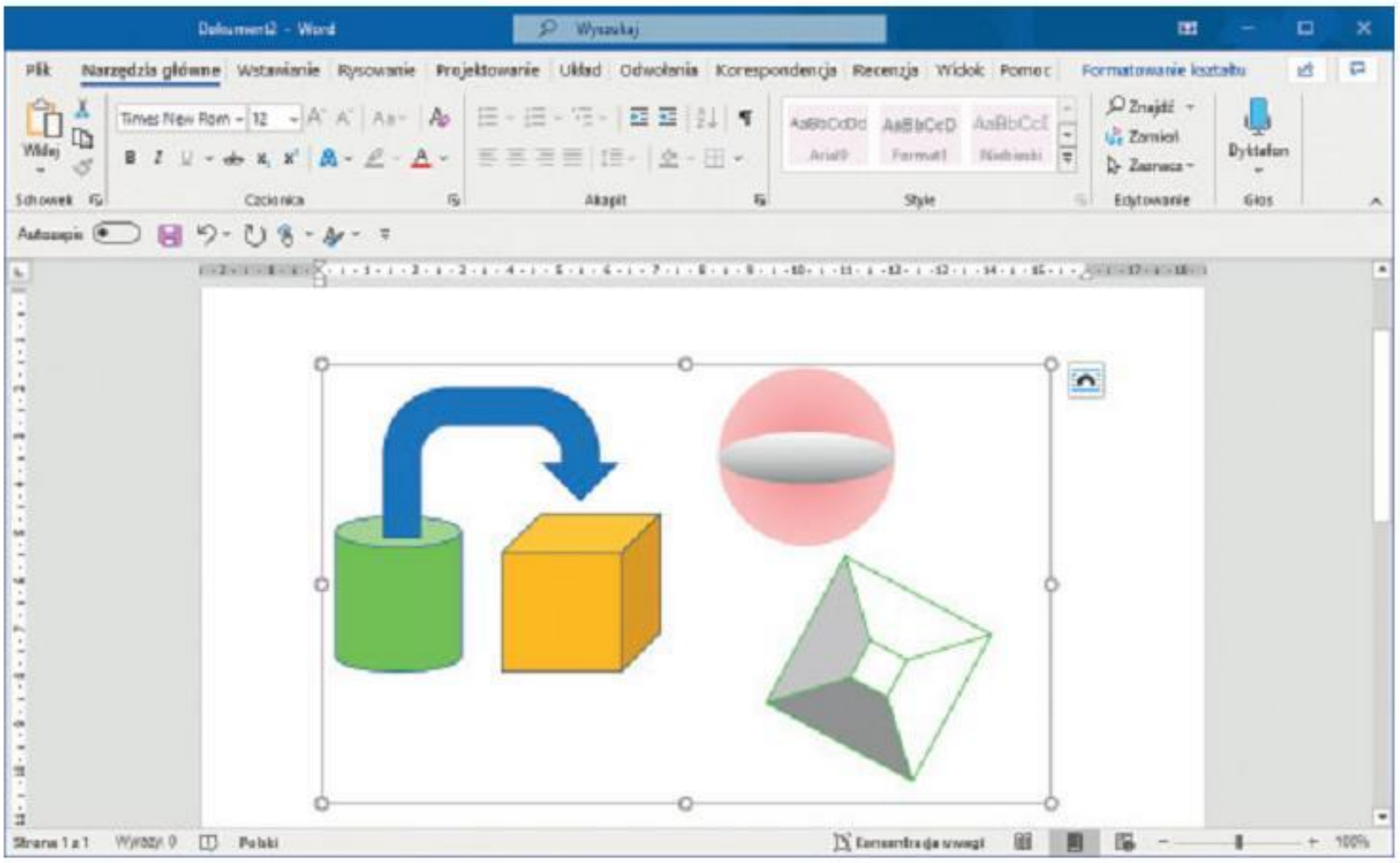
1. W edytorze tekstu Word narysuj prostokąt, korzystając z kształtów. Wypełnij go dwukolorowym gradientem i obramuj granatową linią o grubości 3 punktów. Wyjdź z edycji rysunku.
2. Wróć do edycji rysunku. Zmień obramowanie na jasnoszare, o grubości 1,5 punktu.
3. Zapisz dokument w pliku pod nazwą *Gradient*.



Rys. 4. *Fragment rozwiniętej listy po wybraniu opcji* ***Kształty*** *z widoczną opcją* ***Nowa kanwa rysunku***

W edytorach grafiki wektorowej kilka obiektów można połączyć w jeden obiekt, zaznaczając je (z wciśniętym klawiszem **Shift**) i **grupując**. Obiekty zgrupowane można rozdzielić na części składowe, **rozgrupowując** je. Opcje grupowania i rozgrupowania wybieramy z menu kontekstowego, po wcześniejszym zaznaczeniu obiektów (obiektu).

W programie Microsoft Word obiekty rysunkowe można również umieszczać na kanwie rysunku, czyli wydzielonym obszarze do rysowania (rys. 5.). Opcja **Nowa kanwa rysunku** pojawia się na dole rozwiniętego wykazu opcji **Kształty** (rys. 4.).



Rys. 5. *Kompozycja z figur wykonana przy użyciu kształtów umieszczonych na kanwie rysunku*

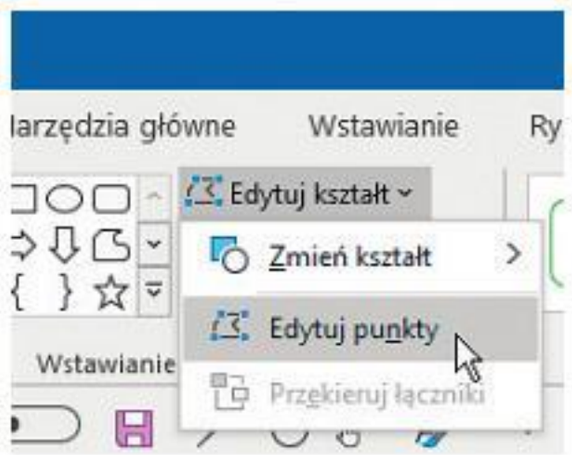
Ćwiczenie 2. Tworzymy kompozycję z kształtów

1. W programie Microsoft Word utwórz kompozycję (na kanwie rysunku) składającą się z różnych figur, np. podobną do pokazanej na rysunku 5. Wykorzystaj różne możliwości przekształceń oraz kolorowania. Zgrupuj wybrane obiekty.
2. Zapisz rysunek w pliku pod nazwą *Kompozycja 1.*

1.2. Edytowanie punktów w kształtach

Korzystając z narzędzi do rysownia dostępnych w programie Microsoft Word, można nie tylko zmienić rozmiar, ale i kształt narysowanej figury. Możemy **edytować punkty** większości kształtów.

Aby edytować punkty kształtu, należy go uaktywnić i na karcie **Formatowanie kształtu** wybrać opcję **Edytuj kształt/Edytuj punkty** (rys. 6.) – na konturze kształtu pojawią się punkty w postaci małych czarnych kwadracików (np. wierzchołki tworzące kontur – rys. 7.).



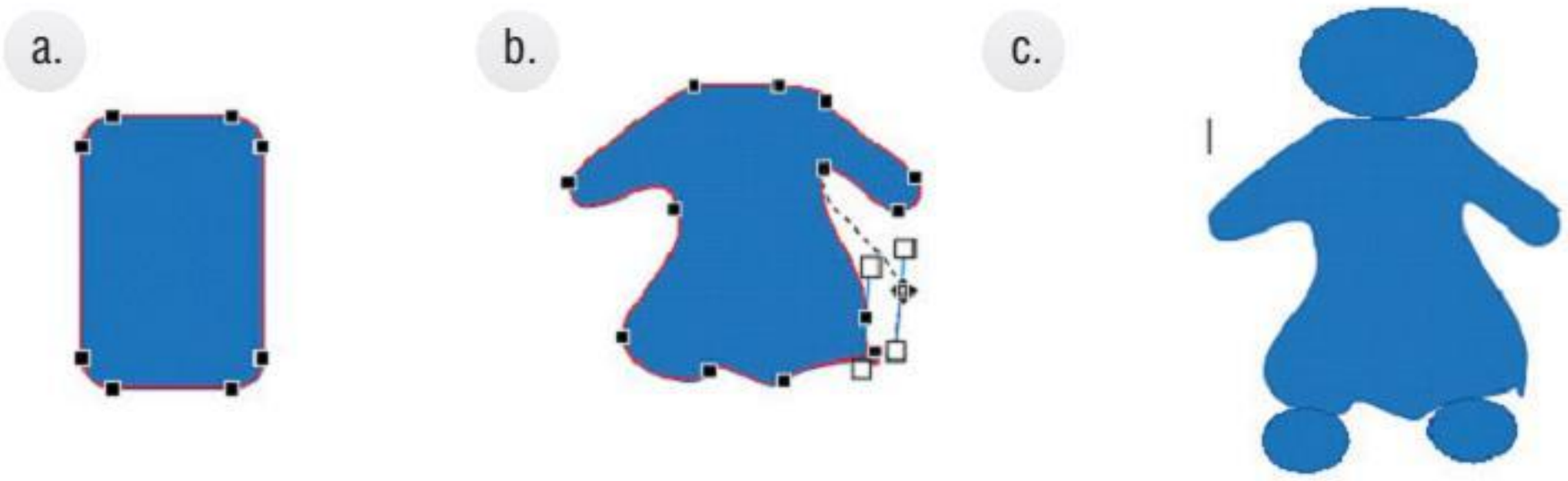
Rys. 6. *Opcja* ***Edytuj punkty***

Przeciągając wybrane punkty, możemy tworzyć dowolną figurę.

Aby dodać nowy punkt, należy kliknąć kontur kształtu, trzymając wciśnięty klawisz **Ctrl**. **Aby usunąć punkt**, należy go kliknąć, trzymając wciśnięty klawisz **Ctrl**.

Ćwiczenie 3. Tworzymy rysunek, edytując punkty kształtu

1. Wyjaśnij, w jaki sposób powstał rysunek ludzika pokazany na rysunku 7c.
2. Utwórz rysunek, korzystając z możliwości edytowania punktów wybranego kształtu.
3. Zapisz rysunek w pliku pod nazwą określającą temat twojego projektu.



Rys. 7. *Przykład edytowania punktów prostokąta z zaokrąglonymi rogami*

2. Przykładowy edytor grafiki wektorowej – program Inkscape

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.

Jednym z zaawansowanych programów służących do edycji obrazów wektorowych jest program Inkscape, rozpowszechniany na otwartej licencji GNU GPL. Podobnie jak inne programy graficzne (np. CorelDRAW, Adobe Illustrator) Inkscape dysponuje dużą liczbą różnorodnych narzędzi malarskich, narzędzi selekcji i paletą kolorów, umożliwia pracę na warstwach i wykonywanie przekształceń. Można korzystać z filtrów, nakładać maski i ustawiać przezroczystości oraz stosować klonowanie.

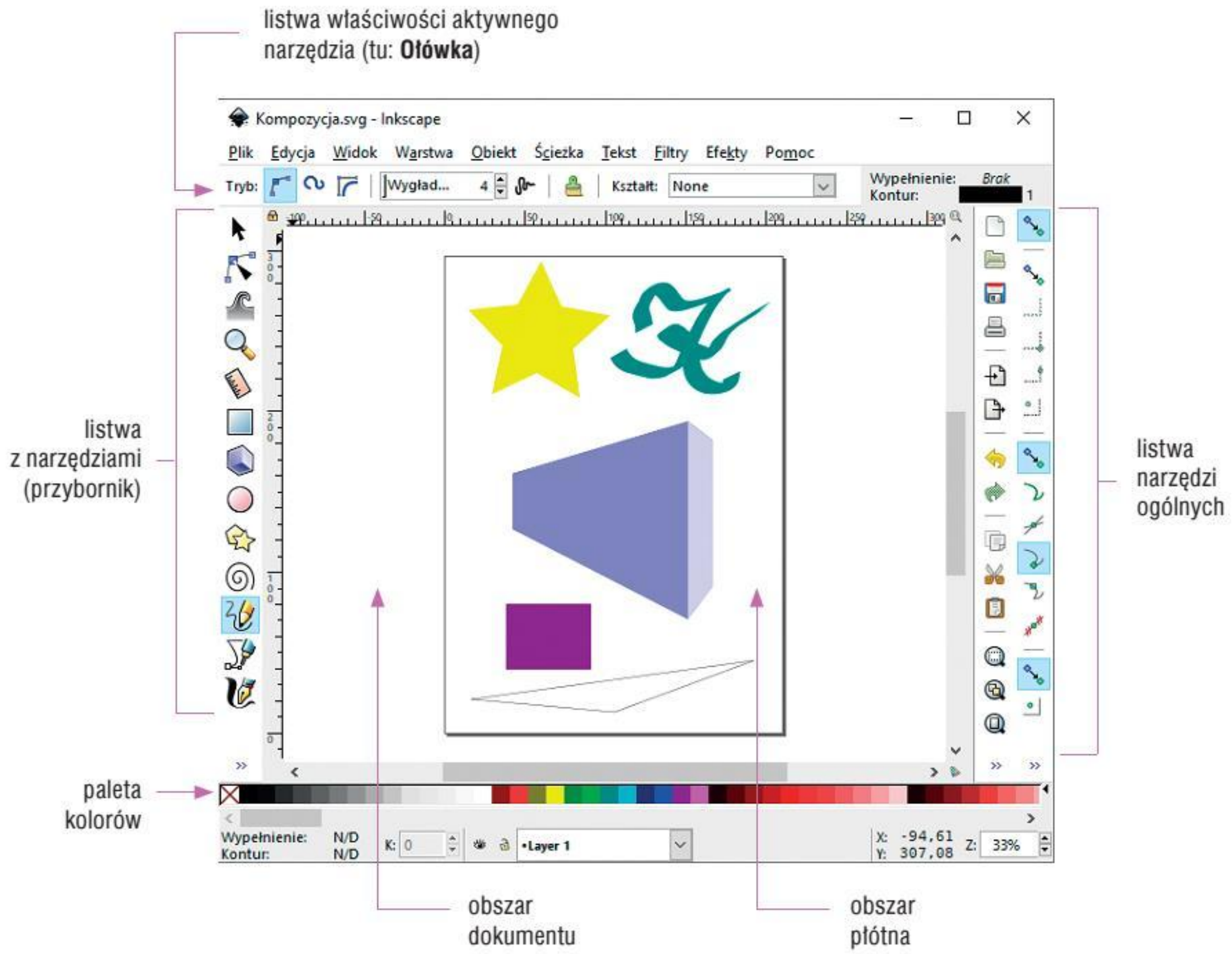
W temacie pokażemy tylko kilka wybranych funkcji programu, pozostawiając osobom zainteresowanym grafiką komputerową zapoznanie się z pozostałymi możliwościami.

W oknie programu Inkscape (rys. 8.) możemy wyróżnić obszar płótna, który jest częścią obszaru dokumentu; rysować można również na pozostałym obszarze dokumentu. Na linijkach otaczających obszar widzimy położenie kursora (sam kursor zmienia kształt w zależności od aktualnie używanego narzędzia).

Ćwiczenie 4. Tworzymy kompozycję z kształtów

1. Uruchom program Inkscape. Zapoznaj się z interfejsem programu, tworząc kompozycję z różnych figur, np. taką jak na rysunku 8.
2. Zapisz rysunek w pliku pod nazwą *Kompozycja 2*.

Wskazówka: W programie Inkscape pliki są domyślnie zapisywane z rozszerzeniem *svg*.



Rys. 8. *Kompozycja z figur wykonana w obszarze płótna*

3. Tworzenie obrazu

3.1. Rysowanie figur

W programie Inkscape, podobnie jak w innych programach graficznych, możemy rysować różne figury: prostokąty, elipsy, wielokąty, linie proste. Dodatkowo można rysować m.in. spirale.

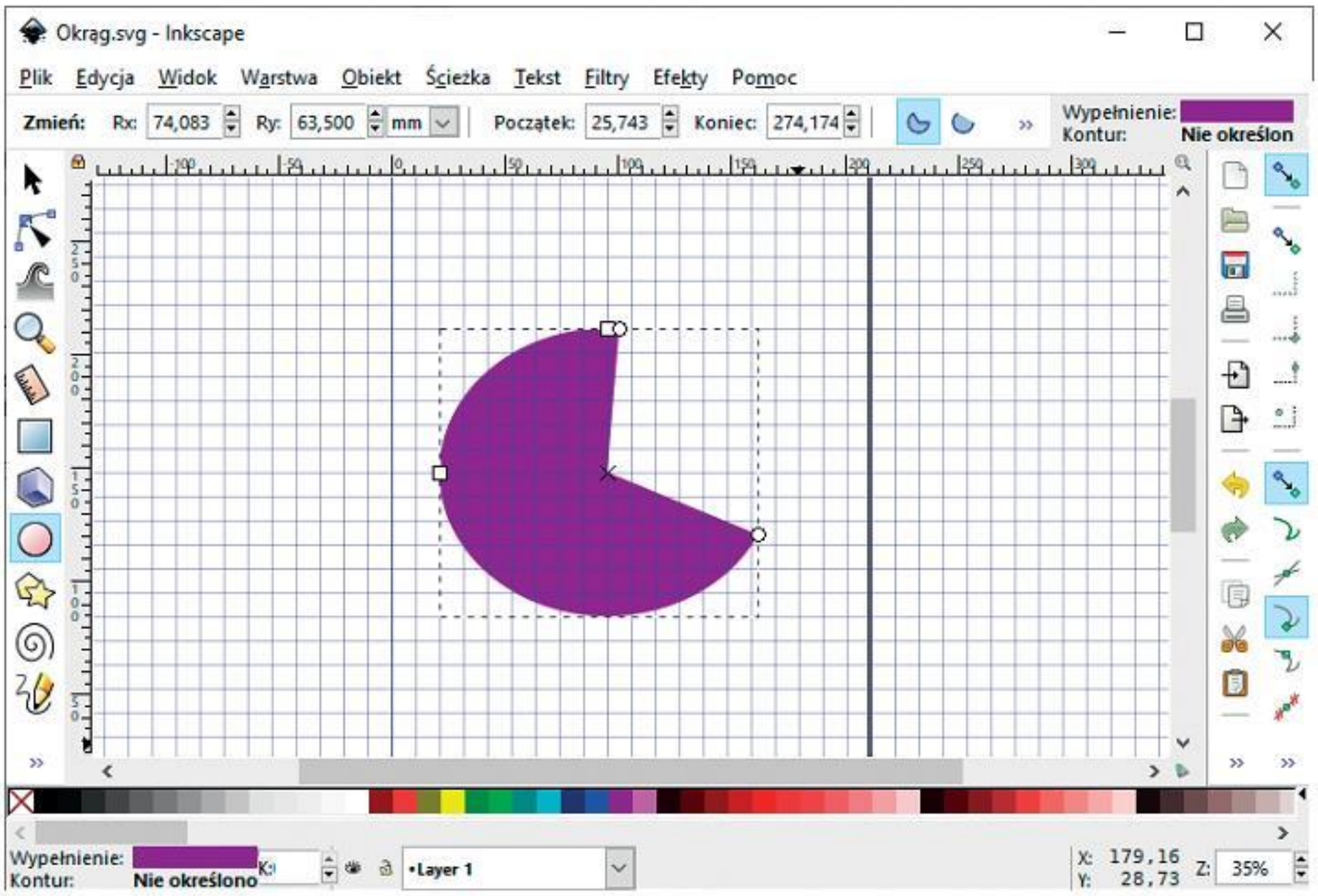
W trybie edycji kształtu widoczne są tzw. **węzły edycyjne** (na rysunku 9. w postaci małych kwadracików i kółeczek), które umożliwiają zmianę wybranych właściwości kształtu. Każdy kształt ma co najmniej jeden taki węzeł. Na przykład w przypadku okręgu pojawiają się dwa kwadratowe węzły do zmiany promienia i dwa okrągłe do zmiany na inną formę (np. na pokazaną na rysunku 9.)

Kolor wypełnienia i konturu figury możemy zmienić, korzystając z menu kontekstowego figury i wybierając polecenie **Wypełnienie i kontur** – otworzy się okno jak na rysunku 10. Kolor tła i konturu można również zmienić, klikając wybrany kolor na palecie kolorów (w przypadku konturu należy nacisnąć i przytrzymać klawisz **Shift**).

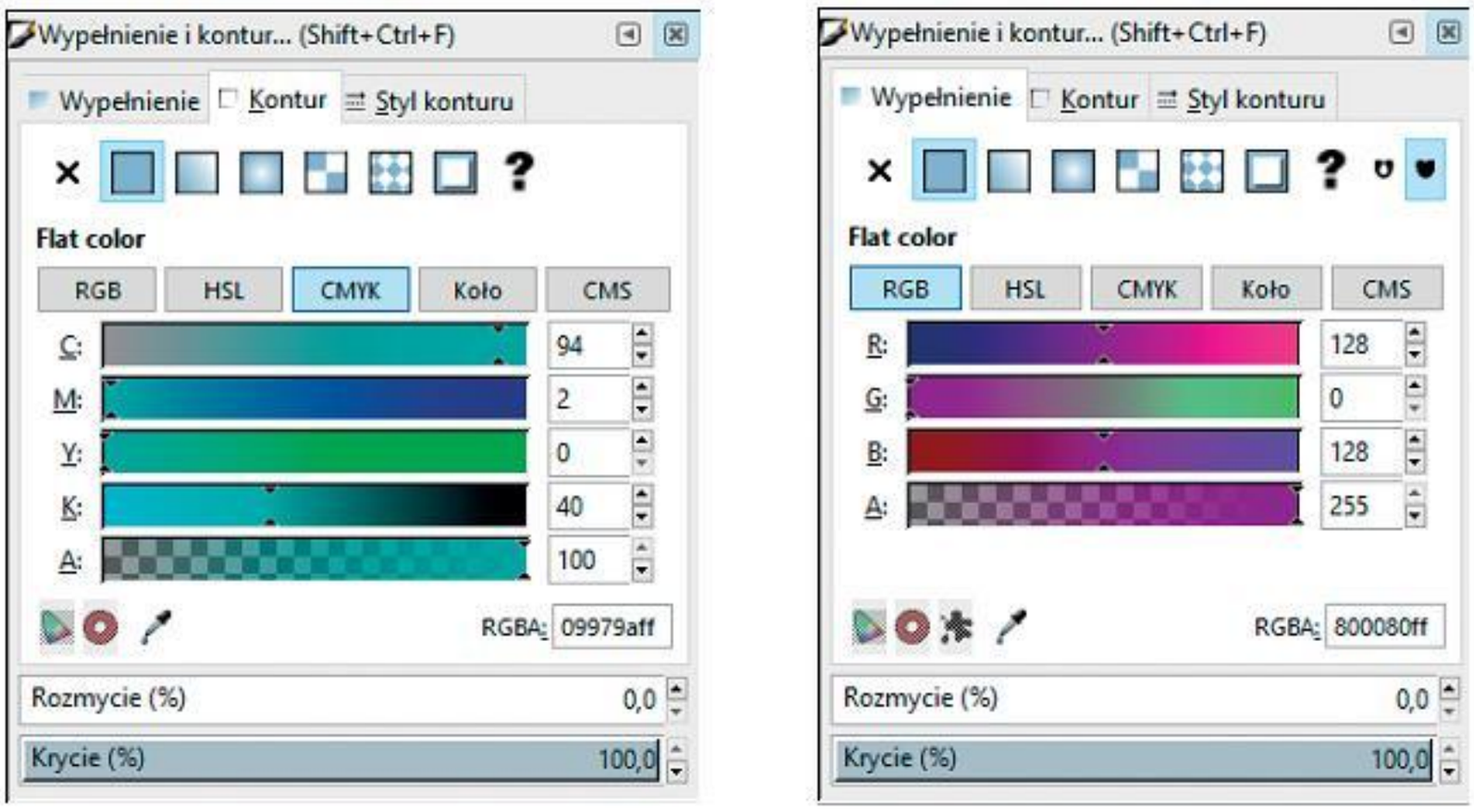
Uwaga

W opcji **Widok** można włączyć linie siatki, które ułatwiają ocenę rozmiarów obiektów oraz ich rozmieszczanie.

Po wybraniu narzędzia na pasku narzędzi edycji obrazu możemy ustalić rodzaj rysowanego kształtu i zmieniać jego właściwości na listwie właściwości. Na przykład po kliknięciu narzędzia **Okrąg** można wybrać rysowanie okręgu, wycinka koła lub łuku. W przypadku narzędzia **Gwiazda** – wybieramy gwiazdę lub wielokąt, możemy też wybrać liczbę narożników (pole **Narożniki**).



*Rys. 9. Zastosowanie narzędzia **Okrąg** do rysowania wycinka koła*

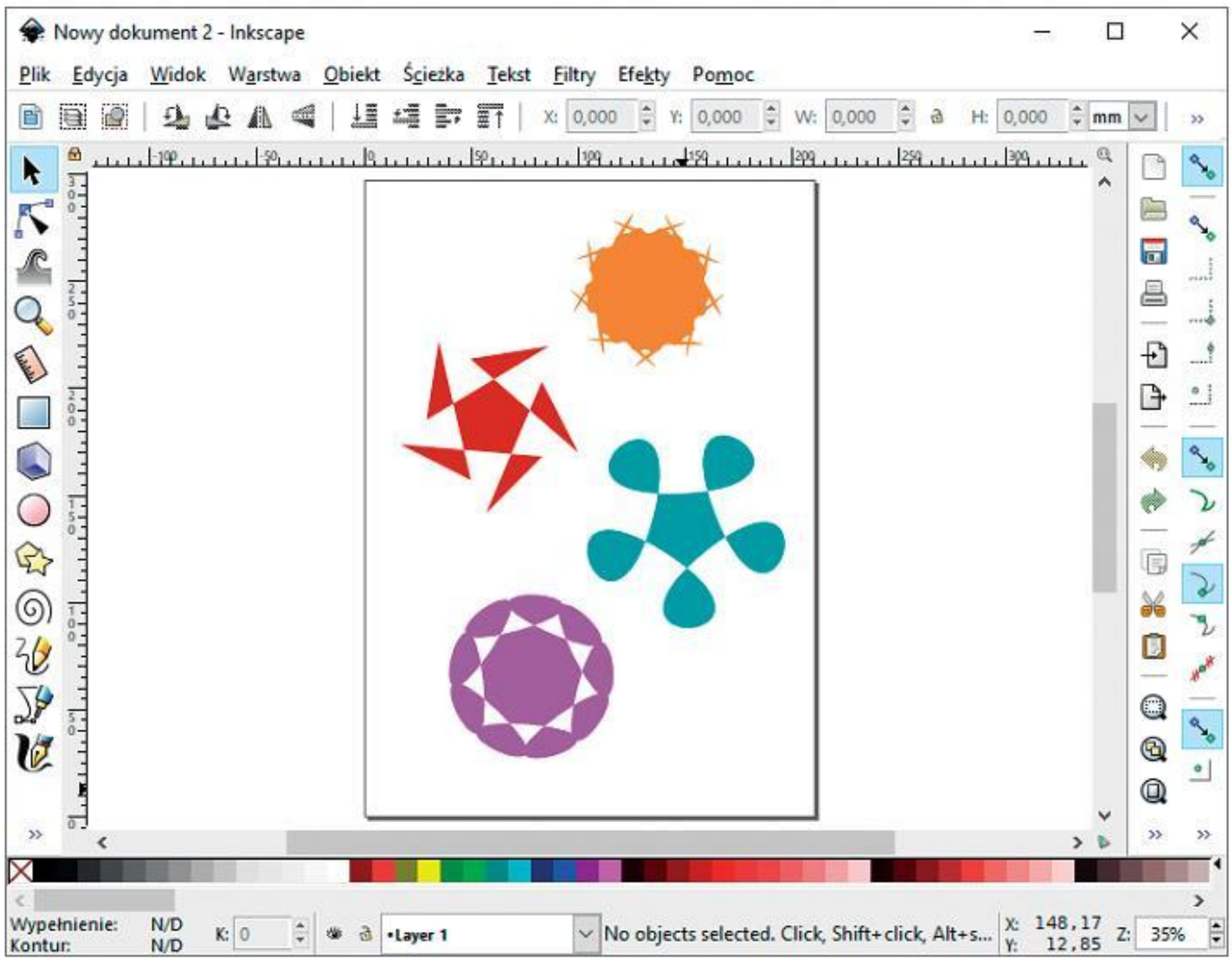


Rys. 10. Wybieranie koloru konturu i wypełnienia

Ćwiczenie 5. Rysujemy figury z wykorzystaniem narzędzia **Okrąg**

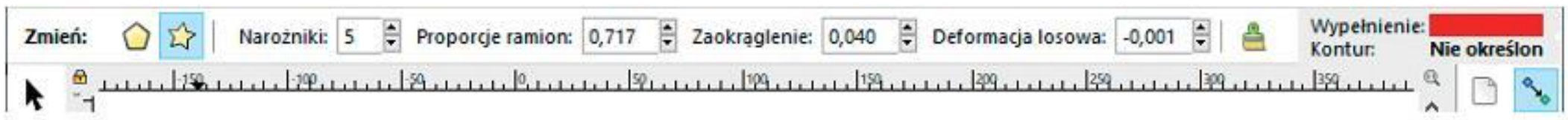
1. Narysuj okrąg, koło, półkole i wypełniony łuk. Każda figura powinna mieć taki sam promień. Zastosuj żółty kolor wypełnienia i granatowy kolor konturu. Pod każdym rysunkiem umieść odpowiedni podpis.
2. Zapisz rysunek w pliku pod nazwą *Okrąg*.

Korzystając z narzędzia **Gwiazda (Tworzenie gwiazd i wielokątów)** , można uzyskać figury o ciekawych, różnorodnych kształtach. Na rysunku 11. pokazano efekty rysowania wielokąta, ale z różnymi wartościami parametrów. Można je zmieniać, wpisując odpowiednie wartości do pól umieszczonych na listwie właściwości danego narzędzia lub przeciągając węzły edycyjne danego kształtu.



Rys. 11. *Rysunki wykonane za pomocą narzędzia* ***Gwiazda***

Ciekawe efekty uzyskuje się po zmianie parametru **Deformacja losowa** – jeśli jego wartość jest większa od zera, kształt zostaje zdeformowany w losowy sposób. Jeśli chcemy anulować wszystkie parametry, należy kliknąć narzędzie **Przywróć domyślne ustawienia dla kształtu** (rys. 12.).



Rys. 12. *Listwa właściwości narzędzia* ***Gwiazda***

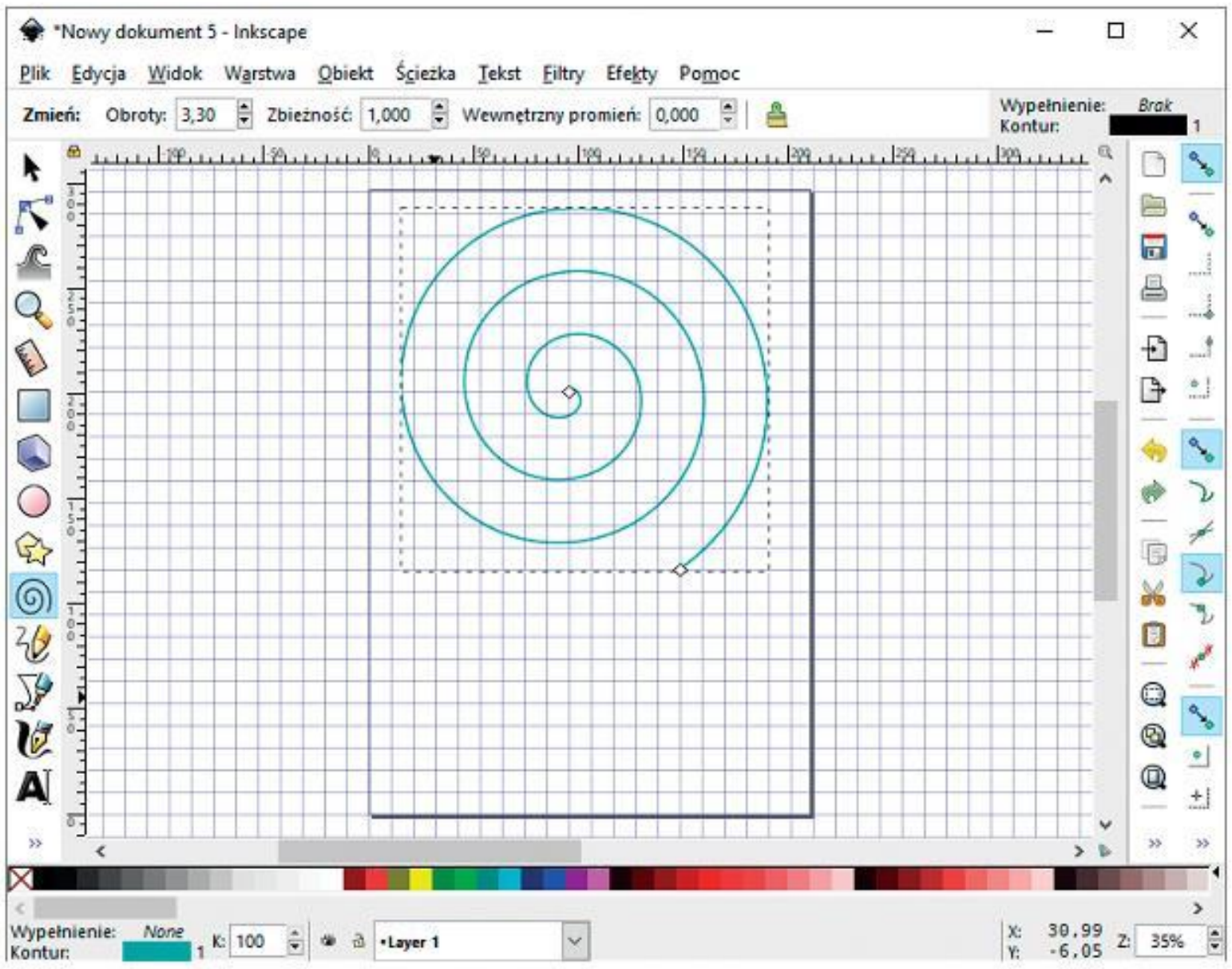
Ćwiczenie 6. Zmieniamy właściwości narzędzia

1. Narysuj figury podobne do pokazanych na rysunku 11., zmieniając odpowiednio właściwości narzędzia **Gwiazda**.
2. Zapisz rysunek w pliku pod nazwą *Figury*.

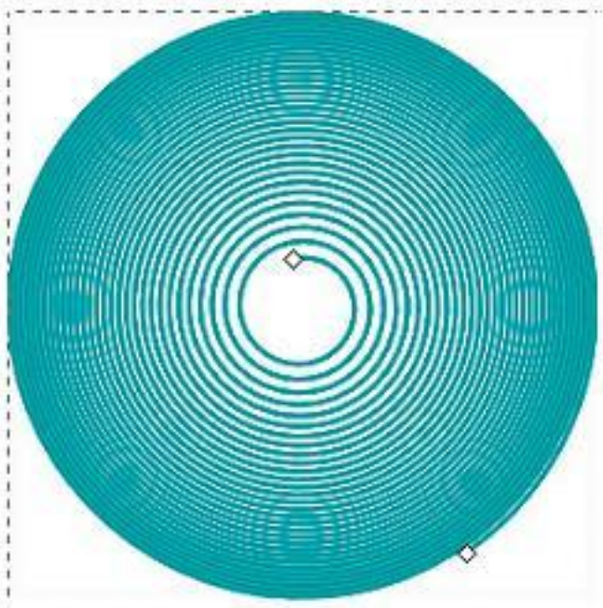
W programie Inkscape dostępne jest narzędzie **Spirala** , które umożliwia rysowane **spirali Archimedesa** (rys. 13.). Na listwie właściwości można zmieniać:

- liczbę obrotów – **Obroty**,
- zbieżność – **Zbieżność**,
- wewnętrzny promień – **Wewnętrzny promień**.

Zmieniając wskazane parametry, można uzyskiwać różne efekty (rys. 14.).



Rys. 13. *Spirala Archimedesa*



Rys. 14. *Efekt rysowania spirali po zmianie wybranych parametrów*

Ćwiczenie 7. Rysujemy spirale

1. Korzystając z **Pomocy** programu Inkscape, sprawdź, co określają poszczególne parametry na listwie narzędziowej dla spirali. Narysuj cztery spirale, zmieniając wybrane parametry.
2. Zapisz rysunek w pliku pod nazwą *Spirale*.

3.2. Rysowanie ścieżek

Ścieżkami nazywamy obiekty tworzone za pomocą narzędzi:

- **Ołówek**,
- **Pióro**,
- **Kaligrafia**.

Podstawowym narzędziem do rysowania ścieżek jest **Ołówek**. Możemy nim rysować m.in. ścieżki odręczne, które składają się z wielu krótkich, połączonych ze sobą **krzywych Béziera**.

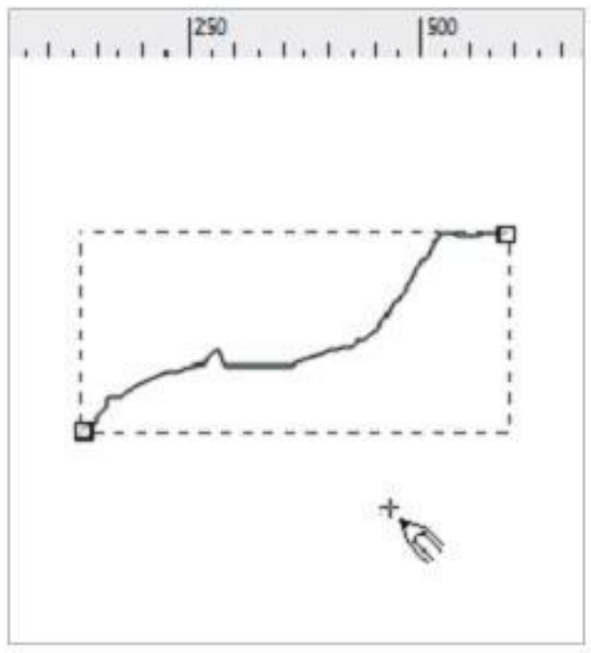
> **Ś Ścieżka**
>
> Sekwencja prostych odcinków i/lub krzywych Béziera. W odróżnieniu od innych kształtów, może być edytowana poprzez swobodne przeciąganie dowolnego węzła, nie tylko wcześniej ustalonych uchwytów. Jak inne kształty może mieć wypełnienie i kontur.

Krzywe Béziera to krzywe parametryczne, powszechnie stosowane w programach do projektowania inżynierskiego i projektowania grafiki komputerowej. Ich cechą charakterystyczną są **punkty węzłowe**, którymi można manipulować, zmieniając w ten sposób kształt krzywej. Taką krzywą można opisać matematycznie.

Korzystając z **Ołówka**, możemy wybrać (na listwie właściwości narzędzia) dwa tryby pracy: **Tworzy regularną ścieżkę krzywych Béziera** lub **Tworzy ścieżkę Spiro**. Możemy również dodać wygładzanie ścieżki, zmieniając parametr **Wygładzenie**, oraz wybrać kształt ścieżki – korzystając z parametru **Kształt**.

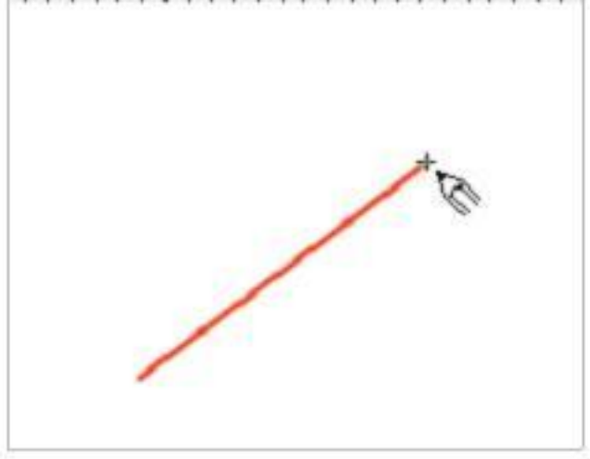
Jeśli przesuwamy kursor w obszarze płótna, rysujemy linię odręczną. Aby zakończyć rysowanie, zwalniamy przycisk myszy. Na rysunku 15. pokazano efekt rysowania ścieżki w trybie **Tworzy regularną ścieżkę krzywych Béziera**.

Za pomocą **Ołówka** możemy również rysować linie proste. Jeśli klikniemy w obszarze płótna (bez przesuwania myszy), wyznaczymy punkt początkowy prostej, a przesuwając kursor, wyznaczymy ścieżkę przedstawioną w postaci linii czerwonego koloru (rys. 16a). Klikając, kończymy rysowanie linii (rys. 16b).

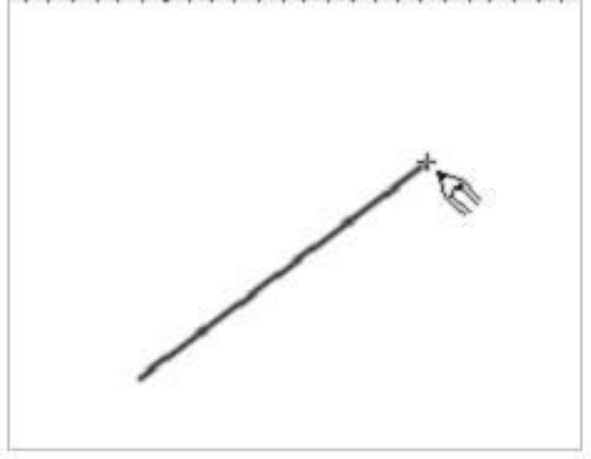


*Rys. 15. Ścieżka odręczna narysowana **Ołówkiem***

a.



b.



*Rys. 16. Linia prosta narysowana **Ołówkiem***

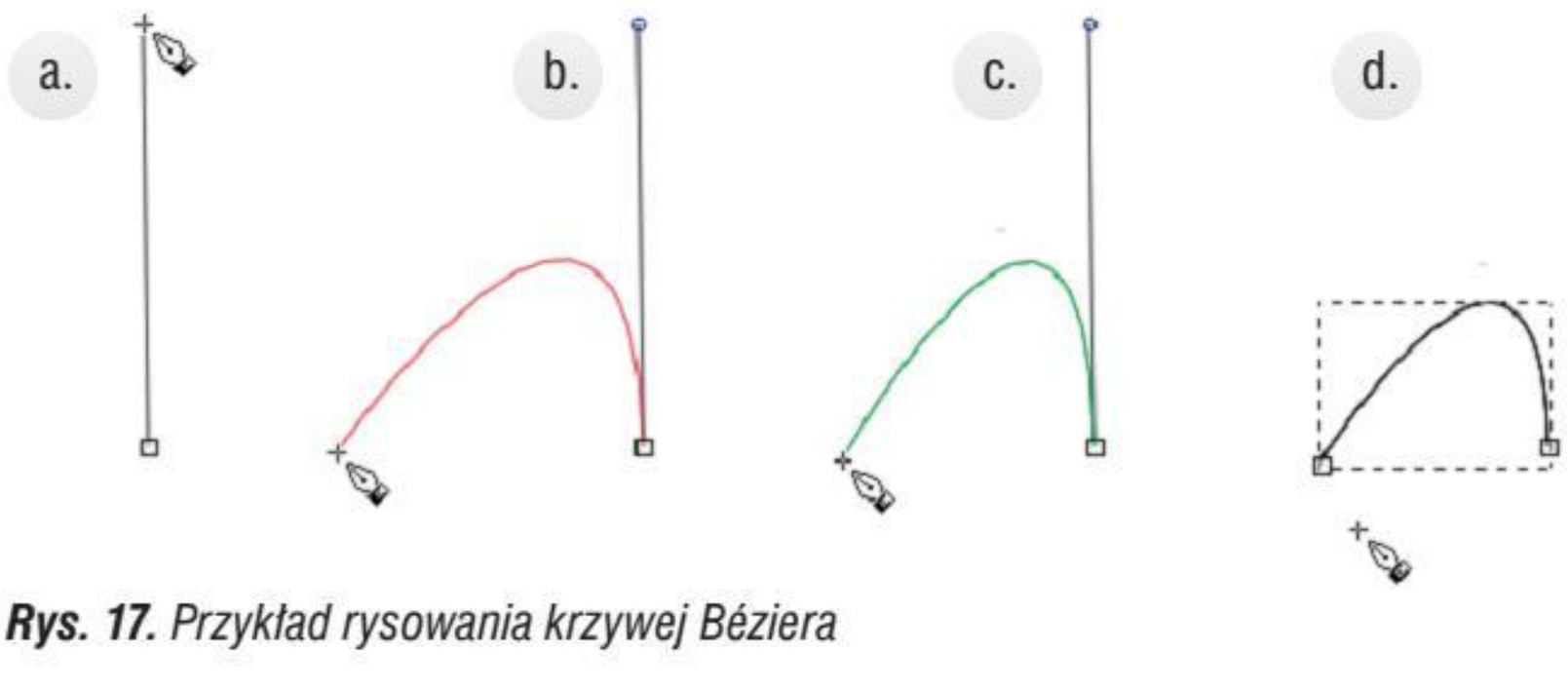
Ćwiczenie 8. Korzystamy z narzędzia **Ołówek**

1. Narysuj **Ołówkiem** kompozycję składającą się z zer i jedynek, stosując do pisania tych cyfr różne możliwości wygładzania, wybrane kształty oraz rysowanie ścieżek odręcznych i linii prostych.
2. Zapisz rysunek w pliku pod nazwą *Ołówek*.

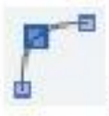
Za pomocą narzędzia **Pióro** możemy rysować ścieżki odręczne i linie proste oraz krzywe Béziera dowolnego kształtu.

Ćwiczenie 9. Rysujemy krzywe Béziera

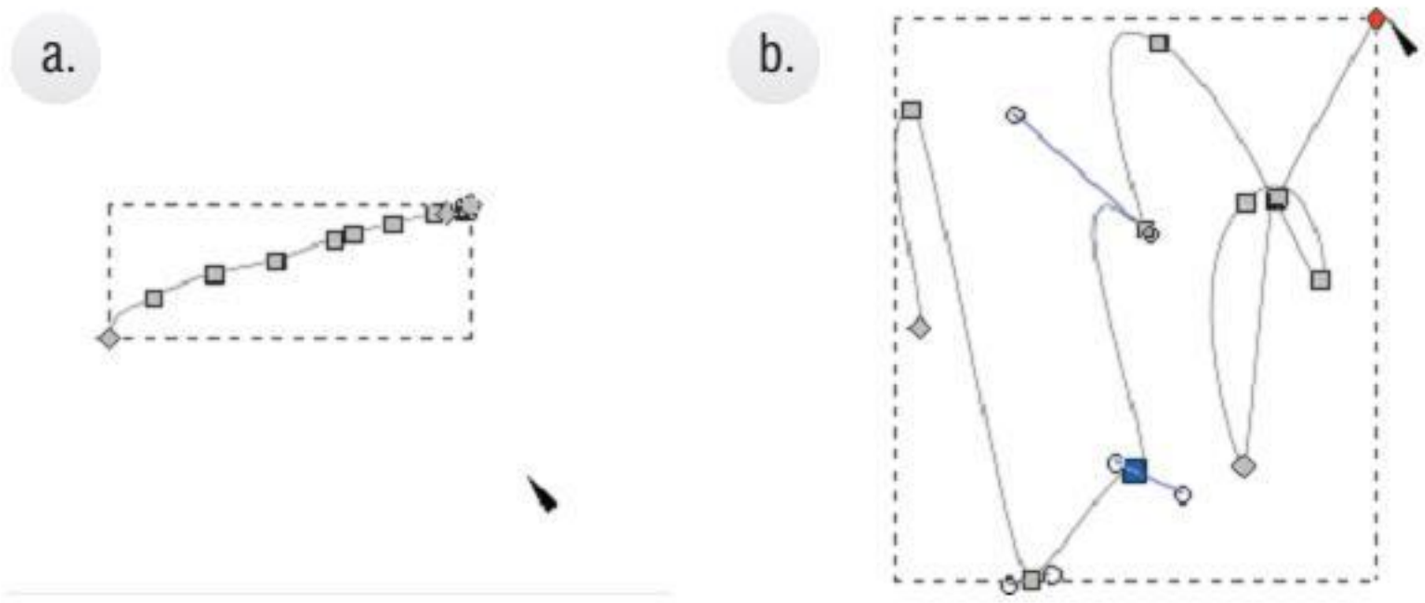
1. Wzorując się na rysunku 17., narysuj krzywą Béziera. Zwróć uwagę na różne kolory krzywej na poszczególnych etapach rysowania.
2. Zapisz rysunek w pliku pod nazwą *Krzywe*.



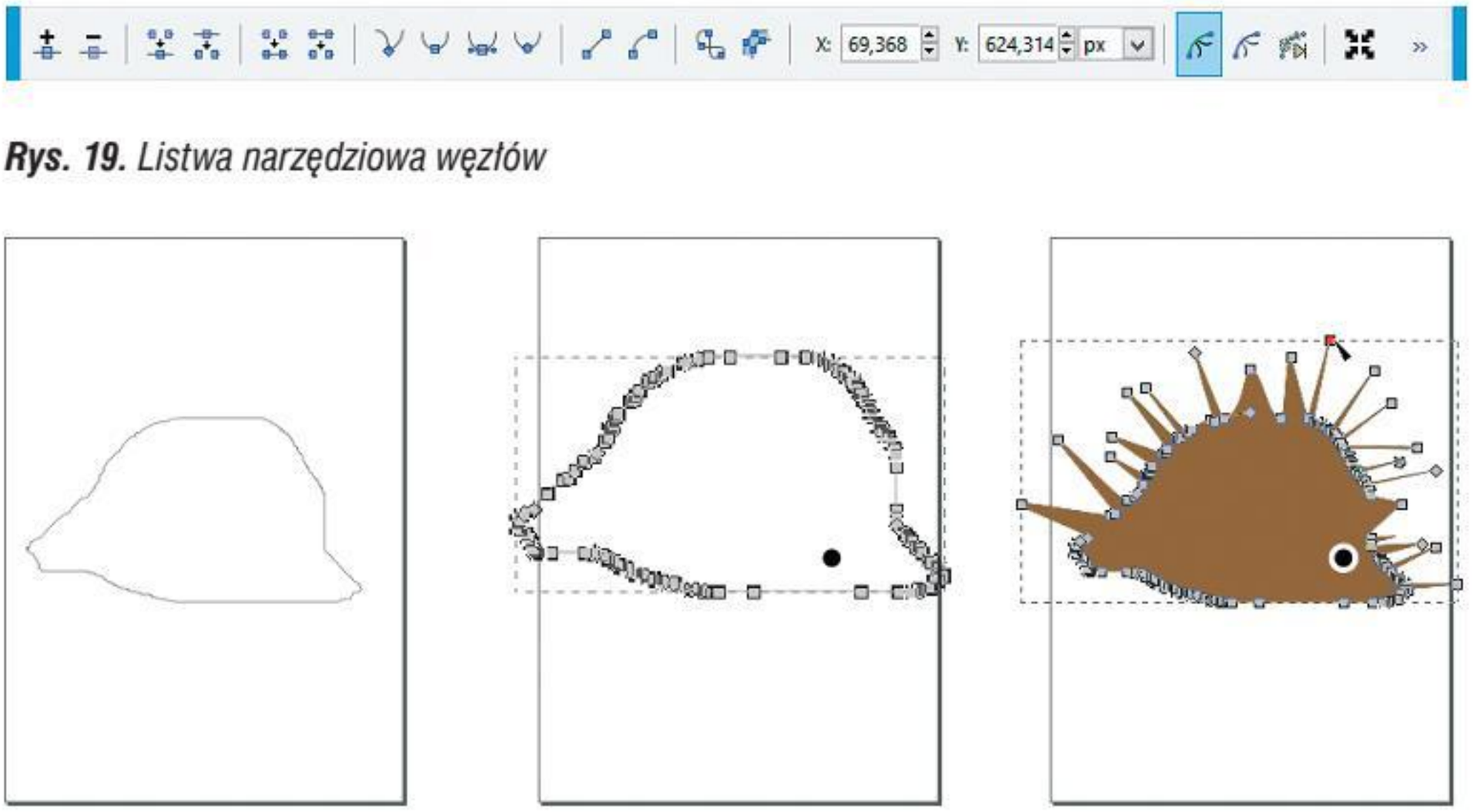
Rys. 17. *Przykład rysowania krzywej Béziera*

Po narysowaniu ścieżki **Ołówkiem** lub **Piórem** możemy modyfikować jej węzły, zmieniając ich położenie, dodając je i usuwając. Po kliknięciu narzędzia **Edycja węzłów**  na ścieżce pokażą się wszystkie węzły w postaci małych kwadratów, rombów, okręgów (różne oznaczenia są związane z różnymi typami węzłów). Węzły można przeciągać, co powoduje zmianę wyglądu krzywej (rys. 18.)

Na listwie narzędziowej węzłów możemy wybrać wiele sposobów modyfikacji ich właściwości (więcej informacji na temat węzłów można znaleźć w **Pomocy;** zapoznanie się z nimi pozostawiamy zainteresowanym).



Rys. 18. *Ścieżka narysowana* ***Ołówkiem****. Po prawej stronie ta sama ścieżka po zmianie położenia niektórych węzłów*



Rys. 19. *Listwa narzędziowa węzłów*

Rys. 20. *Przykładowy rysunek – ćwiczenie 10.*

Ćwiczenie 10. Tworzymy rysunek

1. Narysuj **Ołówkiem** ścieżkę odręczną, a następnie, zmieniając położenie węzłów, wykonaj rysunek jeżyka (przykład na rys. 20.).
2. Zapisz rysunek w pliku pod nazwą *Jeżyk*.

3.3. Rysowanie przestrzenne

Program Inkscape umożliwia rysowanie w perspektywie i tworzenie rzutów brył.

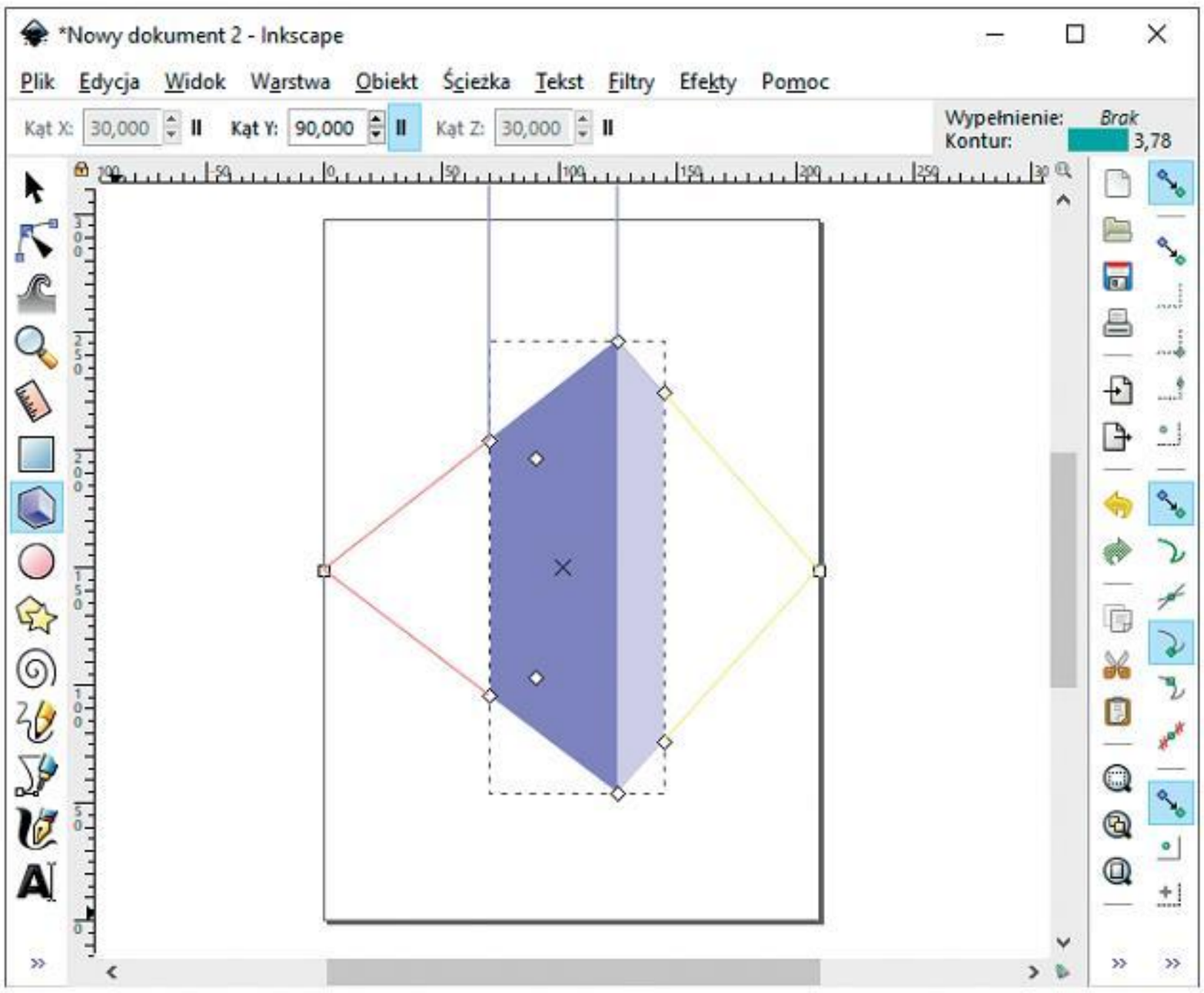
Po kliknięciu narzędzia **Obiekt 3D** możemy rozpocząć tworzenie rzutu bryły. Bryłę najlepiej rysować na obszarze płótna, ponieważ wówczas na lewej i prawej krawędzi obszaru płótna widoczne są **punkty zbieżności** (rys. 21.). Kolorem czerwonym oznaczone są linie prowadzące do punktu zbieżności wzdłuż osi **X**, żółtym – wzdłuż osi **Z**, a niebieskim – wzdłuż osi **Y**.

Na listwie właściwości widać wielkości kątów, jakie ustawiono pomiędzy osiami X, Y i Z. Domyślnie pomiędzy osią X i Y ustawiony jest kąt 90°.

Rozmiary bryły możemy zmieniać, nie zmieniając perspektywy. Należy w tym celu przesuwać odpowiednie węzły bryły.

Ćwiczenie 11. Rysujemy prostopadłościan

1. Sprawdź, jakie wymiary bryły zmieniają się, gdy przesuwasz jej wybrane węzły w kierunku poszczególnych punktów zbieżności. Sprawdź też, co się dzieje, gdy podczas zmiany położenia węzłów przytrzymasz klawisz **Shift**, a co, gdy przytrzymasz klawisz **Ctrl**.
2. W programie Inkscape narysuj prostopadłościan.
3. Zapisz rysunek w pliku pod nazwą *Prostopadłościan*.



Rys. 21. *Rysowanie obrazu przestrzennego*

Ćwiczenie 12. Rysujemy w perspektywie

1. W programie Inkscape narysuj w perspektywie aleję obsadzoną z obu stron drzewami.
2. Zapisz rysunek w pliku pod nazwą *Perspektywa*.

4. Przekształcenia

W programie Inkscape obrazy można poddawać różnym przekształceniom, takim jak: skalowanie, przesuwanie, pochylanie, obracanie, wykonywanie odbicia lustrzanego.

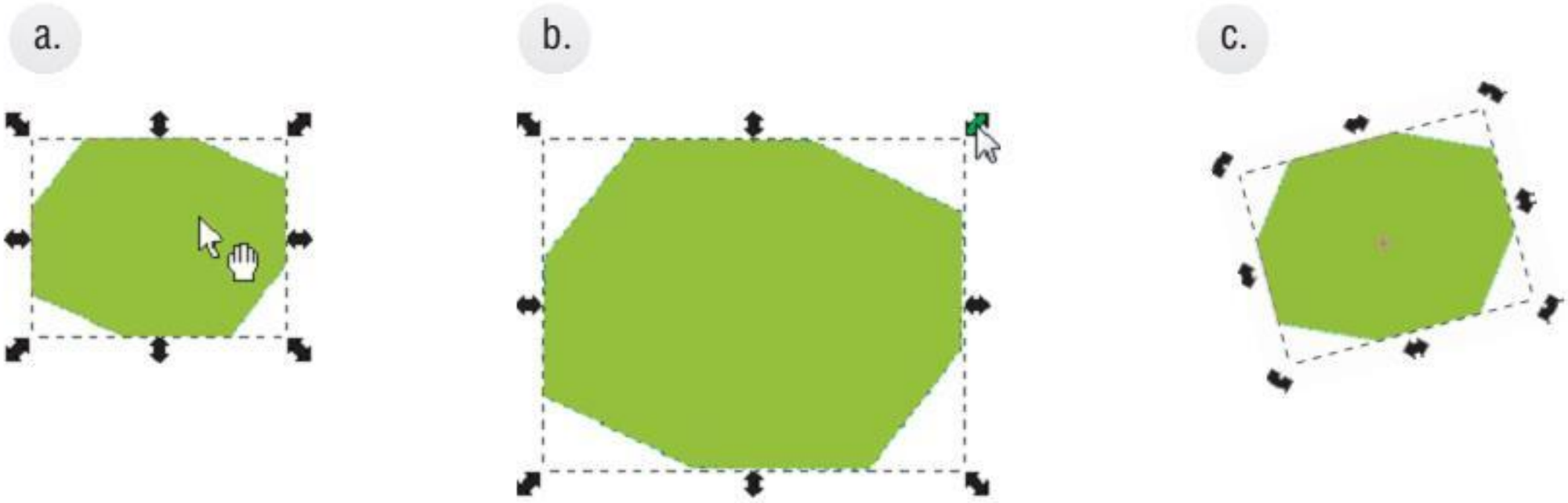
Przed wykonaniem każdej operacji musimy uaktywnić obiekt, klikając narzędzie **Wskaźnik** . Aby przesunąć rysunek, umieszczamy kursor w obszarze uaktywnionego obiektu i przeciągamy obiekt (rys. 22a).

Skalowanie rysunku wykonujemy, przeciągając jedną ze strzałek (rys. 22b). Obraz tworzony w grafice wektorowej nie traci jakości podczas skalowania.

Gdy klikniemy aktywny obiekt, strzałki zmienią postać na pokazaną na rysunku 22c – przeciągając je, obiekt można obracać.

Ćwiczenie 13. Stosujemy przekształcenia

1. W programie Inkscape narysuj obraz podobny do pokazanego na rysunku 22a. Powiększ go.
2. Narysuj podobny obraz w programie Paint i również go wielokrotnie powiększ. Porównaj efekty obydwu powiększeń.
3. Zapisz rysunek utworzony w programie Inkscape w pliku pod wybraną nazwą.



Rys. 22. *Aktywny obraz z widocznymi ośmioma strzałkami: a. przeciąganie, b. skalowanie, c. obracanie*

Ćwiczenie 14. Sprawdzamy efekty skalowania

Sprawdź na figurze utworzonej w ćwiczeniu 13., jaki efekt skalowania uzyskamy, gdy podczas przeciągania strzałki równocześnie wciśniemy i przytrzymamy klawisz **Ctrl**, a jaki, gdy wciśniemy i przytrzymamy klawisz **Shift**.

Jeśli chcemy obrócić obiekt o kąt 90° lub dokonać odbicia lustrzanego, możemy też skorzystać z przycisków umieszczonych na listwie narzędziowej narzędzia **Wskaźnik** . Możemy również wybrać odpowiednią opcję w menu **Obiekt** (rys. 23.).



Rys. 23. *Opcje umożliwiające wykonanie obrotu i odbicia lustrzanego*

Ćwiczenie 15. Sprawdzamy efekty pochylania

Sprawdź, w jaki sposób można pochylić narysowany obiekt. Pochyl figurę narysowaną w ćwiczeniu 13.

5. Warstwy

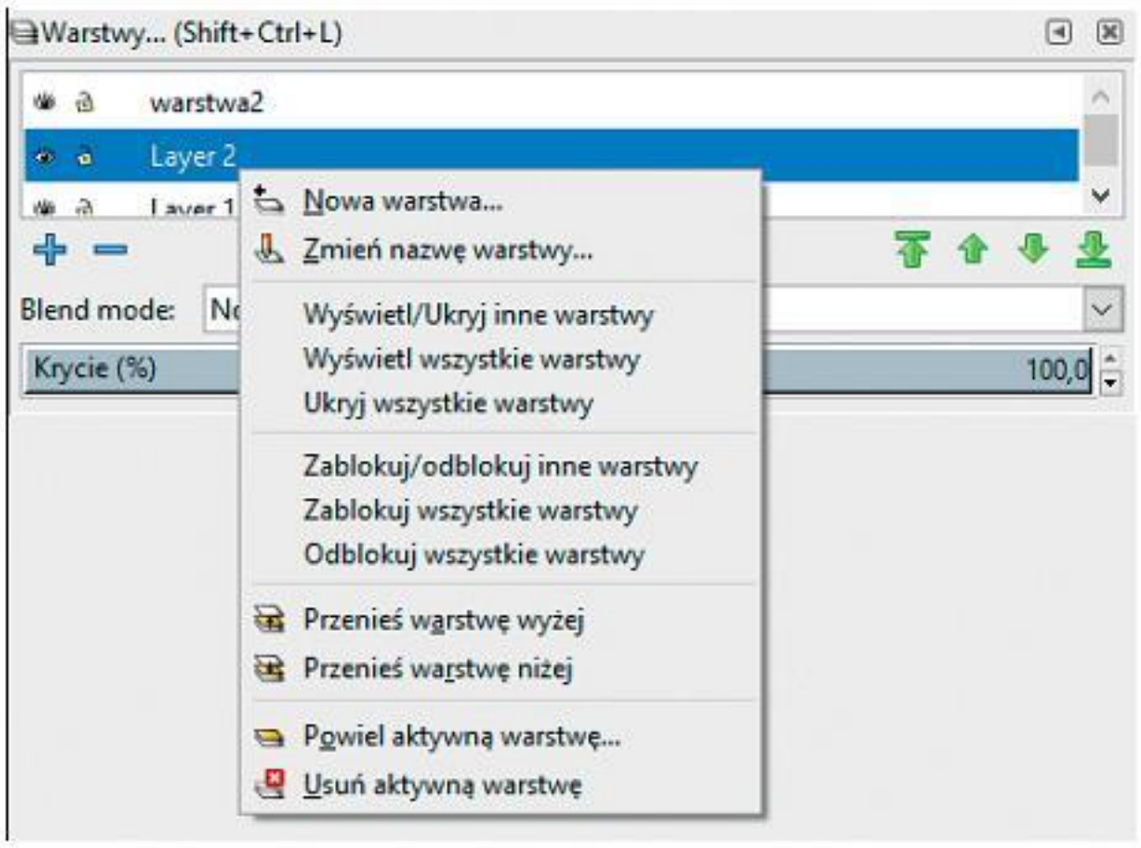
Podobnie jak w programie GIMP, w programie Inkscape można pracować na **warstwach** (menu **Warstwa**).

Korzystając z menu **Warstwy**, można m.in.

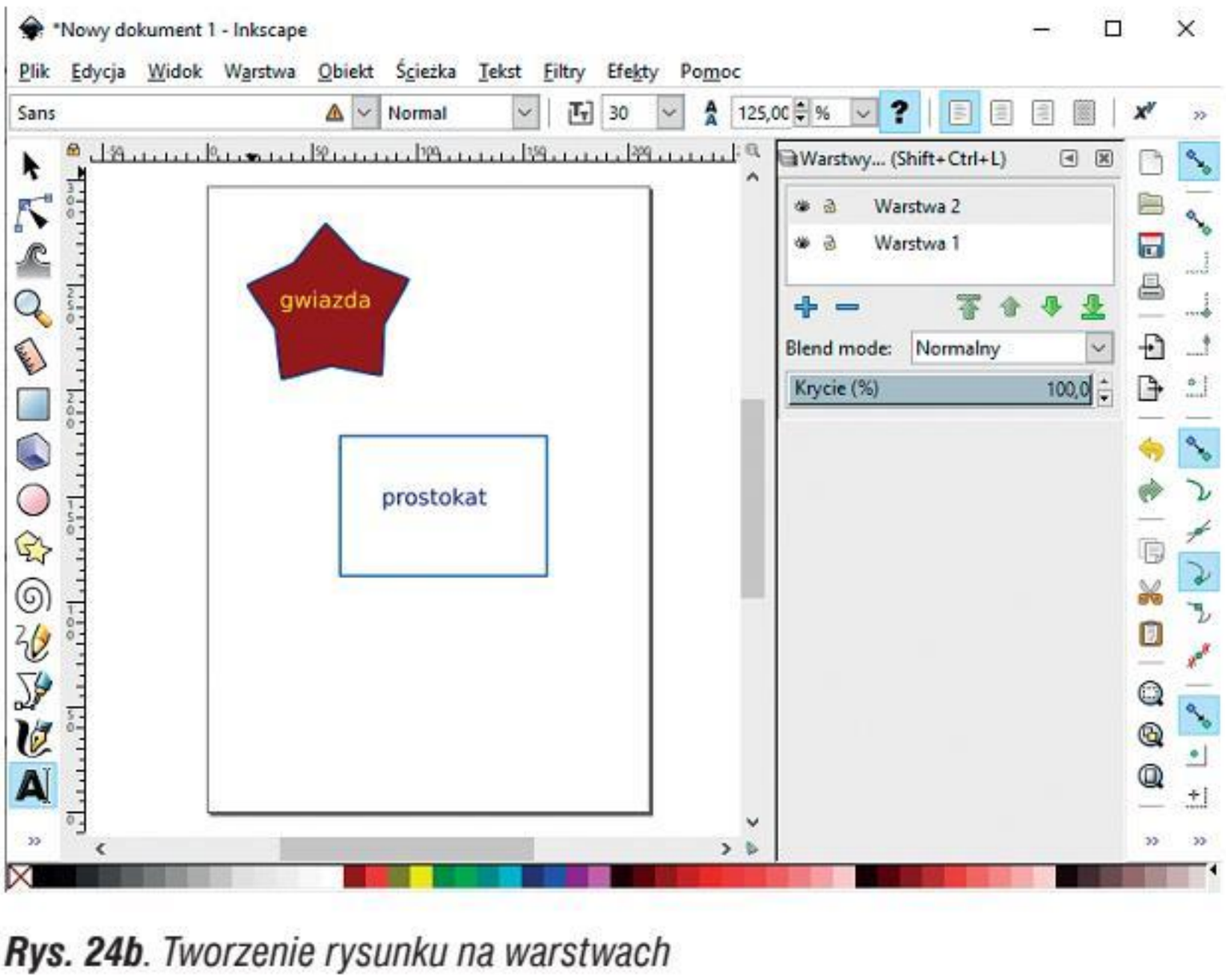
- zmienić nazwę warstwy (**Zmień nazwę warstwy**),
- dodać nową warstwę (**Nowa warstwa**),
- wyświetlić lub ukryć warstwę (**Wyświetl/Ukryj inne warstwy**).

Ukryta warstwa jest oznaczona ikoną , a widoczna .

Warstwy są przydatne podczas rysowania złożonych rysunków. Można np. na każdej warstwie umieścić obiekty o zbliżonych cechach i niezależnie edytować elementy schematu umieszczone na różnych warstwach.



Rys. 24a**. Okno **Warstwy



***Rys. 24b**. Tworzenie rysunku na warstwach*

Ćwiczenie 16. Tworzymy rysunek na warstwach

1. Narysuj na pierwszej warstwie okrąg, wielokąt, gwiazdę, prostokąt, obiekt 3D, a na drugiej umieść podpisy – nazwy narzędzi programu Inkscape, za pomocą których zostały te figury utworzone.
2. Zapisz rysunek w pliku pod wybraną nazwą.

Warto zapamiętać

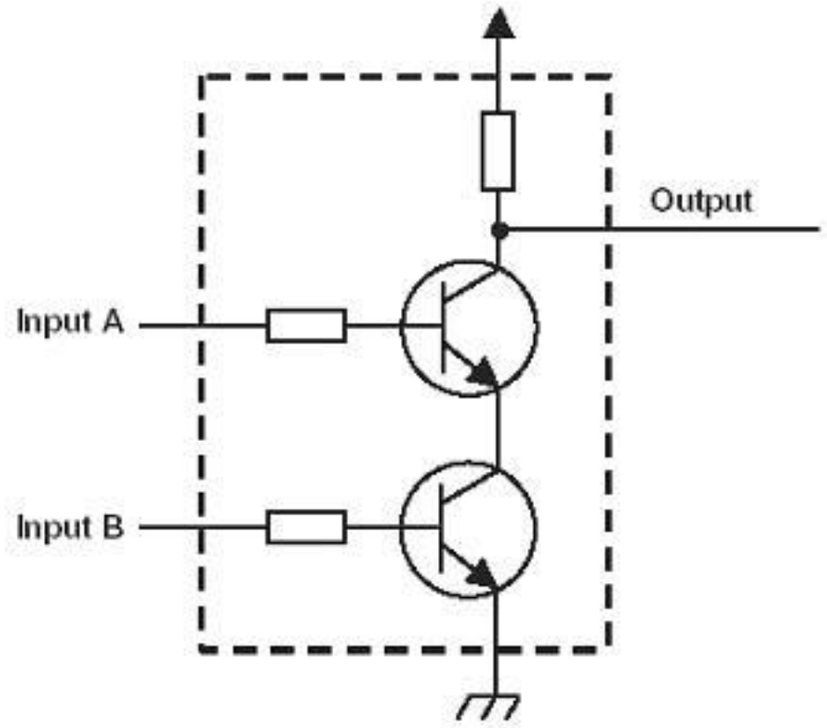
- Inkscape jest przykładem programu umożliwiającego opracowywanie grafiki wektorowej.
- Jedną z cech charakterystycznych grafiki wektorowej jest możliwość dowolnego skalowania obrazów bez utraty jakości.
- Program Inkscape ma wiele możliwości; w temacie omówiliśmy:
  - tworzenie figur geometrycznych, w tym rzutów brył,
  - rysowanie krzywych, w szczególności krzywych Béziera,
  - modyfikowanie krzywych poprzez zmianę położenia węzłów,
  - skalowanie obiektów,
  - przesuwanie, obracanie, pochylanie, odbijanie w pionie i w poziomie.
- Program Inkscape umożliwia pracę na warstwach obrazu.

Pytania, problemy

1. Jakie narzędzia programu Microsoft Word umożliwiają tworzenie grafiki wektorowej?
2. W jaki sposób można edytować punkty w kształtach?
3. W jaki sposób można w programie Inkscape narysować wycinek koła?
4. W jaki sposób można zmienić kolor konturu i wypełnienia figury?
5. Czym różni się rysowanie ścieżek za pomocą **Ołówka** od rysowania ich za pomocą **Pióra**?
6. W jaki sposób rysuje się krzywe Béziera?
7. W jaki sposób można edytować węzły na ścieżce?
8. W jaki sposób można w programie Inkscape narysować sześcian?
9. Jakie przekształcenia obrazu można zastosować w programie Inskscape?

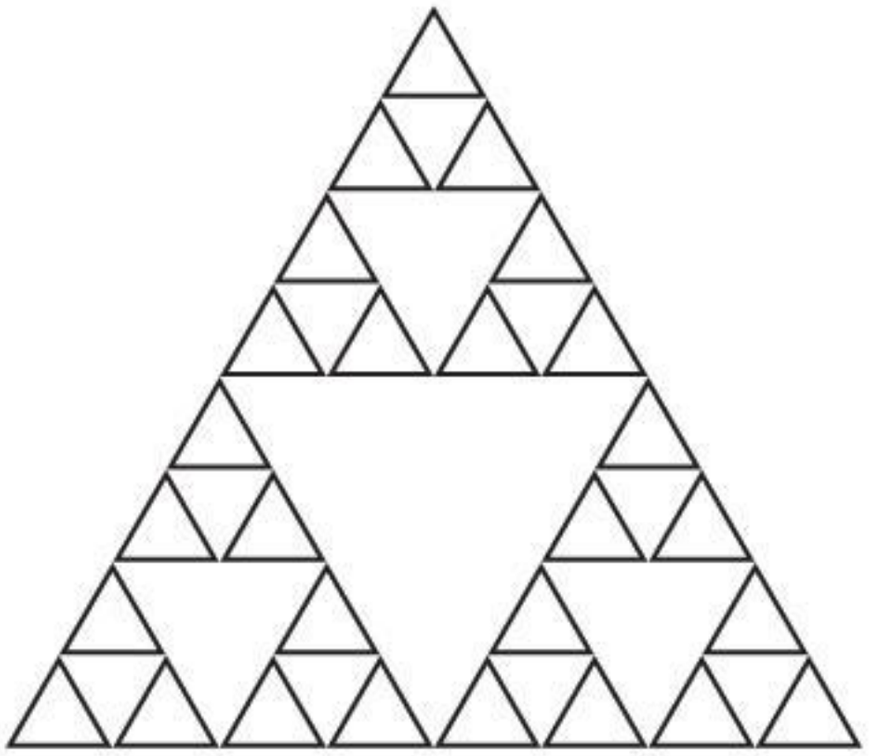
Zadania

1. W edytorze rysunków programu Microsoft Word narysuj flagę Polski, flagę innego wybranego kraju i flagę Unii Europejskiej.
   Wskazówka: Flagę Polski skomponuj z dwóch prostokątów i wykorzystaj grupowanie obiektów.
2. Korzystając z kształtu **Bazgroły** , narysuj figurę według własnego pomysłu. Przekształć ją, edytując punkty w kształcie.
3. Korzystając z wybranego programu, narysuj schemat pokazany na rysunku 25.
   Wskazówki:
   - Pamiętaj o zgrupowaniu obiektów.
   - W programie Inkscape można do tego celu użyć (po zaznaczeniu obiektów) skrótu klawiaturowego **Ctrl+G** lub narzędzia **Grupuj zaznaczone obiekty** (*Group selected objects*), umieszczonego na pasku narzędzi ogólnych.
   - W programie Inkscape możesz wykonać rysunek, stosując pracę na warstwach.



Rys. 25. *Schemat do zadania 3.*

4. Korzystając z narzędzi rysowania i kształtów w programie Microsoft Word, narysuj trójkąt Sierpińskiego podobny do pokazanego na rysunku 26.
   Wskazówka: Aby ułatwić sobie rozmieszczanie obiektów, można skorzystać z **siatki** (**Widok/Linie siatki**). Siatka pełni funkcję podobną do linii w zeszycie w kratkę – po włączeniu przyciągania do siatki rozmiary i położenie obiektów będą musiały być wielokrotnością rozmiaru pojedynczej „kratki". Polecenia dotyczące siatki znajdziesz po wybraniu opcji **Wyrównaj** na karcie **Formatowanie kształtu**. Jeżeli po włączeniu siatki chcemy umieścić obiekt poza „kratkami", należy podczas wykonywania operacji na obiekcie trzymać wciśnięty klawisz **Alt**.



Rys. 26. *Trójkąt Sierpińskiego do zadań 4. i 12.*

5. Zapoznaj się z narzędziem **Kaligrafia**, a następnie zapisz za jego pomocą stwierdzenie Seymoura Paperta (twórcy języka Logo): *Istotą działalności intelektualnej jest stała korekta drogi, którą się podąża*.
6. Narysuj **Piórem** odręczną krzywą i, zmieniając położenie węzłów, wykonaj kompozycję według własnego pomysłu.
7. Korzystając z możliwości przekształceń, utwórz symetryczną kompozycję według własnego pomysłu.
8. Utwórz projekt logo firmy. Zakres działalności firmy pozostawiamy twojej wyobraźni. Wykorzystaj pracę na warstwach.
9. Narysuj ostrosłup o podstawie czworokątnej.
10. Narysuj w perspektywie ulicę, przy której stoi twój dom. Na rysunku możesz umieścić także inne budynki i drzewa.

Dla zainteresowanych

11. W programie Inkscape istnieje możliwość klonowania. Korzystając z tej możliwości oraz różnych przekształceń, przygotuj projekt witrażu.
12. Korzystając z programu Inkscape, narysuj trójkąt Sierpińskiego podobny do pokazanego na rys. 26.
13. W programie Inkscape zaprojektuj logo szkolnego klubu sportowego. Wykorzystaj pracę na warstwach.
14. Zapoznaj się z dodatkowymi operacjami wykonywanymi na ścieżkach i stwórz rysunek z wykorzystaniem poznanych możliwości.
15. Dowiedz się, w jaki sposób wykonuje się w programie Inkscape maskowanie i stwórz rysunek z wykorzystaniem poznanych możliwości.

Wokół algorytmiki i programowania

C

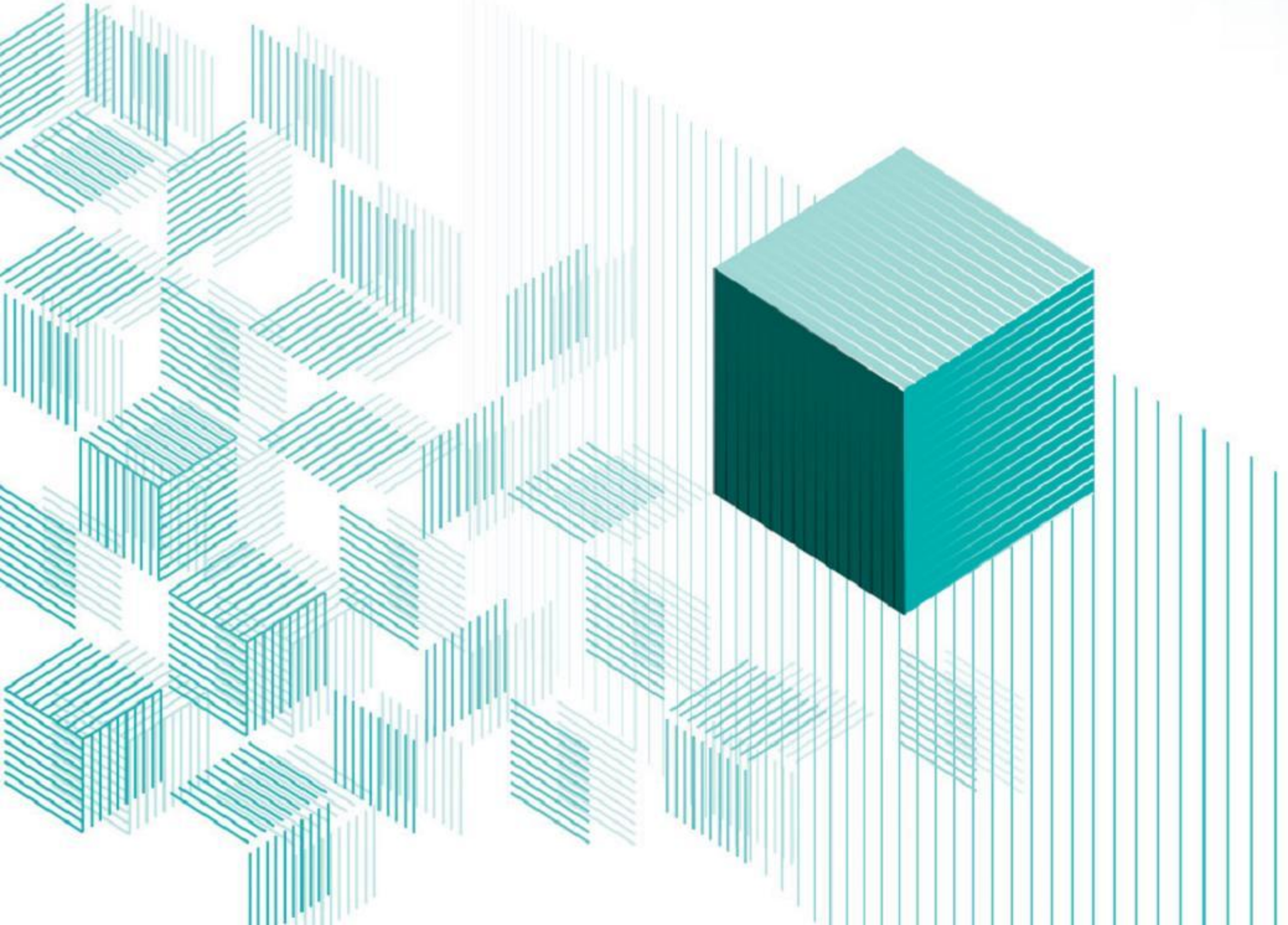
Moduł

Temat C1. Więcej o instrukcjach iteracyjnych w językach C++ i Python

Temat C2. Stosowanie funkcji w językach C++ i Python

Temat C3. Stosowanie tablic w języku C++ i list w języku Python

Temat C4. Wybrane algorytmy na liczbach naturalnych i algorytmy porządkowania



Temat C1. Więcej o stosowaniu instrukcji iteracyjnych w językach C++ i Python

1. Stosowanie instrukcji iteracyjnej `for` w językach C++ i Python – powtórzenie
2. Stosowanie instrukcji `while` w językach C++ i Python
3. Stosowanie instrukcji `do ... while` w języku C++

Warto powtórzyć

1. Na czym polega iteracja?
2. Jak można zapisać algorytm iteracyjny w wybranym języku programowania (C++ lub Python)?
3. Jak określamy liczbę iteracji w instrukcji **`for`** w wybranym języku programowania (C++ lub Python)?
4. W jaki sposób określa się krok iteracji w wybranym języku programowania (C++ lub Python)?
5. Jak działa instrukcja warunkowa w wybranym języku programowania (C++ lub Python)?
6. W jaki sposób oznaczamy blok instrukcji w wybranym języku programowania (C++ lub Python)?

1. Stosowanie instrukcji iteracyjnej `for` w językach C++ i Python – powtórzenie

Zbierasz pieniądze na zakup nowego komputera. Wpłacasz co miesiąc dowolną kwotę na swoje konto. Chcesz obliczyć sumę pieniędzy wpłaconych w ciągu roku. Jak zapisać rozwiązanie tego zadania w wybranym języku programowania?

W przedstawionym problemie powtarzają się dwie czynności: wprowadzania wpłacanej kwoty i powiększania sumy pieniędzy o tę kwotę. W językach programowania do zapisania powtarzających się poleceń wykorzystujemy instrukcje iteracyjne (zwane również instrukcjami pętli).

Aby zapisać algorytm iteracyjny w językach C++ i Python, stosujemy instrukcję iteracyjną **`for`**. W tworzonych przez nas programach liczba powtórzeń (iteracji) będzie z góry określona lub wprowadzana z klawiatury po uruchomieniu programu.

Instrukcja iteracyjna **for**	**for** *(wyrażenie_początkowe; warunek; wyrażenie_pętli)* *lista_instrukcji;* *kolejna_instrukcja;*	C++
	for *zmienna* **in** *lista_wartości:* *lista_instrukcji* *kolejna_instrukcja*	Python

Tabela 1. *Ogólna postać instrukcji iteracyjnej* **for** *w językach C++ i Python*

Lp.	C++	Python
1.	`for (i = 0; i < 50; i++)` `  cout << i << endl;`	`for i in range(50):` `  print(i)`
2.	`for (i = 0; i <= n; i++)` `  cout << "#" << endl;`	`for i in range(n+1):` `  print("#")`
3.	`for (i = 1; i < k; i++)` `{` `  cout << "Podaj liczbe: ";` `  cin >> a;` `  cout << a * a;` `}`	`for i in range(1, k):` `  a = int(input("Podaj liczbę: "))` `  print(a * a)`
4.	`for (i = 100; i > 0; i--)` `  cout << i << endl;`	`for i in range(100, 0, -1):` `  print(i)`

Tabela 2. *Przykłady stosowania instrukcji iteracyjnej* **for** *w językach C++ i Python*

Ćwiczenie 1. Analizujemy działanie instrukcji iteracyjnej **for**

1. Przeanalizuj zapisy podane w tabeli 2., a następnie odpowiedz na pytania:
   a. Jakie wartości przyjmuje zmienna `i`? C+ i Python
   b. Ile razy zostanie wykonana *lista_instrukcji*? C+ i Python
   c. Co oznacza zapis `i++` i `i--`? C++
   d. Co określają parametry funkcji `range()`? Python

Ćwiczenie 2. Stosujemy instrukcję iteracyjną **for**

1. Napisz specyfikację zadania i program, który wyświetli w kolumnie liczby naturalne od 1 do *n*, i dalej, w tej samej kolumnie – liczby od *n* do 1.
2. Zapisz program w pliku pod nazwą *Liczby _rosnaco_malejaco*.

Uwaga

W języku C++ należy pamiętać o skompilowaniu programu przed jego uruchomieniem – program nie zostanie uruchomiony, jeśli kompilacja nie przebiegła prawidłowo.

Ćwiczenie 3. Obliczamy sumę dwunastu liczb wprowadzanych z klawiatury

Napisz specyfikację zadania i program, który obliczy sumę dwunastu comiesięcznych wpłat na zakup komputera i wyświetli wynik na ekranie. Zapisz program w pliku pod nazwą *Suma_12*.

Ćwiczenie 4. Obliczamy sumę *n* liczb wprowadzanych z klawiatury

1. Zmodyfikuj program zapisany w ćwiczeniu 3. tak, aby liczba wpłat była wprowadzana z klawiatury po uruchomieniu programu.
2. Zapisz program w pliku pod nazwą *Suma_n*.

2. Stosowanie instrukcji `while` w językach C++ i Python

Chcesz dalej oszczędzać na zakup nowego komputera, ponieważ potrzebna jest większa kwota. Zamierzasz wpłacać pieniądze, dopóki suma wpłat nie przekroczy potrzebnej kwoty. Jak przedstawić ten problem w wybranym języku programowania?

Nie wiemy z góry, ilu wpłat musimy dokonać, aby osiągnąć lub przekroczyć kwotę potrzebną na zakup komputera. Liczba iteracji zależy w tym przypadku od spełnienia warunku przekroczenia założonej kwoty.

Do zapisywania algorytmów iteracyjnych, w których liczba iteracji nie jest z góry określona, możemy zastosować następujące instrukcje iteracyjne:

- **`while`** – w języku C++ i w języku Python,
- **`do ... while`** – w języku C++.

Instrukcja iteracyjna **`while`**	**`while`** *(warunek)* *lista_instrukcji;* *kolejna_instrukcja;*	C++
	`while` *warunek:* *lista_instrukcji* *kolejna_instrukcja*	Python

Tabela 3. *Ogólna postać instrukcji iteracyjnej* **`while`** *w językach C++ i Python*

Działanie pętli **`while`** jest takie samo w obydwu językach (C++ i Python): najpierw sprawdzany jest *warunek*; jeśli jest spełniony, to wykonywana jest *lista_instrukcji*. Wewnątrz bloku *lista_instrukcji* powinna być zawsze umieszczona instrukcja, która zmienia wartość *warunku* – w przeciwnym wypadku pętla nigdy się nie zakończy. W szczególnej sytuacji, gdy *warunek* od razu nie jest spełniony, *lista_instrukcji* w ogóle nie zostanie wykonana. *Lista_instrukcji* może zawierać jedną lub wiele instrukcji.

Przykład 1. Stosowanie instrukcji `while` w językach C++ i Python

Zadanie: Napisz program obliczający sumę wpłat aż do osiągnięcia lub przekroczenia założonej kwoty.

Dane: liczba rzeczywista dodatnia *kwota*, oznaczająca kwotę potrzebną na zakup komputera, ciąg dowolnych liczb rzeczywistych dodatnich *wplata*, oznaczających kolejne comiesięczne wpłaty.

Wynik: liczba rzeczywista dodatnia *suma_wplat*, oznaczająca wartość sumy wszystkich wpłat.

C++

Zbiorka.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    float kwota, wplata, suma_wplat;

    cout << "Podaj potrzebna kwota: ";
    cin >> kwota;
    suma_wplat = 0;
    while(suma_wplat < kwota)
    {
        cout << "Wprowadz wplate: ";
        cin >> wplata;
        suma_wplat = suma_wplat + wplata;
    }
    cout << "Suma wplat wynosi: " << suma_wplat << " zl";

    return 0;
}
```

Python

Zbiorka.py - C:/Podręcznik IP Klasa II/Moduł C/Temat C1_zrzuty/Zbiorka.py (3.7.2)

File Edit Format Run Options Window Help

```python
kwota = float(input("Podaj potrzebną kwotę: "))
suma_wplat = 0

while suma_wplat < kwota:
    wplata = float(input("Wprowadź wpłatę: "))
    suma_wplat = suma_wplat + wplata

print("Suma wpłat:", suma_wplat, "zł")

input("\n\nAby zakończyć, naciśnij Enter")
```

Ln: 13 Col: 0

Ćwiczenie 5. Analizujemy, testujemy i modyfikujemy gotowy program

1. Przepisz wybrany program z przykładu 1. Zapisz program w pliku pod nazwą *Zbiorka*.
2. Uruchom i przetestuj program dla kilku różnych wartości zmiennych. Wyjaśnij znaczenie poszczególnych wierszy programu.
3. Zmodyfikuj program tak, aby na ekranie wyświetlała się również bieżąca suma, uzyskana po każdej wpłacie, a na koniec – różnica pomiędzy kwotą wpłaconą a założoną (nadpłata). Dodaj odpowiednie komunikaty. Zapisz plik pod tą samą nazwą.

Ćwiczenie 6. Sprawdzamy poprawność danych

1. Dodaj do programu zapisanego w ćwiczeniu 5. sprawdzanie poprawności wprowadzanych wartości zmiennych *kwota* i *wplata*, zgodnie ze specyfikacją zadania (podaną w przykładzie 1.). Wprowadzanie i sumowanie liczb powinny być wykonywane dla poprawnych wartości zmiennych *kwota* i *wplata*:
   a. jeśli użytkownik wprowadzi niepoprawną wartość zmiennej *kwota*, wyświetl na ekranie komunikat „Wprowadzono bledna liczbe" i zakończ program.
   b. jeśli użytkownik wprowadzi niepoprawną wartość zmiennej *wplata*, wyświetl na ekranie komunikat „Wprowadzono bledna liczbe" i zignoruj wprowadzoną wartość.
2. Zapisz plik pod tą samą nazwą. Uruchom program i sprawdź jego działanie dla różnych wartości zmiennych.

Ćwiczenie 7. Stosujemy instrukcję iteracyjną `while`

1. Napisz specyfikację zadania i program, który będzie obliczał sumę liczb całkowitych wprowadzanych z klawiatury aż do wprowadzenia zera. Wprowadzenie zera kończy zliczanie. Wynik sumowania wyprowadź na ekran.
2. Zapisz program w pliku pod nazwą *Suma_while*.

Wskazówki: Na początku programu przypisz zmiennej *liczba* wartość różną od zera, np. 1, a zmiennej *suma* – 0.

C++ Początkową wartość można przypisać zmiennej podczas jej deklaracji:

```
int liczba = 1, suma = 0;
```

Python

```
liczba = 1
suma = 0
```

3. Stosowanie instrukcji `do ... while` w języku C++

W języku C++ występuje instrukcja iteracyjna: **do** … **while**, którą możemy zastosować w przypadku, gdy nie znamy liczby iteracji.

Instrukcja iteracyjna **do** … **while**	**do** *lista _ instrukcji;* **while**(*warunek);* *kolejna_instrukcja;*	C++

Tabela 4. *Ogólna postać instrukcji iteracyjnej* **do** … **while** *w języku C++*

W instrukcji **do** … **while** najpierw jest wykonywana *lista_instrukcji*, a dopiero potem jest sprawdzany *warunek*.

Instrukcje w pętli (*lista_instrukcji*) są wykonywane, dopóki warunek jest prawdziwy. Jeśli warunek jest fałszywy – wykonywana jest *kolejna_ instrukcja*. Niezależnie od wartości początkowej warunku, *lista_instrukcji* zostanie wykonana przynajmniej raz. Jako *lista_instrukcji* może wystąpić pojedyncza instrukcja (w tym instrukcja **do** … **while**) lub więcej instrukcji ujętych w blok {}.

Przykład 2. Stosowanie instrukcji `do ... while` w języku C++

Zadanie: Napisz program zliczający znaki wprowadzane z klawiatury aż do wprowadzenia kropki. Wprowadzenie kropki kończy zliczanie. Wyprowadź na ekran odpowiedni komunikat i liczbę wprowadzonych znaków łącznie z kropką.

Dane: dowolny znak: *znak*.

Wynik: liczba całkowita *liczba_znakow*, oznaczająca liczbę znaków wprowadzonych z klawiatury.

Uwagi:

- Pojedynczy znak umieszczamy w programie w pojedynczych górnych cudzysłowach.
- Instrukcja `liczba_znakow++;` to odpowiednik instrukcji:
  `liczba_znakow = liczba_znakow + 1;`

C++ Zliczanie_znakow.cpp

```cpp
#include <iostream>
using namespace std;

int main()
{
    char znak;
    int liczba_znakow = 0;

    do
    {
        cout << "Wprowadz znak: ";
        cin >> znak;
        liczba_znakow++;
    }
    while(znak != '.');
    cout << "Liczba wprowadzonych znakow wynosi: " << liczba_znakow;
    return 0;
}
```

Ćwiczenie 8. Analizujemy, testujemy i modyfikujemy gotowy program

1. Przepisz program z przykładu 2. Zapisz program w pliku pod nazwą *Zliczanie_znakow*.
2. Uruchom i przetestuj program dla kilku różnych wartości zmiennych. Wyjaśnij znaczenie poszczególnych wierszy programu.
3. Zmodyfikuj program tak, aby nie zliczał kropki. Zapisz program w pliku pod nazwą *Zliczanie_znakow_bez_kropki*.

Ćwiczenie 9. Stosujemy instrukcję iteracyjną `do ... while` w języku C++

1. Napisz specyfikację zadania i program, który będzie obliczał różnicę dwóch liczb całkowitych (*odjemna*, *odjemnik*) wprowadzanych z klawiatury, dopóki wynik odejmowania (*roznica*) będzie dodatni. Wynik niedodatni kończy obliczenia. Zliczaj wykonane działania odejmowania, których wynik jest dodatni, i wyprowadź ich liczbę na ekran.
2. Zapisz program w pliku pod nazwą *Roznice*.

Wskazówka: W warunku logicznym instrukcji `do ... while` sprawdzaj, czy wynik odejmowania (*roznica*) jest dodatni.

Warto zapamiętać

- W językach programowania występują instrukcje iteracyjne, które umożliwiają zapis algorytmów iteracyjnych, np.:
  - **`for`** – w językach C++ i Python,
  - **`while`** – w językach C++ i Python,
  - **`do ... while`** – w języku C++.

  Zwykle gdy znamy liczbę iteracji, stosujemy instrukcję **`for`**; jeśli nie wiemy, ile będzie powtórzeń – pozostałe dwie.
- W instrukcji **`while`** najpierw jest sprawdzany warunek logiczny, a potem (gdy warunek jest prawdziwy) są wykonywane instrukcje znajdujące się wewnątrz instrukcji **`while`**. Gdy warunek od razu nie jest spełniony, instrukcje wewnątrz instrukcji **`while`** nie zostaną wykonane ani razu.
- Instrukcja **`do ... while`** różni się od **`while`** tym, że warunek jest sprawdzany po bloku instrukcji umieszczonym pomiędzy słowami kluczowymi **`do`** i **`while`**, czyli te instrukcje zostaną wykonane przynajmniej raz.

Pytania i polecenia

1. Jakie instrukcje w językach programowania umożliwiają zapis algorytmów iteracyjnych?
2. Wyjaśnij na przykładzie wybranego języka programowania, jak działa instrukcja **`while`**.
3. Czy zostaną wykonane instrukcje wewnątrz instrukcji **`while`**, jeśli warunek jest prawdziwy? Uzasadnij odpowiedź.
4. Czym różni się działanie instrukcji **`while`** od **`do ... while`**?

Zadania

1. Do programu zapisanego w ćwiczeniu 4. dodaj sprawdzanie poprawności danych: wprowadzane kwoty i liczba wpłat powinny być liczbami dodatnimi (liczba wpłat jest liczbą naturalną). Zapisz plik pod tą samą nazwą.
2. Znajdź w Internecie wartość średniej temperatury w Polsce w poszczególnych miesiącach – w dzień i w nocy. Napisz program obliczający na podstawie tych danych średnie roczne temperatury w dzień i w nocy. Wyniki wyświetl na ekranie. Zapisz program w pliku pod nazwą *Temperatury*.
3. Napisz program obliczający i wyświetlający na ekranie iloraz dwóch liczb całkowitych (*dzielna*, *dzielnik*) wprowadzanych z klawiatury aż do wprowadzenia dzielnika równego zero. Ponieważ nie da się obliczyć ilorazu dla dzielnika równego zero, program powinien wyświetlić odpowiedni komunikat. Zapisz program w pliku pod nazwą *Iloraz*.
4. Napisz program obliczający i wyświetlający na ekranie objętość sześcianu o boku *bok* aż do wprowadzenia niedodatniej wartości zmiennej *bok*. Zapisz program w pliku pod nazwą *Objetosc_szescianu*.
5. Napisz specyfikację zadania i program zliczający, ile jest liczb podzielnych przez 3 pośród liczb całkowitych wprowadzanych z klawiatury. W przypadku wprowadzenia zera program ma zakończyć działanie. Zapisz program w pliku pod nazwą *Podzielne_przez_3*.

Dla zainteresowanych

6. Masz *n* zł na kupienie upominków dla znajomych. Upominki możesz kupować, dopóki starczy ci pieniędzy. Ceny upominków wprowadzaj z klawiatury. Nie możesz przekroczyć kwoty *n* zł ani wydać mniej. Jeśli cena ostatniego upominku przekroczy kwotę *n*, musisz kupić coś tańszego. Program kończy się, gdy wydasz dokładnie *n* zł. Napisz odpowiedni program i zapisz go w pliku pod nazwą *Upominki*.
7. Napisz program obliczający część całkowitą pierwiastka kwadratowego z podanej liczby naturalnej *n*.
   Wskazówka: Istnieją różne algorytmy obliczania pierwiastka. Można na przykład obliczać kwadraty kolejnych liczb naturalnych aż do uzyskania liczby większej lub równej *n*.



Przeczytaj jeśli chcesz wiedzieć więcej…

W języku Python można przypisać wiele wartości wielu zmiennym, umieszczając nazwy zmiennych i ich wartości w jednej instrukcji przypisania, np.:

```
liczba, suma = 1, 0
```

co odpowiada instrukcjom:

```
liczba = 1
suma = 0.
```

W języku Python instrukcja **`while`** może występować ze słowem kluczowym **`else`**. Jest ona jednak używana bardzo rzadko. W przypadku użycia **`else`** *kolejna_instrukcja* zostanie wykonana po uprzednim wykonaniu *listy_instrukcji2*.

Instrukcja iteracyjna **`while`** z klauzulą **`else`**	**`while`** *warunek:* *lista_instrukcji1* **`else`***:* *lista_instrukcji2* *kolejna_instrukcja*	Python

Tabela 5. *Ogólna postać instrukcji iteracyjnej* **`while`** *z klauzulą else w języku Python*

Temat C2. Stosowanie funkcji w językach C++ i Python

1. Funkcje w językach C++ i Python
2. Funkcje zwracające wartość w językach C++ i Python
3. Funkcje niezwracające wartości w językach C++ i Python

Warto powtórzyć

1. W jaki sposób deklarujemy zmienne w języku C++?
2. Co w języku C++ określa typ zmiennej?
3. W jaki sposób nadajemy wartość zmiennej w wybranym języku programowania (C++ lub Python)?
4. W jaki sposób wyświetlamy wyniki na ekranie komputera w wybranym języku programowania (C++ lub Python)?
5. W jakich środowiskach programowania tworzyliśmy programy z wykorzystaniem podprogramów?

1. Funkcje w językach C++ i Python

Do magazynu sprzętu RTV dowożone są kilkakrotnie w ciągu miesiąca różne artykuły.

Jak napisać program zliczający, ile sztuk każdego artykułu dostarczono we wszystkich dostawach łącznie?

Podprogram (procedura lub funkcja) **P**

Wyodrębniona część programu, mająca unikatową nazwę i ustalony sposób wymiany danych z innymi częściami programu.

Jeśli, wykonując zadanie, musimy kilkakrotnie zsumować liczby w różnych zbiorach, możemy opracować cząstkowe rozwiązanie obliczające sumę, a potem wielokrotnie wykorzystać je w programie. Do opracowywania problemów cząstkowych, które są częściami zadania rozwiązywanego przez dany program, służą **podprogramy**.

Podprogram realizujący określone zadanie (np. sumowanie liczb) może być wielokrotnie wykorzystywany – w tym samym programie lub w innych. Stosowanie podprogramów zwiększa przejrzystość programu.

W językach C++ i Python wszystkie podprogramy nazywamy **funkcjami**.

Poznaliśmy kilka wbudowanych funkcji Pythona, m.in.: `print()`, `input()`, a w języku C++ funkcję `main()` stanowiącą główny program. W językach C++ i Python można definiować własne funkcje.

Funkcje mogą **zwracać wartość** do programu głównego lub nic nie zwracać, mogą mieć określone **parametry** lub ich nie posiadać.

Funkcje dzielimy na: zwracające wartość i niezwracające wartości.

Nazwy funkcji tworzymy na takich samych zasadach jak nazwy zmiennych (można używać tylko liter, znaków podkreślenia i cyfr; nazwa nie może zaczynać się od cyfry; przyjęte jest niestosowanie polskich liter). Funkcjom należy nadawać nazwy opisujące ich działanie. Ważne jest, aby przyjąć jednolity sposób nazewnictwa, np. w nazwach funkcji i zmiennych używa się wyłącznie małych liter, a w nazwach wieloczłonowych stosuje się znak podkreślenia, np. *liczba_dostaw*, *suma_sztuk*, *wprowadz_dane*, *wyprowadz_dane*, *najwiekszy*.

P **Parametr**

Wartość przekazywana do funkcji.

W **Wartość zwracana**

Wartość przekazywana z funkcji do miejsca wywołania (np. programu głównego).

Aby zastosować funkcję, trzeba ją **zdefiniować**, a następnie **wywołać** w programie.

2. Funkcje zwracające wartość w językach C++ i Python

Funkcję zwracającą wartość stosujemy, jeśli celem funkcji jest obliczenie i zwrócenie pewnej wartości do programu. Funkcja musi zawierać instrukcję **`return`** z wartością zwracaną do miejsca wywołania (np. programu głównego).

Funkcja zwracająca wartość	Nagłówek funkcji *opis_typu nazwa_funkcji(lista_parametrów_formalnych)* { *lista_instrukcji;* **return** wartość; } Treść funkcji	C++
	Nagłówek funkcji **def** *nazwa_funkcji(lista_parametrów_formalnych)*: *lista_instrukcji* **return** wartość Treść funkcji	Python

***Tabela 1.** Ogólna postać definicji funkcji zwracającej wartość w językach C++ i Python*

Miejsce definiowania funkcji	Funkcja może zostać zdefiniowana przed jej pierwszym wywołaniem lub, jeżeli zostanie zadeklarowana (poprzez umieszczenie nagłówka funkcji) przed funkcją `main()`, w dowolnym miejscu programu – ale poza funkcją `main()`.	C++
	Funkcję definiujemy w dowolnym miejscu programu (zazwyczaj na początku), ale przed pierwszym użyciem.	Python
Określanie typu funkcji i parametrów	Należy wskazać typ funkcji (typ zwracanej wartości) i określić typy oraz nazwy parametrów formalnych.	C++
	Nie określamy typu funkcji ani typów parametrów formalnych funkcji – interpreter rozpozna je automatycznie (podobnie jak w przypadku zmiennych).	Python
Treść funkcji	Treść (*lista_instrukcji* i instrukcja **`return`**) musi się znaleźć w bloku **`{}`**.	C++
	Treść funkcji (*lista_instrukcji* i instrukcja **`return`**) musi być przesunięta w prawo przynajmniej o jedną spację względem nagłówka funkcji (przyjęte jest wcięcie składające się z czterech spacji).	Python

Tabela 2. *Zapis funkcji w językach C++ i Python*



Wywołanie funkcji powoduje wykonanie instrukcji będących treścią funkcji i zwrócenie wyniku jej działania. W momencie wywołania funkcji parametry formalne są zastępowane przez parametry aktualne.

Uwaga

W przypadku większej liczby parametrów oddzielamy je przecinkami.



Aby wywołać funkcję, należy podać nazwę funkcji, a w nawiasach – wartości parametrów. Wartość zwróconą przez funkcję możemy np. przypisać zmiennej w programie głównym. W przypadku definiowania i wywoływania funkcji bez parametrów nawiasy po nazwie funkcji pozostawiamy puste.

Sposób definiowania i wywołania funkcji zwracającej wartość pokażemy na przykładzie funkcji z jednym parametrem, obliczającej sumę *n* liczb.



Przykład 1. Definiowanie i wywołanie funkcji zwracającej wartość z jednym parametrem w językach C++ i Python

Zadanie: Do pewnego magazynu sprzętu RTV dowożone są kilkakrotnie w ciągu miesiąca telewizory i głośniki. Utwórz program obliczający i wyprowadzający na ekran łączne wielkości dostaw (transportów) telewizorów i głośników. Liczby transportów telewizorów (*l_dostaw_tv*) i głośników (*l_dostaw_glosnikow*) wprowadzaj z klawiatury po uruchomieniu programu.
Na początku programu zdefiniuj funkcję *suma* z jednym parametrem, obliczającą i zwracającą sumę *n* liczb wprowadzanych z klawiatury. Parametrem przekazywanym do funkcji jest liczba elementów zbioru *n*. Funkcję *suma* wywołaj dwukrotnie w programie głównym z parametrami aktualnymi *l_dostaw_tv* i *l_dostaw_glosnikow*.

Dane: liczby naturalne większe od zera *l_dostaw_tv* i *l_dostaw_glosnikow*, oznaczające odpowiednio liczbę dostaw telewizorów i głośników,
l_dostaw_tv i *l_dostaw_glosnikow* liczb naturalnych większych od zera oznaczających odpowiednio wielkości kolejnych dostaw telewizorów i głośników.

Wyniki: łączna liczba telewizorów z *l_dostaw_tv* transportów: *suma_tv*, łączna liczba głośników z *l_dostaw_glosnikow* transportów: *suma_glosnikow*.

Opis rozwiązania: Rysunek przedstawia trzy dostawy telewizorów (czyli *l_dostaw_tv* = 3), w których dostarczono kolejno: 8, 15 i 20 telewizorów, a więc *suma_tv* = 43.



W programie dwa razy obliczamy sumę, a zatem zdefiniowaną funkcję *suma* wywołamy dwukrotnie, ale z innym parametrem aktualnym (raz z parametrem *l_dostaw_tv*, a drugi raz – *l_dostaw_glosnikow*). Wynik działania funkcji (zwracaną wartość) przypiszemy zmiennej.

Wywołanie funkcji *suma* – sposób 1.	
`suma_tv = suma(l_dostaw_tv);` `suma_glosnikow = suma(l_dostaw_glosnikow);`	C++
`suma_tv = suma(l_dostaw_tv)` `suma_glosnikow = suma(l_dostaw_glosnikow)`	Python

Wynik działania funkcji zwracającej wartość możemy również wyprowadzić na ekran bez używania dodatkowych zmiennych (tu: *suma_tv* i *suma_glosnikow*) i przypisywania im wartości funkcji.

Wywołanie funkcji *suma* – sposób 2.	
`cout << "Liczba tv z " << l_dostaw_tv << " dostaw wynosi " << suma(l_dostaw_tv);` `cout << "Liczba glosnikow z " << l_dostaw_glosnikow << " dostaw wynosi " << suma(l_dostaw_glosnikow);`	C++
`print("Liczba tv z", l_dostaw_tv, "dostaw wynosi", suma(l_dostaw_tv))` `print("Liczba głośników z", l_dostaw_glosnikow, "dostaw wynosi", suma(l_dostaw_glosnikow))`	Python

C++

[*] Sumy_dostaw.cpp

```cpp
#include <iostream>
using namespace std;

int suma(int n)
{
    int a, s = 0;

    for(int i = 0; i < n; i++)
    {
        cout << "Podaj liczbe sztuk w " << i+1 << " dostawie: ";
        cin >> a;
        s += a;
    }
    return s;
}

int main()
{
    int l_dostaw_tv, suma_tv;

    cout << "Ile bylo dostaw tv: ";
    cin >> l_dostaw_tv;
    suma_tv = suma(l_dostaw_tv);
    cout << "Liczba tv z " << l_dostaw_tv << " dostaw wynosi " << suma_tv;
    return 0;
}
```

parametr formalny *n*

definicja funkcji *suma*

wartość zwracana

wywołanie funkcji *suma* z parametrem aktualnym *l_dostaw_tv*

część wykonawcza programu

Python

Sumy_dostaw.py - C:\Python\Sumy_dostaw.py (3.7.2)

File Edit Format Run Options Window Help

```python
def suma(n):
    s = 0
    for i in range(n):
        print("Podaj liczbę sztuk w", i + 1, "dostawie:" )
        a = int(input())
        s = s + a
    return s

l_dostaw_tv = int(input("Ile było dostaw tv: "))
suma_tv = suma(l_dostaw_tv)
print("Liczba tv z", l_dostaw_tv, "dostaw wynosi", suma_tv)

input("\n\nAby zakończyć, naciśnij Enter")
```

parametr formalny *n*

definicja funkcji *suma*

wartość zwracana

część wykonawcza programu

wywołanie funkcji *suma* z parametrem aktualnym *l_dostaw_tv*

Ćwiczenie 1. Definiujemy funkcję zwracającą wartość z jednym parametrem i wywołujemy ją w programie głównym

1. Utwórz nowy plik źródłowy i przepisz wybrany program z przykładu 1.
2. Zapisz program w pliku pod nazwą *Sumy_dostaw*.
3. Uruchom i przetestuj program dla kilku różnych wartości zmiennych. Wyjaśnij znaczenie poszczególnych wierszy programu.
4. W programie zapisano tylko obliczenia łącznej wielkości dostaw telewizorów. Dodaj podobne instrukcje dla obliczenia łącznej wielkości dostaw głośników.
5. Zapisz plik pod tą samą nazwą.

Wskazówki:

- W edytorze, w którym tworzysz kod źródłowy programu, można kopiować instrukcje i wklejać je w inne miejsce.
- Ułatwiaj sobie pracę i kopiuj podobne fragmenty programu, w których wystarczy tylko zmienić nazwy zmiennych.

Ćwiczenie 2. Dodajemy do funkcji i programu sprawdzanie poprawności danych

1. Otwórz plik *Sumy_dostaw* zapisany w ćwiczeniu 1.
2. Zmodyfikuj funkcję i program tak, aby obliczenia były wykonywane dla poprawnych (czyli większych od zera) wartości wszystkich zmiennych: *a*, *l_dostaw_tv*, *l_dostaw_glosnikow*. Jeśli zostanie wprowadzona niepoprawna wartość, wyświetl na ekranie komunikat „Wprowadzono bledna liczbe" i zignoruj wprowadzoną wartość.
3. Zapisz program w pliku pod nazwą *Sumy_dostaw_popr*.
4. Uruchom i przetestuj program dla kilku różnych wartości zmiennych.

Ćwiczenie 3. Modyfikujemy program

1. Otwórz plik *Sumy_dostaw_popr* zapisany w ćwiczeniu 2.
2. Zmniejsz liczbę zmiennych, czyli do wyprowadzenia wyników funkcji użyj sposobu 2. z przykładu 1.
3. Do programu dodaj zliczanie dwóch innych artykułów przywożonych do magazynu: smartfonów i komputerów.
4. Zapisz program w pliku pod nazwą *Sumy_dostaw_zmod*.

Funkcje można definiować bez parametrów, ale wówczas, aby przekazać do nich dane, musimy zastosować tzw. **zmienne globalne** (widoczne w dowolnym miejscu programu) – w funkcjach i w programie głównym.

Niezależnie od zmiennych globalnych można w funkcji stosować **zmienne lokalne**. W języku Python zmienne lokalne są widoczne w całej funkcji, a w C++ od miejsca deklaracji do końca funkcji (do nawiasu klamrowego zamykającego definicję funkcji). W funkcji *suma* (przykład 1.) zmienne *i*, *a* oraz *s* są lokalne (przy czym w języku C++ zakres zmiennej *i* jest ograniczony tylko do pętli `for()`).

Przykład 2. Stosowanie funkcji zwracającej wartość bez parametrów w językach C++ i Python

Zadanie: Zdefiniuj funkcję *obwod* bez parametrów, zwracającą do programu głównego obwód kwadratu o boku *bok*. Wywołaj funkcję w programie głównym i wyprowadź wartość obwodu. Wartość zmiennej *bok* wprowadzaj z klawiatury.

Dane: liczba naturalna *bok* oznaczająca długość boku kwadratu.

Wynik: wartość obwodu kwadratu.

Opis rozwiązania: W programie zastosujemy zmienną globalną *bok*.

C++ W języku C++ zmienną *bok* musimy zadeklarować przed definicją funkcji *obwod*.

Python W języku Python instrukcja `bok = int(input("Podaj bok kwadratu: "))` określa typ zmiennej *bok*.

C++

Obwod_kwadratu.cpp

```cpp
#include <iostream>
using namespace std;
int bok;

int obwod()
{
    return 4*bok;
}

int main()
{
    cout << "Podaj bok kwadratu: ";
    cin >> bok;
    cout << "Obwod kwadratu o boku " << bok << " wynosi " << obwod();
    return 0;
}
```

Python

Obwod_kwadratu.py - C:\Python\Obwod_kwadratu.py (3.7.2)

File Edit Format Run Options Window Help

```python
def obwod():
    return 4*bok

bok = int(input("Podaj bok kwadratu: "))
print("Obwod kwadratu o boku", bok, " wynosi", obwod())

input("\n\nAby zakończyć, naciśnij Enter")
```

Ćwiczenie 4. Definiujemy funkcję zwracającą wartość bez parametrów i wywołujemy ją w programie głównym

1. W edytorze kodu źródłowego przepisz wybrany program z przykładu 2. Zapisz program w pliku pod nazwą *Obwod_kwadratu*.
2. Uruchom i przetestuj program dla kilku różnych wartości zmiennej. Wyjaśnij znaczenie poszczególnych wierszy programu.

Ćwiczenie 5. Modyfikujemy program

1. Dodaj do programu zapisanego w ćwiczeniu 4. funkcję *pole* bez parametru obliczającą pole kwadratu o boku *bok*.
2. Zapisz program w pliku pod nazwą *Obwod_i_pole_kwadratu*. Uruchom i przetestuj program dla kilku różnych wartości zmiennej.

Dla zwiększenia przejrzystości i uniwersalności programu należy unikać wykorzystywania zmiennych globalnych, a stosować funkcje z parametrami.

Ćwiczenie 6. Definiujemy funkcję zwracającą wartość z parametrem

Otwórz plik *Obwod_i_pole_kwadratu* zapisany w ćwiczeniu 5. Zmodyfikuj program, rezygnując ze zmiennej globalnej i zmieniając definicje funkcji *suma* i *obwod* na funkcje z parametrem. W programie głównym wywołaj funkcję z parametrem aktualnym *x*, wprowadzanym z klawiatury jako liczba całkowita. Zapisz program w pliku pod nazwą *Obwod_i_pole_kwadratu_z_par*.

Wskazówka: W obydwu funkcjach można użyć tej samej nazwy parametru. W języku C++ w 3. wierszu programu (przykład 2.) zrezygnuj z deklaracji `int bok;`

C++

```
4  int obwod(int bok)
5  {
6      return 4*bok;
7  }
8  int pole(int bok)
9  {
10     return bok*bok;
11 }
```

Python

```
def obwod(bok):
    return 4*bok

def pole(bok):
    return bok*bok
```

Rys. 1. *Definicje funkcji z parametrem – ćwiczenie 6.*

3. Funkcje niezwracające wartości w językach C++ i Python

Funkcje niezwracające wartości stosujemy, gdy mamy wykonać pewne czynności, np. wprowadzić dane, wyprowadzić dane lub obliczyć wiele wartości.

W języku C++ funkcja taka nie ma typu, dlatego w części nagłówkowej przed nazwą funkcji należy umieścić słowo kluczowe **`void`**. Funkcja nie zwraca wartości, dlatego instrukcja **`return`** nie musi wystąpić (a jeśli występuje, to nie może określać wartości).

W języku Python nie określamy typu parametru formalnego funkcji, ponieważ, podobnie jak w przypadku zmiennych, interpreter automatycznie rozpoznaje typ parametru. Funkcja nie zwraca wartości, dlatego instrukcja **`return`** nie musi wystąpić (a jeśli występuje, to nie może określać wartości).

Funkcja niezwracająca wartości	`void nazwa_funkcji(lista_parametrów_formalnych)` `{` `  lista_instrukcji;` `}`	C++
	`def nazwa_funkcji(lista_parametrów_formalnych):` `  lista_instrukcji`	Python

Tabela 3. *Ogólna postać definicji funkcji niezwracającej wartości w językach C++ i Python*

Aby wywołać funkcję niezwracającą wartości, należy umieścić nazwę funkcji (z **parametrami aktualnymi**) w odpowiednim miejscu programu głównego. W przypadku braku parametrów nawiasy po nazwie funkcji pozostawiamy puste.

Wywołanie funkcji niezwracającej wartości	`nazwa_funkcji(lista_parametrów_aktualnych);` lub `nazwa_funkcji();`	C++
	`nazwa_funkcji(lista_parametrów_aktualnych)` lub `nazwa_funkcji()`	Python

Tabela 4. *Ogólna postać wywołania funkcji niezwracającej wartości w językach C++ i Python*

Przykład 3. Stosowanie funkcji niezwracającej wartości bez parametrów w językach C++ i Python

Zadanie: Napisz program umożliwiający wyprowadzenie na ekran monitora napisu „Poznajemy funkcje", a pod nim dwudziestu gwiazdek (znaków *). Zdefiniuj funkcję niezwracającą wartości o nazwie *gwiazdki* bez parametrów. Zadaniem funkcji jest wyprowadzenie w jednym wierszu dwudziestu gwiazdek i przeniesienie kursora do nowego wiersza. Funkcję wywołaj w programie głównym.

C++

Gwiazdki.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  void gwiazdki()
5  {
6      for(int i=0; i<20; i++)
7          cout << "*";
8      cout << endl;
9  }
10
11 int main()
12 {
13     cout << "Poznajemy funkcje" << endl;
14     gwiazdki();
15     return 0;
16 }
```

definicja funkcji

wywołanie funkcji

Python

Gwiazdki.py - C:\Python\Gwiazdki.py (3.7.2)

File Edit Format Run Options Window Help

```
def gwiazdki():
    print("*"*20)

print("Poznajemy funkcje")
gwiazdki()

input("\n\nAby zakończyć, naciśnij Enter")
```

definicja funkcji

wywołanie funkcji

Opis rozwiązania: Celem funkcji jest wyświetlenie z góry określonej liczby gwiazdek. Żadna wartość nie jest przekazywana z programu głównego, więc funkcja nie musi mieć parametrów.

C++ Jedyną zmienną w programie jest zmienna *i*, użyta w funkcji jako zmienna lokalna.
Python W programie nie używamy żadnych zmiennych.

Ćwiczenie 7. Definiujemy funkcję niezwracającą wartości bez parametrów i wywołujemy w programie głównym

1. W edytorze kodu źródłowego przepisz wybrany program z przykładu 3. Zapisz program w pliku pod nazwą *Gwiazdki*.
2. Uruchom i przetestuj program. Wyjaśnij znaczenie poszczególnych wierszy programu.

Ćwiczenie 8. Modyfikujemy program

1. Dodaj do programu zapisanego w ćwiczeniu 7. wyświetlenie w kolejnych wierszach tekstów: „w języku C++" (lub „w języku Python"), „na lekcji informatyki". Pod każdym napisem wyświetl gwiazdki. Ile razy wywołasz funkcję *gwiazdki()*?
2. Zapisz program w pliku pod nazwą *Gwiazdki_z_napisami*.

Przykład 4. Stosowanie funkcji niezwracającej wartości z parametrami w językach C++ i Python

Chcemy, aby liczba wyświetlanych gwiazdek była wprowadzana z klawiatury i przekazywana do funkcji jako parametr.
Do definicji funkcji dodamy parametr *n*, który w momencie wywołania funkcji będzie zastąpiony parametrem aktualnym *l_gwiazdek* (liczbą całkowitą większą od zera wprowadzaną z klawiatury).

C++

```
4  void gwiazdki(int n)
5  {
6      for(int i=0; i<n; i++)
7          cout << "*";
8      cout << endl;
9  }
```

Python

```
def gwiazdki(n):
    print("*"*n)
```

Wywołanie funkcji w programie głównym:

C++ `gwiazdki(l_gwiazdek);`

Python `gwiazdki(l_gwiazdek)`

Ćwiczenie 9. Definiujemy funkcję niezwracającą wartości z parametrami i wywołujemy w programie głównym

1. Otwórz plik *Gwiazdki* zapisany w ćwiczeniu 7. Zmień definicję funkcji *gwiazdki* na funkcję z parametrem *n* i wywołaj w programie głównym z parametrem aktualnym, będącym liczbą całkowitą (*l_gwiazdek*). Wartość parametru wprowadzaj z klawiatury. Zapisz program w pliku pod nazwą *Gwiazdki_par*.
2. Uruchom i przetestuj program dla różnych wartości zmiennej *l_gwiazdek*.

Ćwiczenie 10. Modyfikujemy program

1. Otwórz plik *Gwiazdki_z_napisami* zapisany w ćwiczeniu 8. Zmień definicję funkcji *gwiazdki* na funkcję z parametrem *n* i wywołaj w programie głównym z parametrem aktualnym będącym liczbą całkowitą (*l_gwiazdek*). Wartość parametru wprowadzaj z klawiatury. Zapisz program w pliku pod nazwą *Gwiazdki_z_napisami_par*.
2. Uruchom i przetestuj program dla różnych wartości zmiennej *l_gwiazdek*.

Warto zapamiętać

- Podprogramy (funkcje) stosujemy, aby opracowywać oddzielnie problemy cząstkowe i w ten sposób pisać przejrzyste programy.
- Funkcje w językach C++ i Python mogą zwracać wartości lub nie. Każdy rodzaj funkcji możemy definiować z parametrami lub bez.
- Funkcję wywołujemy w programie głównym, podając jej nazwę i parametry aktualne. W przypadku braku parametrów, po nazwie funkcji umieszczamy puste nawiasy. W przypadku funkcji zwracającej wartość, wartość zwracaną możemy np. przypisać zmiennej w programie głównym.
- Program główny komunikuje się z funkcją przez przekazywanie parametrów – parametry aktualne są wprowadzane w miejsce parametrów formalnych w momencie wywołania funkcji. Gdy wywołanie funkcji się zakończy, zwraca wartość jako swój wynik.

Pytania i polecenia

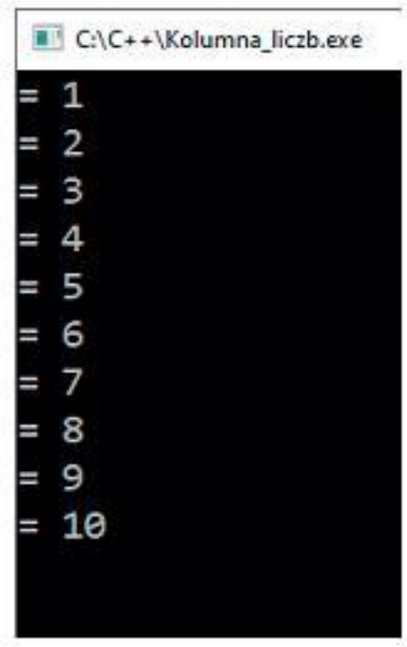
1. Dlaczego definiujemy podprogramy?
2. Czym jest parametr formalny, a czym aktualny?
3. Kiedy definiujemy funkcję niezwracającą wartości, a kiedy zwracającą wartość w języku C++ lub Python?
4. Jak wywołujemy funkcję bez parametrów, a jak z parametrami w języku C++ lub Python?

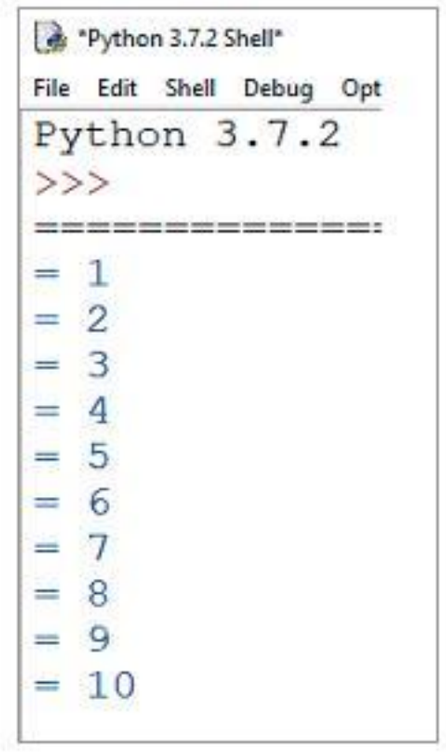
Zadania

1. Do programu *Sumy_dostaw_zmod* zapisanego w ćwiczeniu 3. dodaj jeszcze dwa inne artykuły, które są dostarczane do magazynu i wykonaj te same obliczenia co dla telewizorów i innych artykułów. Zapisz program w pliku *Sumy_dostaw_6*.
2. Napisz program umożliwiający obliczenie objętości graniastosłupa prawidłowego czworokątnego. Zdefiniuj funkcję *objetosc* z dwoma parametrami (*a*, *h*), obliczającą objętość graniastosłupa prawidłowego czworokątnego o krawędzi podstawy *a*

i wysokości *h* i zwracającą do programu głównego wynik obliczenia. Wywołaj funkcję w programie głównym z parametrami aktualnymi *bok* i *wysokosc* wprowadzanymi z klawiatury jako liczby całkowite. Wyprowadź na ekran wynik – objętość graniastosłupa prawidłowego czworokątnego o krawędzi podstawy *bok* i wysokości *wysokosc*. Zapisz program w pliku pod nazwą *Objetosc_graniastoslupa*.

3. Do programu *Objetosc_graniastoslupa* zapisanego w zadaniu 2. dodaj sprawdzanie poprawności wprowadzanych danych. Jeśli użytkownik wprowadzi niepoprawną (ujemną lub równą zero) wartość zmiennej *bok* lub *wysokosc*, wyświetl na ekranie komunikat „Wprowadzono bledna liczbe". Zapisz program w pliku pod nazwą *Objetosc_graniastoslupa_popr*.
4. Zdefiniuj funkcję *kolumna* typu `void` bez parametrów, wyświetlającą na ekranie w kolumnie liczby od 1 do 10 poprzedzone znakiem „=" i spacją – jak pokazano na rysunku 2. Funkcję wywołaj w programie głównym. Zapisz program w pliku pod nazwą *Kolumna_liczb*.
5. Zmodyfikuj program *Kolumna_liczb* zapisany w zadaniu 4., dodając do funkcji parametr *ile* określający liczbę wyświetlanych rzędów liczb. Funkcję wywołaj w programie głównym z parametrem aktualnym *ile_liczb*. Dodaj sprawdzanie poprawności wprowadzanych danych (wartość zmiennej *ile* musi być większa od zera). Zapisz program w pliku pod nazwą *Kolumna_liczb_popr*.





Rys. 2. *Wynik działania programu – zadanie 4.*

Dla zainteresowanych

6. Wymyśl samodzielnie problem (podobnie jak w przykładzie 1.), w którym możesz wykorzystać funkcję *suma*. Sformułuj zadanie i jego specyfikację. W wybranym języku programowania napisz program realizujący to zadanie.
7. Utwórz program zliczający, ile liczb parzystych wprowadzono i wyświetlający wynik na ekranie. Zliczanie wykonuj, dopóki nie zostanie wprowadzone zero. Zdefiniuj funkcję logiczną (typu `bool`) *czy _parzysta* z parametrem całkowitym *a*, sprawdzającą, czy liczba jest parzysta. Wywołaj funkcję w programie głównym. Zapisz program w pliku pod nazwą *Zlicz_parzyste_popr*.
   Wskazówka: Funkcja logiczna *czy _parzysta* zwraca wartość logiczną PRAWDA lub FAŁSZ (w języku C++: `true` / `false`, w języku Python: `True` / `False`). Postać funkcji może być następująca:

C++

```
4  bool czy_parzysta(int a)
5  {
6      return a%2 == 0;
7  }
```

Python

```
def czy_parzysta(a):
    return a%2 == 0
```

Rys. 3. *Definicje funkcji – zadanie 7.*

8. Napisz funkcję *nwd*(*a*, *b*) obliczającą największy wspólny dzielnik dwóch liczb naturalnych *a* i *b*, będących parametrami funkcji. Wykorzystaj funkcję w programie głównym do wyznaczenia największego wspólnego dzielnika dwóch liczb naturalnych.

Temat C3. Stosowanie tablic w języku C++ i list w języku Python

1. Zmienne indeksowane
2. Deklarowanie tablicy w języku C++
3. Definiowanie listy w języku Python
4. Wprowadzanie elementów do tablicy lub listy i wyprowadzanie elementów na ekran
5. Wykonywanie operacji na elementach tablicy lub listy

Warto powtórzyć

1. Do czego służą funkcje w językach programowania?
2. Kiedy stosujemy funkcję zwracającą wartość, a kiedy niezwracającą wartości?
3. W jaki sposób definiujemy funkcje w wybranym języku programowania (C++ lub Python)?
4. Jak wywołujemy funkcję z parametrami, a jak funkcję bez parametrów?

1. Zmienne indeksowane

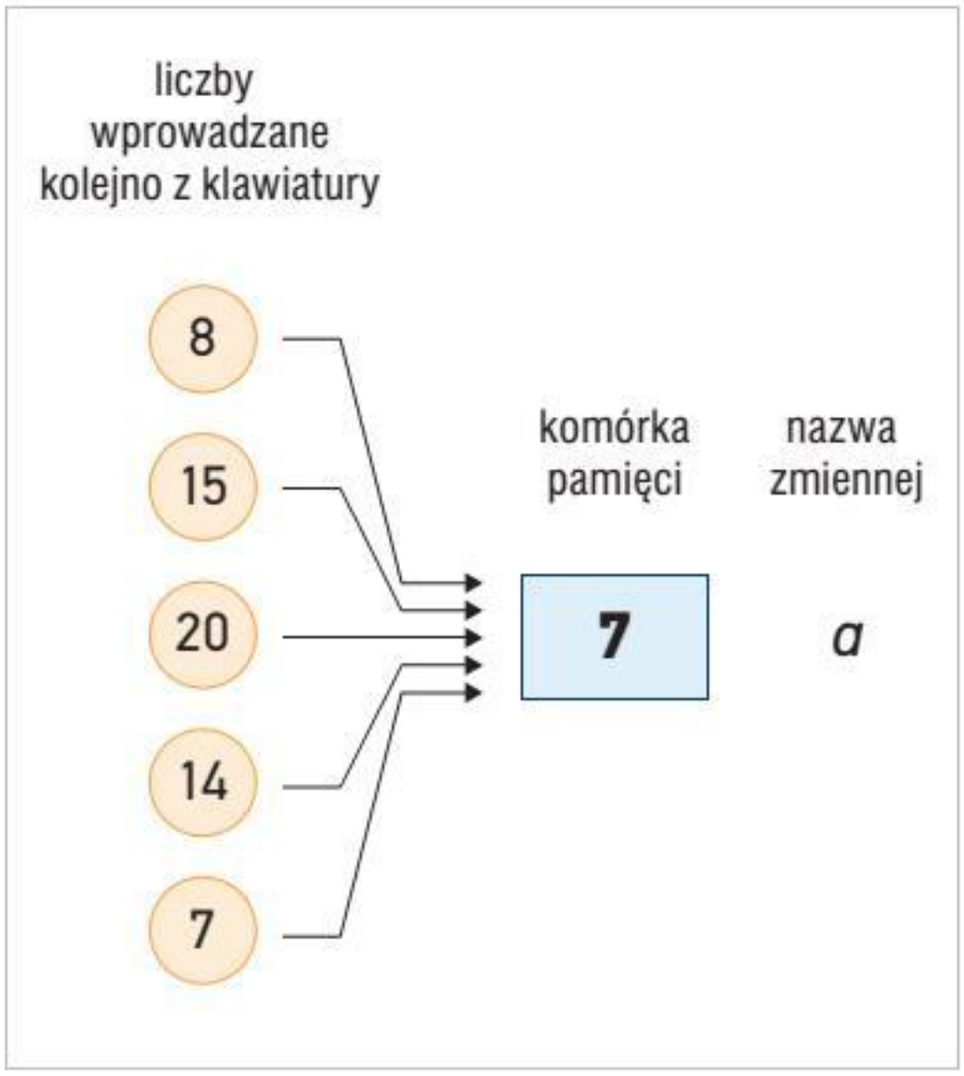
W funkcji *suma* utworzonej w temacie C2 wszystkie wprowadzane liczby komputer pamiętał kolejno w jednej zmiennej (o nazwie *a*), czyli nie zapamiętywał wielkości poszczególnych dostaw. W jaki sposób zapamiętać wszystkie wprowadzone dane?

Gdy z klawiatury wprowadzimy kolejno liczby, na przykład wielkości dostaw telewizorów ze zbioru {8, 15, 20, 14, 7}, każda wprowadzana liczba zastępuje poprzednią (rys. 1a).

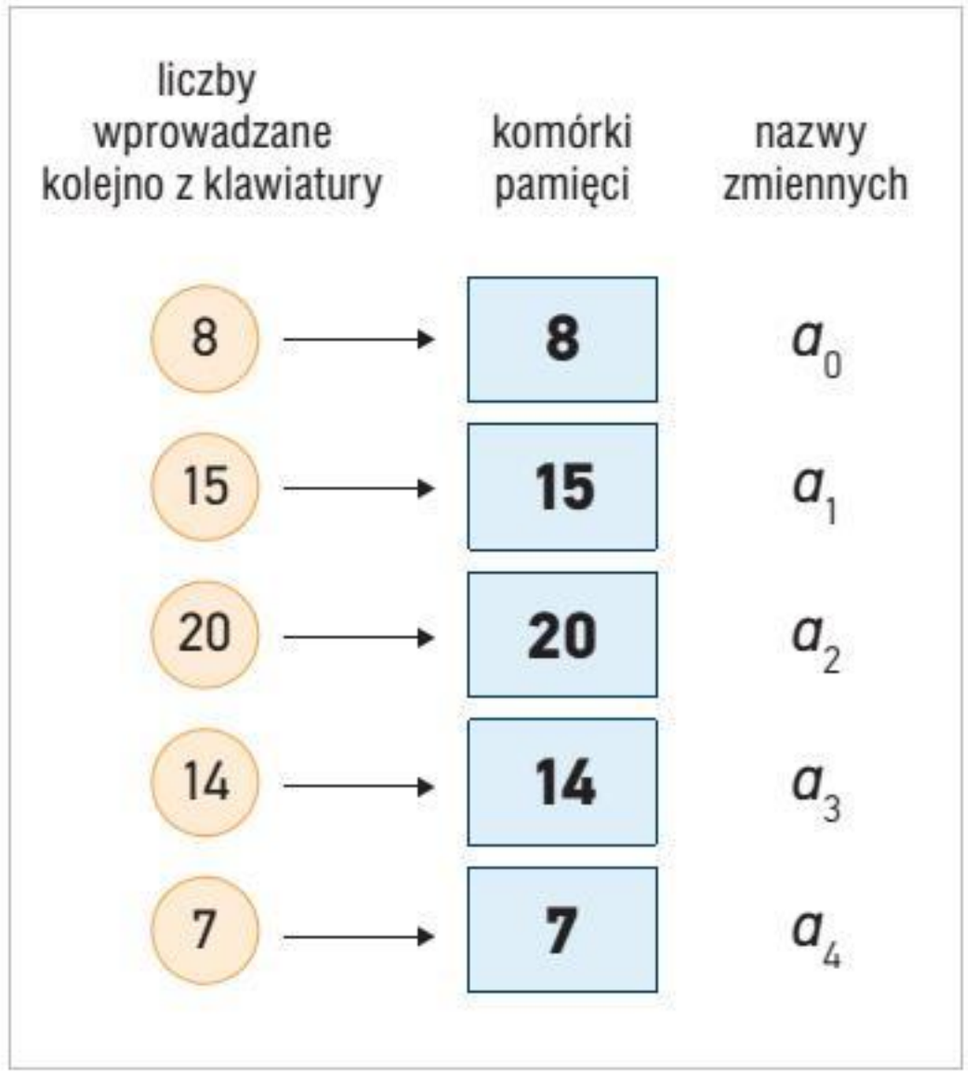
Aby zapamiętać wszystkie dane, musimy je różnie nazwać. Moglibyśmy użyć liter alfabetu, ale w przypadku dużej liczby danych zabrakłoby liter. Ponadto odwoływanie się do wielu zmiennych byłoby niewygodne. Dlatego stosujemy tzw. **zmienne indeksowane**, w których do nazwy zmiennej (np. *a*) dodaje się kolejny numer (**indeks**), np.: a_0, a_1, …, a_{n-1}. Wówczas dla każdej zmiennej komputer rezerwuje oddzielną komórkę pamięci, np. na poszczególne wielkości dostaw telewizorów z pięciu transportów (rys. 1b).

Ćwiczenie 1. Wyjaśniamy, czym są zmienne indeksowane

Objaśnij, czym są zmienne indeksowane, korzystając z przygotowanych pomocy dydaktycznych, np. kartek z przykładowymi liczbami.



Rys. 1a. *Zapisywanie w pamięci komputera wartości danych wprowadzanych z klawiatury z użyciem jednej zmiennej*



Rys. 1b. *Zapisywanie w pamięci komputera wartości danych wprowadzanych z klawiatury z użyciem pięciu zmiennych indeksowanych*

2. Deklarowanie tablicy w języku C++

Aby wykorzystać zmienne indeksowane w języku C++, musimy utworzyć specjalną strukturę danych – **tablicę**.

Deklaracja tablicy	`opis_typu nazwa_tablicy [liczba_elementów_tablicy];`	C++

Tabela 1. *Ogólna postać deklarowania tablicy w języku C++*

Do elementów tablicy odwołujemy się, podając nazwę tablicy i indeks elementu umieszczony w nawiasach kwadratowych, np.: $a[0]$, $a[1]$, …, $a[n - 1]$ – dla tablicy n-elementowej o nazwie a. W języku C++ pierwszy indeks jest zawsze równy 0.

Liczbę elementów tablicy musimy określić wcześniej:

- wpisać w deklaracji tablicy jako konkretną wartość (liczbę naturalną większą od 0)
  lub
- podać za pomocą wcześniej zdefiniowanej stałej (`const`).

Nie można zmienić liczby elementów raz zadeklarowanej tablicy.

! Uwaga

Stałe w językach C++ i Python będziemy zapisywać wielkimi literami. W obydwu językach wielkie i małe litery są rozróżniane (np. *n* i *N* oznaczają dwie różne zmienne lub stałe).

Deklaracja tablicy	Opis	Sposób odwołania
`int sztuki[100];`	oznacza zadeklarowanie tablicy o nazwie *sztuki* składającej się ze stu liczb całkowitych o indeksach od 0 do 99	do elementów tablicy odwołujemy się przez zmienne: *sztuki*[0], *sztuki*[1], …, *sztuki*[99]
`const int N = 20;` `int a[N];`	oznacza zadeklarowanie tablicy o nazwie *a* składającej się z dwudziestu elementów typu całkowitego (**int**).	do elementów tablicy odwołujemy się przez zmienne: *a*[0], a[1], …, *a*[*N*-1]
`const int LICZBA_EL = 80;` `float tablica[LICZBA_EL];`	oznacza utworzenie tablicy o nazwie *tablica* składającej się z osiemdziesięciu elementów typu rzeczywistego (**float**)	do elementów tablicy odwołujemy się przez zmienne: *tablica*[0], *tablica*[1], …, *tablica*[*LICZBA_EL*-1]

Tabela 2. *Przykłady deklarowania tablicy w języku C++*

Początkowe wartości elementów tablicy można podać w momencie jej definiowania, zapisując je w nawiasach klamrowych.

Deklaracja tablicy	Opis
`int DOSTAWY[5] = {8,15,20,14,7};`	tablica *DOSTAWY* składająca się z pięciu liczb całkowitych zostanie wypełniona liczbami ze zbioru {8, 15, 20, 14, 7}
`int DOSTAWY[] = {8,15,20,14,7};`	jak wyżej, ale liczba elementów tablicy *DOSTAWY* (pięć elementów) zostanie ustalona na podstawie liczby danych początkowych
`int sztuki[100] = {0};`	tablica *sztuki*, składająca się ze stu liczb całkowitych, zostanie wypełniona liczbami 0

Tabela 3. *Przykłady ustalania początkowych wartości tablicy (inicjalizowania tablicy) w języku C++*

3. Definiowanie listy w języku Python



Aby wykorzystać zmienne indeksowane w języku Python, możemy zdefiniować specjalną strukturę danych – **listę**.

Definicja listy	`nazwa_listy = [element1, element2, …, element_n]`	**Python**

Tabela 4. *Ogólna postać definiowania listy w języku Python*

Do elementów listy odwołujemy się, podając nazwę listy i indeks elementu umieszczony w nawiasach kwadratowych, np. *a*[0], *a*[1], …, *a*[*n* - 1] – dla listy *n*-elementowej o nazwie *a*. W języku Python pierwszy indeks jest zawsze równy 0.

Lista (w odróżnieniu od tablicy) może zawierać elementy różnego typu, dlatego w języku Python nie określamy typu elementów listy.

Zarówno typ zmiennych przechowywanych w liście, jak i liczba elementów listy mogą się zmienić w trakcie działania programu.

Wartości elementów listy możemy:

- podać od razu w definicji listy,
- przypisać od razu taką samą wartość wszystkim elementom listy.

Liczbę elementów listy możemy przypisać zmiennej.

> **! Uwaga**
>
> Liczbę przykładów definiowania list w języku Python ograniczyliśmy do kilku przydatnych do dalszych ćwiczeń.

Definicja listy	Opis	Sposób odwołania
`DOSTAWY = [8,15,20,14,7]`	oznacza zdefiniowanie listy o nazwie *DOSTAWY*, składającej się z pięciu elementów o indeksach od 0 do 4	do elementów listy odwołujemy się przez zmienne: *DOSTAWY*[0], *DOSTAWY*[1], …, *DOSTAWY*[4]
`lista_elementow = [0] * 20`	oznacza zdefiniowanie listy o nazwie *lista_elementow* składającej się z dwudziestu elementów, z których każdy jest równy zero (wyzerowanie wszystkich elementów listy)	do elementów listy odwołujemy się przez zmienne: *lista_elementow* [0], *lista_elementow*[1], …, *lista_elementow*[19]
`N = 50` `a = [0] * N`	oznacza zdefiniowanie listy o nazwie *a* składającej się z pięćdziesięciu elementów o wartości początkowej zero	do elementów listy odwołujemy się przez zmienne: *a*[0], *a*[1], …, *a*[N-1]

Tabela 5. *Przykłady definiowania listy w języku Python*

4. Wprowadzanie elementów do tablicy lub listy i wyprowadzanie elementów na ekran

Do wprowadzania danych do tablicy (C++) i listy (Python) oraz do wyprowadzania elementów możemy zastosować funkcję niezwracającą wartości bez parametrów lub z parametrami.

Przykład 1. Deklarowanie i stosowanie tablicy w języku C++

Zadanie: Wprowadź *N* liczb całkowitych do tablicy o nazwie *a*, następnie wyprowadź w kolumnie elementy tablicy na ekran. Zdefiniuj dwie funkcje niezwracające wartości bez parametrów *wprowadz_dane()* i *wyprowadz_dane()*, które wywołaj w programie głównym.

Dane: stała wartość *N*, tablica liczb *a*[*N*].

Wyniki: wyświetlone na ekranie w kolumnie elementy tablicy *a*: *a*[0], a[1], …, *a*[*N*-1].

C++ [*] Elementy_tablicy.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  const int N = 5;
5  int a[N];
6
7  void wprowadz_dane()
8  {
9      for(int i = 0; i < N; i++)
10     {
11         cout << "Podaj dana nr " << i << ": ";
12         cin >> a[i];
13     }
14 }
15
16 void wyprowadz_dane()
17 {
18     for(int i = 0; i < N; i++)
19         cout << "a[" << i << "] = " << a[i] << endl;
20 }
21
22 main ()
23 {
24     wprowadz_dane();
25     wyprowadz_dane();
26     return 0;
27 }
```

deklaracja stałej *N* o wartości 5

deklaracja tablicy o nazwie *a*

definicja funkcji *wprowadz_dane*

definicja funkcji *wyprowadz_dane*

wywołanie funkcji *wprowadz_dane*

wywołanie funkcji *wyprowadz_dane*

Przykład 2. Definiowanie i stosowanie listy w języku Python

Zadanie: Wprowadź *N* liczb całkowitych do listy o nazwie *a*, następnie wyprowadź w kolumnie elementy listy na ekran. Zdefiniuj dwie funkcje niezwracające wartości bez parametrów *wprowadz_dane()* i *wyprowadz_dane()*, które wywołaj w programie głównym.

Dane: stała wartość *N*, lista liczb *a*[*N*].

Wyniki: wyświetlone w kolumnie elementy listy *a*: *a*[0], *a*[1], …, *a*[*N*-1].

Uwagi:

- Funkcja *wprowadz_dane* zmodyfikowała wartości istniejących zmiennych (które na początku były równe zero).
- Zapisany w języku Python program można również uruchomić, gdy zamkniemy okno z programem. Należy w tym celu z menu kontekstowego ikony programu (widocznej w Eksploratorze plików) wybrać polecenie:
  - **Otwórz** – w otwartym oknie zobaczymy uruchomiony program,
  - **Edit with IDLE** – program otworzy się w oknie powłoki Pythona (Python Shell) i będzie można go uruchomić.

Python

Elementy_listy.py - C:\Python\Elementy_listy.py (3.7.2)

File Edit Format Run Options Window Help

```
N = 5
a = [0] * N

def wprowadz_dane():
    for i in range(N):
        a[i] = int(input("Podaj liczbe: "))

def wyprowadz_dane():
    for i in range(N):
        print("a[", i, "] = ", a[i])

wprowadz_dane()
wyprowadz_dane()

input("\n\nAby zakończyć, naciśnij Enter")
```

określenie liczby elementów listy

zdefiniowanie listy o nazwie *a* i wyzerowanie jej elementów

definicja funkcji *wprowadz_dane*

definicja funkcji *wyprowadz_dane*

wywołanie funkcji *wprowadz_dane*

wywołanie funkcji *wyprowadz_dane*

Ćwiczenie 2. Wprowadzamy dane do tablicy lub listy i wyprowadzamy je na ekran

1. Utwórz nowy plik źródłowy i przepisz program z przykładu 1. lub z przykładu 2. Zapisz program w pliku pod nazwą *Elementy_tablicy* lub *Elementy_listy*.
2. Uruchom i przetestuj program dla kilku różnych wartości elementów tablicy *a* lub listy *a*. Wyjaśnij znaczenie poszczególnych wierszy programu.

! Uwaga

W języku Python wszystkie zmienne, których nazwa jest zapisana wielkimi literami (i ewentualnie z podkreślnikami) będziemy traktować w programie jak stałe.

5. Wykonywanie operacji na elementach tablicy lub listy

Jeśli wszystkie wprowadzone dane pamiętamy w tablicy lub liście, możemy już po ich wprowadzeniu wykonywać na nich różne operacje, np. wyświetlić w innej kolejności, zsumować wybrane elementy czy uporządkować je (zastosowanie tablic i list w algorytmach porządkowania poznamy w temacie C4).

Ćwiczenie 3. Wyprowadzamy elementy z tablicy lub listy w odwrotnej kolejności

1. Otwórz program *Elementy_tablicy* lub *Elementy_listy* zapisany w ćwiczeniu 2. Zmodyfikuj program tak, aby dane wyświetlały się w odwrotnym porządku. Dodatkowo zwiększ liczbę danych do dziesięciu.
2. Zapisz program w pliku pod nazwą *Odwrotna_kolejnosc*.

Wskazówki:

C++ W instrukcji iteracyjnej **`for`** użyj odpowiedniej wartości początkowej, warunku zakończenia pętli oraz operatora dekrementacji `i--` zamiast operatora inkrementacji `i++`.

Python W instrukcji iteracyjnej **`for`** użyj w funkcji `range` trzech argumentów: `range(początek, koniec, krok)`.

Ćwiczenie 4. Wyświetlamy na ekranie wybrany element tablicy lub listy

1. Otwórz program *Elementy_tablicy* lub *Elementy_listy* zapisany w ćwiczeniu 2. Zmodyfikuj program tak, aby na ekranie wyświetlać tylko element *k*-ty, gdzie *k* jest liczbą całkowitą wprowadzaną z klawiatury po uruchomieniu programu.
2. Zapisz program w pliku pod nazwą *Wybrany_element*.

Wskazówka: Należy sprawdzić, czy wprowadzona wartość *k* jest większa lub równa 0 oraz mniejsza od *N*.

Ćwiczenie 5. Porównujemy dwa elementy tablicy lub listy i wyświetlamy na ekranie większy z nich

1. Korzystając z funkcji *wprowadz_dane()* z programów *Elementy_tablicy* lub *Elementy_listy* (ćwiczenie 2.) oraz programu *Sumy_dostaw* (ćwiczenie 1., temat C2), napisz program sprawdzający dla dziesięciu dostaw, która dostawa telewizorów jest większa: pierwsza czy ostatnia. Zależnie od wyniku, program powinien wyprowadzać na ekran komunikat: „Wieksza jest pierwsza” lub „Wieksza jest ostatnia”. W przypadku dostaw o tej samej liczbie telewizorów wyprowadź komunikat „Dostawy rowne”.
2. Zapisz program w pliku pod nazwą *Wieksza*.

Warto zapamiętać

- Aby zaprogramować algorytm, w którym w trakcie działania programu musimy pamiętać wszystkie elementy zbioru, możemy zapisywać dane w tablicy (C++) lub na liście (Python).
- W języku C++ liczba elementów (rozmiar) tablicy musi być z góry określona, np. jako zdefiniowana wcześniej stała. Musimy również określić typ elementów.
- W języku Python nie określamy typu elementów listy. Interpreter określa typ elementów na podstawie rodzaju danych, jaki im przypiszemy.
- Do elementów tablicy (C++) i listy (Python), np. o nazwie *lista* i liczbie elementów *N*, odwołujemy się poprzez podanie nazwy i indeksu umieszczonego w nawiasach kwadratowych *lista*[0], *lista*[1], …, *lista*[*N* - 1].

Pytania i polecenia

1. Czym jest zmienna indeksowana?
2. W jakim celu definiujemy tablicę? Podaj przykład użycia tablicy.
3. Wyjaśnij na przykładach sposób deklarowania tablicy w języku C++.
4. W jakim celu definiujemy listę? Podaj przykład użycia listy.
5. Wyjaśnij na przykładach sposób definiowania listy w języku Python.

Zadania

1. Otwórz program *Elementy_tablicy* lub *Elementy_listy* zapisany w ćwiczeniu 2. Zmodyfikuj program tak, aby na ekranie wyświetlić tylko pierwszy i ostatni element. Zapisz program w pliku pod nazwą *Element_pierwszy_i_ostatni*.

2. Otwórz program *Element_pierwszy_i_ostatni* zapisany w zadaniu 1. Zmodyfikuj program tak, aby sprawdzać, który element jest większy: pierwszy czy drugi. Wyświetlaj tylko element większy oraz komunikat, który to element. W przypadku elementów równych wyświetlaj komunikat „Elementy rowne". Zapisz program w pliku pod nazwą *Element_wiekszy*.
3. Otwórz program *Elementy_tablicy* lub *Elementy_listy* zapisany w ćwiczeniu 2. Dodaj do programu definicję funkcji (typu całkowitego w języku C++) *suma_danych*(), obliczającej sumę elementów tablicy lub listy i zwracającą tę wartość do programu głównego. Zapisz program w pliku pod nazwą *Suma_elementow*.
   Wskazówka: W treści funkcji umieść instrukcję: `suma += a[i]` odpowiadającą instrukcji `suma = suma + a[i]`.
4. Napisz program, który wczytuje liczby całkowite do ośmioelementowej tablicy lub listy i wypisuje na ekran indeks pierwszego elementu tablicy lub listy, którego wartość wynosi zero. Jeśli nie ma takiego elementu, program powinien wypisać komunikat „Brak elementu zero". Zapisz program w pliku pod nazwą *Indeks_elementu_zero*.
5. Zadeklaruj tablicę (C++) lub zdefiniuj listę (Python) składającą się z pięciu elementów, będących liczbami godzin lekcyjnych od poniedziałku do piątku. Utwórz program wypisujący w kolumnie elementy tablicy lub listy, czyli liczby lekcji każdego dnia. Zapisz program w pliku pod nazwą *Lekcje_w_tygodniu*.
   Wskazówka: Zastosuj sposób definiowania tablicy pokazany w pierwszym wierszu tabeli 3. lub listy pokazany w pierwszym wierszu tabeli 5.
6. Korzystając z tablicy lub z listy zdefiniowanej w zadaniu 5., utwórz program, który dla podanego z klawiatury numeru dnia tygodnia wyświetli liczbę godzin lekcji tego dnia. Zapisz program w pliku pod nazwą *Lekcje_w_dniu*.
7. Zmodyfikuj program *Sumy_dostaw* zapisany w ćwiczeniu 1. z tematu C2 tak, aby liczby dostaw telewizorów były pamiętane jako elementy tablicy lub listy *tv*[*T*], a liczby dostaw głośników jako elementy tablicy lub listy *glosniki*[*G*], gdzie *T* i *G* są stałymi równymi 5. Wyprowadzaj na ekran elementy obydwu tablic, obliczaj sumy dostaw telewizorów i głośników i wyprowadź je na ekran. Zapisz program w pliku pod nazwą *Sumy_dostaw_tablice* lub *Sumy_dostaw_listy*.

Dla zainteresowanych

8. Otwórz program *Suma_elementow* zapisany w zadaniu 3. Zwiększ wielkość tablicy (listy) do 10 ($N = 10$). Zmodyfikuj funkcję *suma*(), dodając parametr *ile* określający, ile początkowych elementów tablicy lub listy ma być zsumowanych. W programie głównym wywołaj funkcję z parametrem aktualnym *liczba_el* wprowadzanym z klawiatury. Sprawdź, czy jego wartość jest poprawna (czyli większa lub równa 0 i mniejsza lub równa *N*). Jeżeli nie jest, wyświetl komunikat „Wprowadzana wartosc danych jest spoza zakresu". Zapisz program w pliku *Suma_elementow_ile_z8*.
9. Otwórz program *Suma_elementow_ile_z8* zapisany w zadaniu 8. Zwiększ wielkość tablicy (listy) do 100 ($N = 100$). Zmodyfikuj funkcje wprowadzania i wyprowadzania danych, dodając do każdej parametr *ile* określający, ile elementów tablicy lub listy ma być wprowadzonych i wyprowadzonych oraz zsumowanych. Zsumowane mają być wszystkie wprowadzone elementy. Wszystkie operacje wykonuj dla poprawnych danych. Zapisz program w pliku *Suma_elementow_ile_z9*.
   Wskazówka: W programie głównym sprawdzaj, czy wartość zmiennej *liczba_el* jest mniejsza lub równa *N*.
10. Zmodyfikuj program *Suma_elementow_ile_z9* zapisany w zadaniu 9. tak, aby zsumować oddzielnie elementy o indeksach parzystych i nieparzystych. Element o indeksie 0 dodaj do sumy elementów parzystych. Zapisz program w pliku *Suma_parzyste_i_nieprzyste*.

Temat C4. Wybrane algorytmy na liczbach naturalnych i algorytmy porządkowania

1. Algorytm badania pierwszości liczby
2. Obliczanie wartości elementów ciągu Fibonacciego
3. Algorytmy porządkowania
   - 3.1. Porządkowanie metodą bąbelkową
   - 3.2. Porządkowanie przez wstawianie

Warto powtórzyć

1. Jakie instrukcje umożliwiają zapisywanie w wybranym języku programowania (C++ lub Python) algorytmów iteracyjnych?
2. W jaki sposób zapisujemy w wybranym języku programowania (C++ lub Python) algorytmy z warunkami?
3. Jakie rodzaje funkcji można definiować w językach C++ lub Python?
4. Czym jest parametr formalny funkcji, a czym aktualny?
5. W jaki sposób deklarujemy tablicę w języku C++?
6. W jaki sposób definiujemy listę w języku Python?

1. Algorytm badania pierwszości liczby

Algorytm badania pierwszości liczb polega na sprawdzaniu dla każdej liczby wszystkich jej dzielników większych od 1 i mniejszych od niej samej. Jeżeli liczba nie posiada takich dzielników, jest **liczbą pierwszą**.

Liczba pierwsza L

Liczba naturalna większa od 1, która ma dokładnie dwa dzielniki naturalne: jedynkę i siebie samą, np.: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Aby określić, czy dana liczba n jest pierwsza, należy sprawdzić jej dzielniki. Nie musimy jednak sprawdzać wszystkich dzielników od 2 do n – wystarczy sprawdzić, czy dana liczba ma dzielniki w zbiorze $\langle 2; \sqrt{n} \rangle$. Jeśli ma, nie jest liczbą pierwszą.

Na przykład:

- liczba 9 w zbiorze $\langle 2; \sqrt{9} \rangle$ ma jeden dzielnik: 3 – czyli nie jest liczbą pierwszą,
- liczba 17 w zbiorze $\langle 2; \sqrt{17} \rangle$ nie ma dzielników – czyli jest liczbą pierwszą.

Ćwiczenie 1. Sprawdzamy, czy liczba jest liczbą pierwszą

Sprawdź, która z podanych liczb: 18, 31, 42, 89 jest liczbą pierwszą.

Uwaga

Dowód, dlaczego wystarczy sprawdzać dzielniki do $\sqrt{n}$ pozostawiamy osobom zainteresowanym.

Przykład 1. Zapisywanie algorytmu badania pierwszości liczby w językach programowania C++ i Python

Zadanie: Napisz program w wybranym języku programowania, sprawdzający, czy liczba naturalna *n* (większa od 1) jest liczbą pierwszą.

Dane: liczba naturalna *n* większa od 1.

Wynik: komunikat „jest pierwsza" lub „nie jest pierwsza".

Opis rozwiązania:

W programie zdefiniujemy funkcję *jest_pierwsza* z jednym parametrem formalnym *n* sprawdzającą, czy liczba *n* jest liczbą pierwszą. Dla liczb pierwszych funkcja zwraca wartość logiczną PRAWDA, w przeciwnym przypadku – wartość logiczną FAŁSZ.

W implementacji funkcji *jest_pierwsza* zastosowano funkcje standardowe:

`floor()` – zaokrągla liczbę z częścią ułamkową w dół (obcina część ułamkową), np. wywołanie `floor(18.7777)` zwróci 18,

`sqrt()` – oblicza pierwiastek drugiego stopnia, np. `sqrt(9)` zwróci liczbę 3.

Zastosowano również operator % oznaczający resztę z dzielenia i operator `(int)` powodujący zamianę wartości rzeczywistej na całkowitą.

Funkcję *jest_pierwsza* wywołamy w programie głównym. Wywołanie funkcji jest umieszczone jako warunek w instrukcji **if**, ponieważ funkcja zwraca wartość logiczną.

C++

Pierwsza.cpp

```
1  // Sprawdzanie, czy liczba jest pierwsza
2  #include <iostream>
3  #include <cmath>
4
5  using namespace std;
6
7  bool jest_pierwsza(int n)
8  {
9      bool rezultat = true;
10     int m = (int) floor(sqrt(n));
11     for(int i = 2; i <= m; i++)
12         if(n % i == 0)
13             return false;
14     return true;
15 }
16
17 int main()
18 {
19     int liczba;
20     cout << "Podaj liczbe: ";
21     cin >> liczba;
22
23     cout << "Liczba " << liczba;
24     if( jest_pierwsza(liczba) )
25         cout <<" jest pierwsza" << endl;
26     else
27         cout << " nie jest pierwsza" << endl;
28     return 0;
29 }
```

Dołączamy bibliotekę funkcji matematycznych.

Zmiennej całkowitej *m* przypisujemy część całkowitą pierwiastka drugiego stopnia z *n*.

Sprawdzamy, czy *i* dzieli *n*.

Wartość parametru aktualnego *liczba* wprowadzamy z klawiatury.

Wywołujemy funkcję *jest_pierwsza* z parametrem aktualnym *liczba*.

Python

Pierwsza.py - C:\Python\Pierwsza.py (3.7.2)

File Edit Format Run Options Window Help

```
# Sprawdzanie, czy liczba jest pierwsza

from math import floor, sqrt

def jest_pierwsza(n):
    m = floor(sqrt(n))
    for i in range (2, m+1):
        if n%i==0:
            return False
    return True

liczba = int(input("Podaj liczbę: "))
if jest_pierwsza(liczba):
    print("Liczba", liczba, "jest pierwsza")
else:
    print("Liczba", liczba, "nie jest pierwsza")

input("\n\nAby zakończyć, naciśnij Enter")
```

Dołączamy bibliotekę funkcji matematycznych.

Zmiennej całkowitej *m* przypisujemy część całkowitą pierwiastka drugiego stopnia z *n*.

Sprawdzamy, czy *i* dzieli *n*.

Wartość parametru aktualnego *liczba* wprowadzamy z klawiatury.

Wywołujemy funkcję *jest_pierwsza* z parametrem aktualnym *liczba*.

Ćwiczenie 2. Zapisujemy algorytm badania pierwszości liczb w języku programowania

1. Utwórz nowy plik źródłowy i przepisz wybrany program z przykładu 1. Zapisz program w pliku pod nazwą *Pierwsza*.
2. Uruchom i przetestuj program dla kilku różnych wartości parametru aktualnego *liczba*. Wyjaśnij znaczenie poszczególnych wierszy programu.

2. Obliczanie wartości elementów ciągu Fibonacciego

W roku 1202 wybitny włoski matematyk okresu średniowiecza, Leonardo Pisano, zwany Fibonaccim, przedstawił problem, który do tej pory dostarcza natchnienia rzeszom matematyków. Dotyczy on liczebności populacji królików.

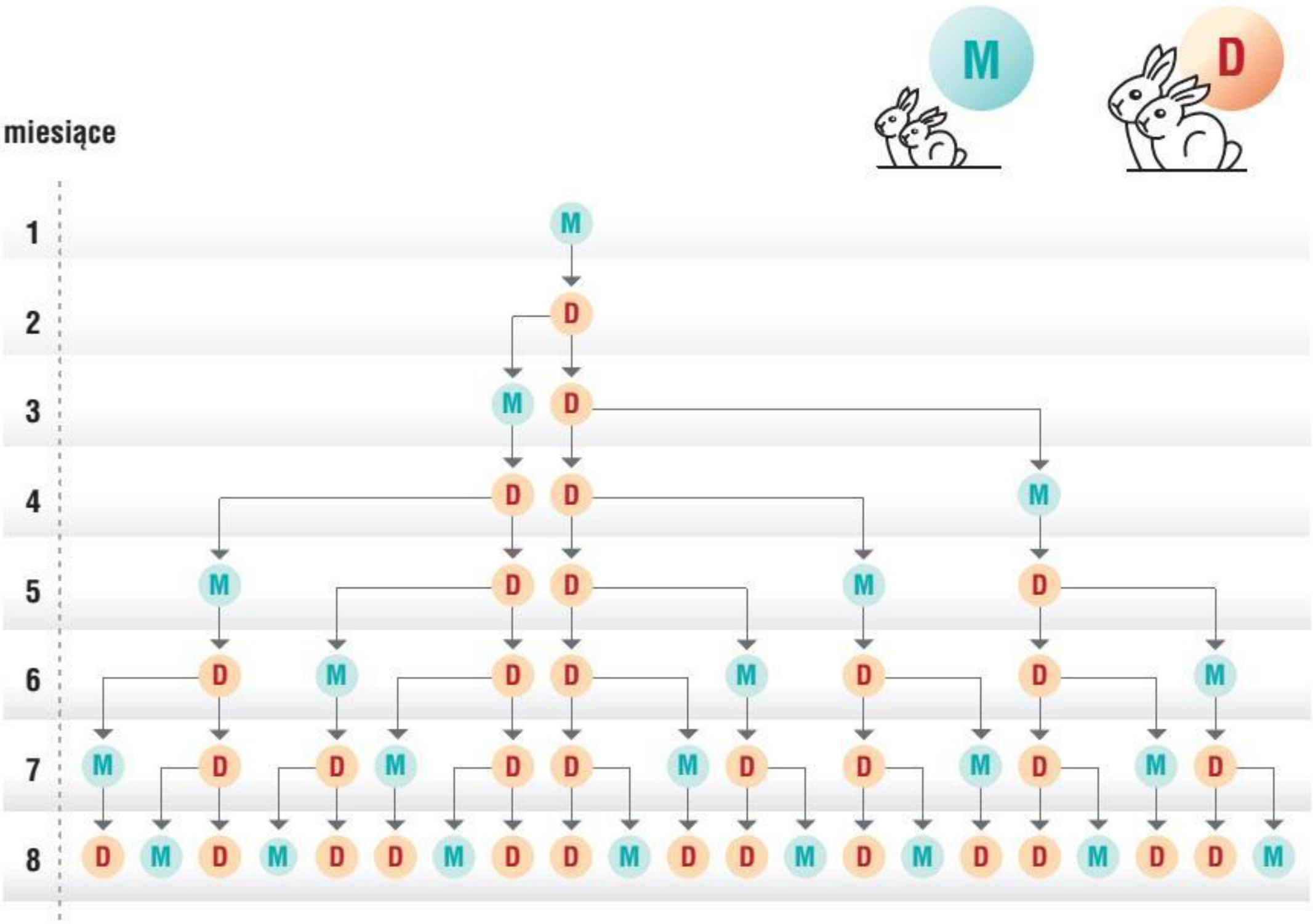
Opis problemu: Początkowo rodzi się para królików, która po miesiącu osiąga dojrzałość. Po następnym miesiącu wydaje na świat parę królików, po miesiącu następną parę i tak dalej, tzn. pierwsza para co miesiąc wydaje na świat potomstwo (parę królików). Z kolei każda nowa para po miesiącu osiąga dojrzałość i po każdym następnym miesiącu również wydaje na świat kolejną parę królików. Proces ten nie ma końca. Zakładamy, że króliki żyją wiecznie.

Pytanie brzmi: jaka jest liczba par królików po *n* miesiącach?

Opis algorytmu: Zauważmy, że liczba par królików w danym miesiącu (z wyjątkiem pierwszego i drugiego miesiąca) zależy od dwóch wartości: liczby par królików dorosłych i liczby par królików młodych w poprzednim miesiącu. Tych ostatnich jest z kolei tyle, ile było par królików dorosłych dwa miesiące wcześniej.

Jeśli oznaczymy całkowitą liczbę par królików w *n*-tym miesiącu jako F_n, to będziemy ją mogli określić następująco:

$$F_n = \begin{cases} 1 & \text{dla } n \in \{1, 2\} \\ F_{n-1} + F_{n-2} & \text{dla } n \geq 3. \end{cases}$$



Rys. 1. *Schemat przedstawia liczebność populacji królików w pierwszych ośmiu miesiącach doświadczenia (M – para królików młodych, D – para królików dojrzałych); ósma liczba Fibonacciego to 21 par*

Ćwiczenie 3. Obliczamy wybrane liczby Fibonacciego

Oblicz (w zeszycie przedmiotowym lub na kartce) trzecią i czwartą liczbę ciągu Fibonacciego. Czy możesz obliczyć piątą liczbę bez wcześniejszego obliczenia trzeciej i czwartej? Uzasadnij odpowiedź.

Liczba F_n opisuje całkowitą wielkość populacji królików po upływie n miesięcy i nazywana jest **liczbą Fibonacciego**. Natomiast ciąg liczb naturalnych, w którym każda liczba (z wyjątkiem pierwszej i drugiej) jest sumą dwóch poprzednich, nazywamy **ciągiem Fibonacciego**.

Oto kilka początkowych liczb tego ciągu: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 …

By obliczyć poszczególne liczby Fibonacciego, można posłużyć się algorytmem iteracyjnym. W postaci iteracyjnej obliczenia zaczynamy „od początku”, przyjmując, że wartość dwóch pierwszych liczb Fibonacciego wynosi 1.

Przykład 2. Zapisywanie algorytmu obliczania wartości elementu ciągu Fibonacciego w językach programowania C++ i Python

Zadanie: Napisz w wybranym języku programowania program obliczający i wyświetlający na ekranie *k* początkowych liczb Fibonacciego.

Dane: liczba naturalna *k* większa od zera oznaczająca liczbę wyświetlonych liczb Fibonacciego.

Wynik: *k* kolejnych liczb Fibonacciego.

Opis rozwiązania: W programie zdefiniujemy funkcję iteracyjnego sposobu obliczania liczb Fibonacciego *fibonacci*(*n*) – z jednym parametrem formalnym *n*. Funkcja będzie zwracać do programu głównego wartość *n*-tej liczby Fibonacciego.
Funkcję wywołamy *k* razy w programie głównym z parametrem aktualnym *i* oznaczającym numer kolejnej liczby Fibonacciego (*i* przyjmuje wartości od 1 do *k*).

C++

[*] Liczby_Fibonacciego.cpp

```
1  //Iteracyjna realizacja algorytmu obliczania liczb Fibonacciego
2
3  #include <iostream>
4  using namespace std;
5
6  int fibonacci(int n)
7  {
8      int wynik;
9
10     if( n < 3 )
11         wynik = 1;
12     else
13     {
14         int poprzedni1 = 1;
15         int poprzedni2 = 1;
16         for(int i = 3; i <= n; ++i)
17         {
18             wynik = poprzedni1 + poprzedni2;
19             poprzedni2 = poprzedni1;
20             poprzedni1 = wynik;
21         }
22     }
23     return wynik;
24 }
25
26 int main()
27 {
28     int k;
29     cout << "Ile liczb Fibonacciego wyswietlic: ";
30     cin >> k;
31     for(int i = 1; i <= k; ++i)
32         cout << "F(" << i << ")=" << fibonacci(i) << endl;
33
34     return 0;
35 }
```

Dwie pierwsze liczby są równe jeden.

Zmiennej *wynik* przypisujemy sumę dwóch poprzednich liczb, zmiennej *poprzedni2* przypisujemy wartość zmiennej *poprzedni1*, a zmiennej *poprzedni1* – wartość zmiennej *wynik*.

wywołanie funkcji *fibonacci(i)*

Python

Liczby_Fibonacciego.py - C:\Python\Liczby_Fibonacciego.py (3.7.2)

File Edit Format Run Options Window Help

```
# Iteracyjna realizacja algorytmu obliczania liczb Fibonacciego

def fibonacci(n):
    if n < 3:
        wynik = 1
    else:
        poprzedni1 = 1
        poprzedni2 = 1
        for i in range(3, n+1):
            wynik = poprzedni1 + poprzedni2
            poprzedni2 = poprzedni1
            poprzedni1 = wynik
    return wynik

k = int(input("Ile liczb Fibonacciego wyświetlić: "))
for i in range(1, k+1):
    print("F(", i, ")=", fibonacci(i))

input("\n\nAby zakończyć, naciśnij Enter")
```

Dwie pierwsze liczby są równe jeden.

Zmiennej *wynik* przypisujemy sumę dwóch poprzednich liczb, zmiennej *poprzedni2* przypisujemy wartość zmiennej *poprzedni1*, a zmiennej *poprzedni1* – wartość zmiennej *wynik*.

wywołanie funkcji *fibonacci(i)*

Ćwiczenie 4. Zapisujemy algorytm obliczania wartości elementu ciągu Fibonacciego w języku C++ lub Python

1. Utwórz nowy plik źródłowy i przepisz wybrany program z przykładu 2. Zapisz program w pliku pod nazwą *Liczby _Fibonacciego*.
2. Uruchom i przetestuj program dla kilku różnych danych. Wyjaśnij znaczenie poszczególnych wierszy programu.

3. Algorytmy porządkowania

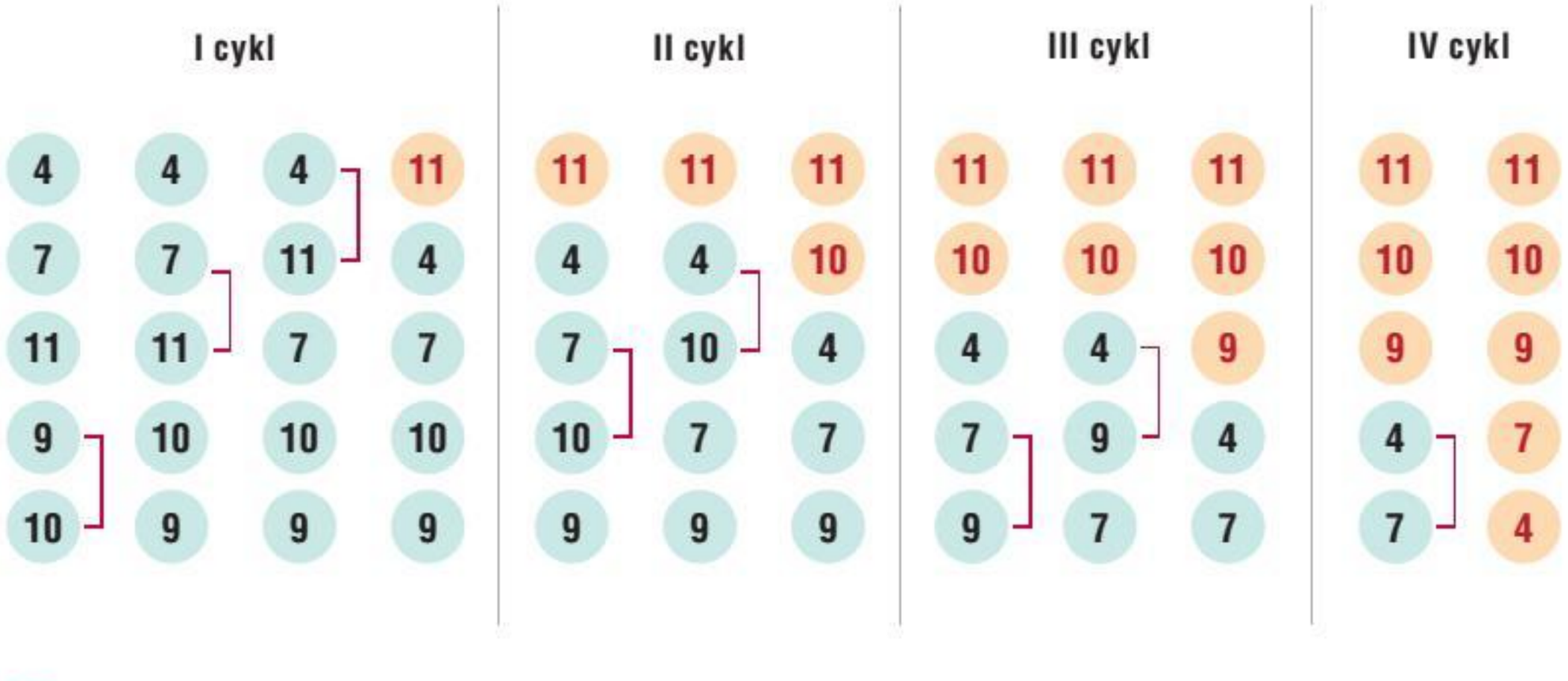
Chcemy uporządkować malejąco wprowadzone z klawiatury liczby.
W jaki sposób napisać program porządkujący elementy ciągu liczb?

Elementy ciągu liczb możemy uporządkować, stosując wybrany algorytm sortowania (porządkowania), np. **metodę bąbelkową** lub **przez wstawianie**.

3.1. Porządkowanie metodą bąbelkową

W algorytmie sortowania metodą bąbelkową w celu uporządkowania ciągu liczb – w tym przypadku malejąco – porównujemy sąsiednie liczby, rozpoczynając od porównania ostatniej i przedostatniej. Jeśli liczba ostatnia jest większa od przedostatniej, zamieniamy je miejscami.

Następnie liczbę, która stała się przedostatnią, porównujemy z trzecią od końca i przestawiamy, jeśli jest większa itd.



Rys. 2. *Przykład zastosowania algorytmu sortowania bąbelkowego*

Na rysunku 2. pokazano przykład sortowania bąbelkowego pięciu liczb – sortowanie przebiega od dołu ku górze. Po pierwszym cyklu ciąg nie jest jeszcze uporządkowany. Należy wykonywać kolejne cykle, aż ciąg zostanie uporządkowany i nie trzeba już będzie przestawić żadnego elementu.

W algorytmie sortowania bąbelkowego porównujemy parami kolejne elementy zbioru i przestawiamy je, jeśli są ustawione w niewłaściwej kolejności.
Zbiór jest przeglądany w tym samym kierunku tak długo, jak długo występuje w nim para elementów ustawionych w niewłaściwym porządku.

Nietrudno zauważyć, że już w pierwszym cyklu sortowania liczba największa (w tym przypadku liczba 11) zostaje ustawiona na właściwym miejscu. Aby ustawić we właściwej kolejności pozostałe liczby, należy powtarzać cykl sortowania, dopóki nie zostaną wykonane wszystkie przestawienia. Rysunek 2. jest poglądową ilustracją pokazującą działanie algorytmu sortowania bąbelkowego. Elementy o większych wartościach są wynoszone wyżej, co przypomina bąbelki powietrza (większe szybciej przemieszczają się ku górze) – stąd też wzięła się nazwa algorytmu.

Oczywiście tą metodą możemy uporządkować zbiór, zaczynając od góry, czyli od porównania elementu pierwszego z drugim, drugiego z trzecim itd.

Ćwiczenie 5. Stosujemy sortowanie bąbelkowe

Uporządkuj metodą bąbelkową malejąco (od największej do najmniejszej) liczby 34, 87, 45, 12, 11, 0, -56, -28, -11, 72. Podaj liczbę porównań, wykonanych w celu uporządkowania tego ciągu.

Wskazówki:

- Przygotuj kartki z zapisanymi przykładowymi liczbami.
- Ułóż kartki na stole lub na podłodze i przestawiaj parami. Zacznij od ostatniej.

Przykład 3. Zapisywanie algorytmu sortowania bąbelkowego w językach programowania C++ i Python

Zadanie: Uporządkuj malejąco *N* liczb, stosując algorytm sortowania bąbelkowego.

Dane: stała wartość *N*, tablica (lista) liczb *a*[*N*].

Wynik: Tablica (lista) liczb *a*[*N*], w której liczby ułożone są w porządku malejącym.

Opis rozwiązania:
W programie zastosujemy funkcje niezwracające wartości bez parametrów do wprowadzania danych do tablicy oraz do wyprowadzania danych z tablicy na ekran (temat C3, przykład 1.).
Zdefiniujemy nową funkcję niezwracającą wartości bez parametrów: *sort_babel*, która posłuży nam do posortowania danych.
Funkcje *wprowadz_dane*, *wyprowadz_dane* i *sort_babel* wywołujemy w programie głównym.

C++

```
23  void sort_babel()
24  {
25      for(int i = N-1; i > 0; i--)
26          for(int j = 0; j < i; j++)
27              if(a[j] < a[j + 1])
28              {
29                  int t = a[j];
30                  a[j] = a[j + 1];
31                  a[j + 1] = t;
32              }
33  }
34
35  int main()
36  {
37      wprowadz_dane();
38      cout << endl;
39      cout << "Wprowadzone dane:" << endl;
40      wyprowadz_dane();
41      cout << "Dane posortowane malejaca:" << endl;
42      sort_babel();
43      wyprowadz_dane();
44      return 0;
45  }
```

zamiana elementów miejscami

wywołanie funkcji sortującej dane

Python

```
def sort_babel():
    for i in range(N - 1, 0, -1):
        for j in range(i):
            if a[j] < a[j+1]:
                t = a[j]
                a[j] = a[j + 1]
                a[j + 1] = t

wprowadz_dane()
print("Wprowadzone dane:")
wyprowadz_dane()
print("Dane posortowane malejąco:")
sort_babel()
wyprowadz_dane()

input("\n\nAby zakończyć, naciśnij Enter")
```

zamiana elementów miejscami

wywołanie funkcji sortującej dane

Ćwiczenie 6. Zapisujemy algorytm sortowania bąbelkowego w językach programowania C++ i Python

1. W wybranym języku programowania napisz program sortujący dane wprowadzone do tablicy (listy) składającej się z pięciu elementów. Wykorzystaj funkcje *wprowadz_dane*, *wyprowadz_dane* z przykładu 1. (temat C3), a następnie wybrany fragment programu podany w przykładzie 3.
2. Zapisz program w pliku pod nazwą *Sortowanie_babelkowe*.
3. Uruchom i przetestuj program dla kilku różnych danych. Wyjaśnij znaczenie poszczególnych wierszy programu.

3.2. Porządkowanie przez wstawianie

Metodę sortowania (porządkowania) **przez wstawianie** (inaczej **przez umieszczanie**) stosujemy często w życiu codziennym – na przykład gdy odkładamy na półkę książkę w bibliotece, znajdujemy odpowiednie miejsce, patrząc na nazwiska autorów książek już stojących na półce.

W metodzie sortowania przez wstawianie porządkowany ciąg dzielimy na dwie części (rys. 3.):

- uporządkowaną (na początku działania algorytmu nie zawiera ona żadnych elementów),
- nieuporządkowaną (na początku zawiera wszystkie elementy ciągu).

Z części nieuporządkowanej pobieramy kolejne elementy i umieszczamy je w odpowiednich miejscach w części uporządkowanej.

Aby znaleźć miejsce umieszczenia nowego elementu w ciągu uporządkowanym, zastosujemy metodę **liniową**, w której porównuje się wstawiany element z kolejnymi elementami ciągu uporządkowanego.

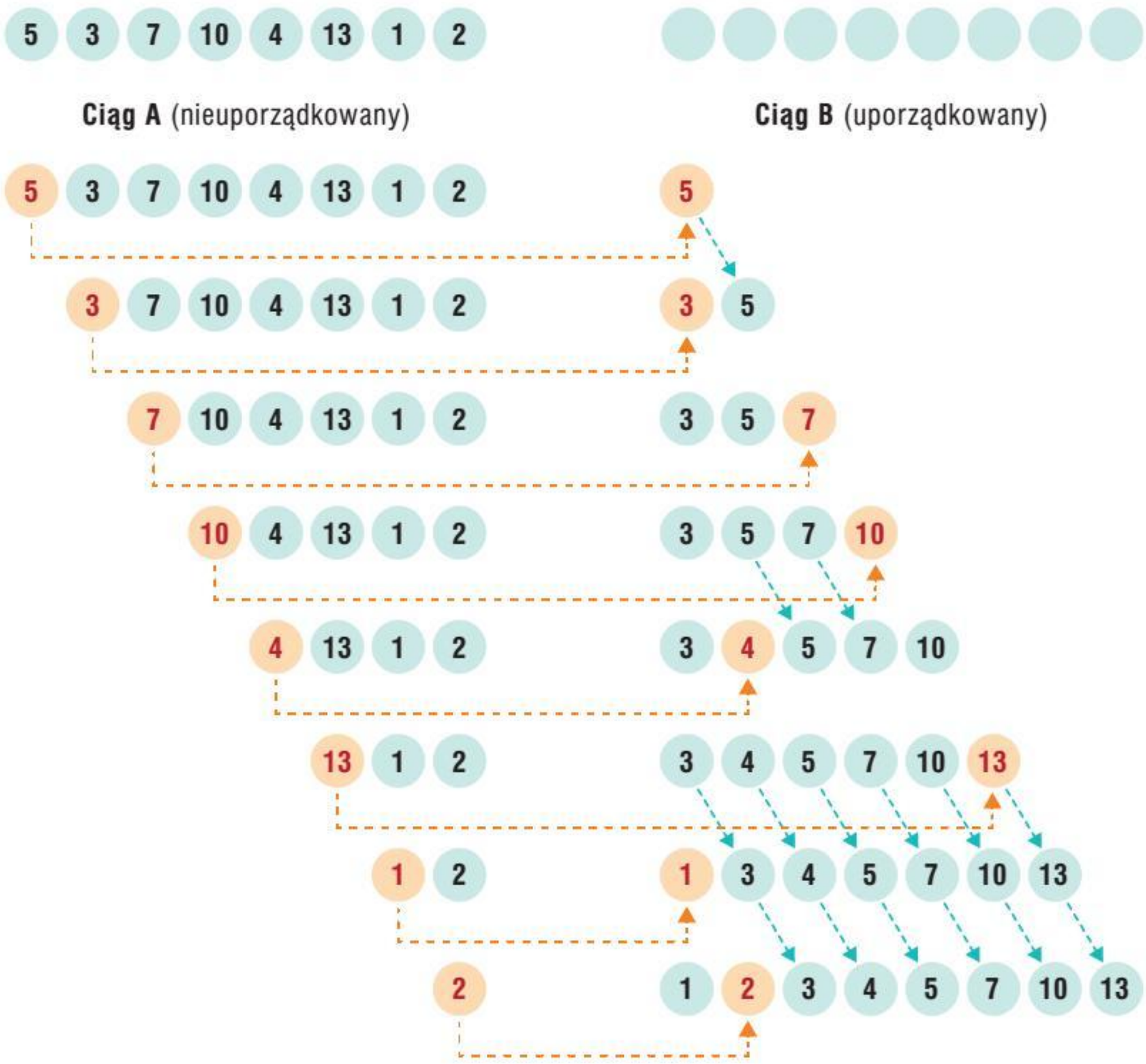
Ćwiczenie 7. Analizujemy algorytm sortowania metodą przez wstawianie

Przeanalizuj algorytm porządkowania przez wstawianie, wykonując go dla ciągu (3, 4, 10, 7, 1, 5, 2, 8).

Wskazówka: Przygotuj niezbędne pomoce dydaktyczne, np. kartoniki z zapisanymi liczbami.

Ćwiczenie 8. Sprawdzamy liczbę porównań elementów

Podaj liczbę porównań, wykonanych w celu uporządkowania ciągu z ćwiczenia 7.



Rys. 3. *Przykład działania algorytmu sortowania przez wstawianie*

Przykład 4. Zapisywanie algorytmu sortowania ciągu elementów metodą przez wstawianie w językach programowania C++ i Python

Zadanie: Uporządkuj rosnąco ciąg elementów liniową metodą przez wstawianie.

Dane: stała wartość *N*, tablica (lista) liczb *a*[*N*].

Wynik: Tablica (lista) liczb *b*[*N*], w której liczby ułożone są w porządku rosnącym.

Opis rozwiązania:

W programie zastosujemy funkcje niezwracające wartości bez parametrów do wprowadzania danych do tablicy (listy) oraz do wyprowadzania danych z tablicy (z listy) na ekran (skorzystamy z funkcji z przykładu 3.).

Zdefiniujemy nową funkcję niezwracającą wartości bez parametrów: *sort_wstaw*, która posłuży nam do posortowania danych.

Funkcje *wprowadz_dane*, *wyprowadz_dane* i *sort_wstaw* wywołujemy w programie głównym.

C++

```
23  void sort_wstaw()
24  {
25      int b[N];
26      int licz_posort = 0;
27
28      for(int i = 0; i < N; i++)
29      {
30          int akt = a[i];
31          int indB = licz_posort;
32          for(int j = 0; j < licz_posort; j++)
33              if(b[j]>akt)
34              {
35                  indB = j;
36                  break;
37              }
38          for(int k = licz_posort; k > indB; k--)
39              b[k] = b[k - 1];
40          b[indB] = akt;
41          licz_posort++;
42      }
43      for(int i = 0; i < N; i++)
44          a[i] = b[i];
45  }
46
47  int main()
48  {
49      wprowadz_dane();
50      cout << endl;
51      cout << "Wprowadzone dane:" << endl;
52      wyprowadz_dane();
53      cout << "Dane posortowane rosnaco:" << endl;
54      sort_wstaw();
55      wyprowadz_dane();
56      return 0;
57  }
```

zapamiętanie dla każdego elementu tablicy aktualnie sprawdzanego elementu

liczba elementów posortowanych

znalezienie punktu wstawienia

przesunięcie elementów w tablicy docelowej, od końca

przepisanie tablicy docelowej (posortowanej) do tablicy pierwotnej

Python

```
def sort_wstaw():
    b = [0] * N
    licz_posort = 0

    for i in range (N):
        akt = a[i]
        indB = licz_posort
        for j in range(licz_posort):
            if b[j] > akt:
                indB = j
                break
        for k in range(licz_posort,indB, -1):
            b[k] = b[k - 1]
        b[indB] = akt
        licz_posort = licz_posort + 1
    for i in range(N):
        a[i] = b[i]

wprowadz_dane()
print("Wprowadzone dane:")
wyprowadz_dane()
print("Dane posortowane rosnaco:")
sort_wstaw()
wyprowadz_dane()

input("\n\nAby zakończyć, naciśnij Enter")
```

zapamiętanie dla każdego elementu tablicy aktualnie sprawdzanego elementu

liczba elementów posortowanych

znalezienie punktu wstawienia

przesunięcie elementów w tablicy docelowej, od końca

przepisanie tablicy docelowej (posortowanej) do tablicy pierwotnej

Ćwiczenie 9. Zapisujemy algorytm sortowania metodą przez wstawianie w językach C++ i Python

1. W wybranym języku programowania napisz program sortujący metodą przez wstawianie dane wprowadzone do tablicy składającej się z pięciu elementów. Wykorzystaj funkcje *wprowadz_dane*, *wyprowadz_dane* z przykładu 3., a następnie wybrany fragment programu podany w przykładzie 4.
2. Zapisz program w pliku pod nazwą *Sortowanie_wstawianie*.
3. Uruchom i przetestuj program dla kilku różnych danych. Wyjaśnij znaczenie poszczególnych wierszy programu.

Warto zapamiętać

- Do znajdowania kolejnych liczb pierwszych można zastosować podejście polegające na sprawdzaniu kolejnych potencjalnych dzielników danej liczby.
- Ciągiem Fibonacciego nazywamy ciąg liczb naturalnych, w którym każda liczba (z wyjątkiem pierwszej i drugiej) jest sumą dwóch poprzednich.
- Algorytm znajdowania liczb pierwszych i wyznaczania liczb Fibonacciego można zrealizować, stosując technikę iteracji.
- Sortowanie bąbelkowe ciągu *n* liczb polega na porównywaniu parami kolejnych liczb i przestawianiu ich, jeśli występują w niewłaściwej kolejności.
- Sortowanie przez wstawianie ciągu *n* liczb polega na porównywaniu wstawianego elementu z kolejnymi elementami ciągu już uporządkowanego i umieszczaniu ich w odpowiednim miejscu.

Pytania i polecenia

1. Na czym polega algorytm badania pierwszości liczby? Przedstaw na przykładzie.
2. Na czym polega algorytm wyznaczania liczb Fibonacciego? Wyjaśnij, korzystając z rysunku 1.
3. Na czym polega algorytm sortowania bąbelkowego?
4. Dlaczego w sortowaniu bąbelkowym (rosnąco) element największy już w pierwszym przebiegu przechodzi na właściwą pozycję?
5. Od czego zależy liczba wykonanych operacji „zamiany miejscami elementów" w algorytmie sortowania bąbelkowego?
6. Jakie struktury danych powinno się zastosować, zapisując algorytm sortowania bąbelkowego w językach C++ lub Python?
7. Na czym polega algorytm sortowania przez wstawianie?

Zadania

1. Ilu porównań wymaga wykonanie algorytmu sortowania bąbelkowego i algorytmu sortowania przez wstawianie dla następujących ciągów liczb:
   a. (4, 3, 12, 3, 20, 2, –2, 7),
   b. (2, 5, 12, 18, 90, 112, 300, 350),
   c. (67, 34, 12, 10, 6, 0, -3, -13)?

2. Korzystając z wybranej metody sortowania, napisz program porządkujący rosnąco liczby dostaw telewizorów i oddzielnie liczby dostaw głośników dostarczanych do sklepu RTV. Skorzystaj z programu utworzonego w zadaniu 7. z tematu C3 i programu utworzonego w ćwiczeniu 6. w tym temacie. Zapisz program w pliku pod nazwą *Dostawy_uporzadkowane*.
3. Korzystając z metody sortowania bąbelkowego, uporządkuj malejąco ciąg wszystkich dzielników liczby 48. Jakiej metody użyjesz do znalezienia tych dzielników?
Wskazówka: Liczbę dzielników danej liczby znajdujemy, dokonując jej rozkładu na czynniki pierwsze i mnożąc przez siebie liczbę wystąpień każdego z czynników powiększoną o 1:
$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3^1$.
Zatem liczba dzielników liczby 48 wynosi $D_{48} = (4 + 1)\cdot(1 + 1) = 5 \cdot 2 = 10$.

Dla zainteresowanych

4. Napisz program realizujący rozwiązanie zadania 3. Zapisz program w pliku pod nazwą *Dzielniki_liczby_48*.
5. Korzystając z wybranej metody sortowania, napisz program porządkujący rosnąco *n*-elementowy zbiór liczb rzeczywistych według ich wartości bezwzględnych. Zapisz program w pliku pod nazwą *Wartosci_bezwgledne*.

Przeczytaj, jeśli chcesz wiedzieć więcej…

Przydomek **Fibonacci** oznacza: Filius (z łac. „syn") Bonacciego. Fibonacci nazywał się w rzeczywistości Leonardo Pisano (Leonardo z Pizzy). Żył w latach ok. 1170-1240. Studiował księgi Muhammada Al-Chuwarizmiego i w 1202 roku opublikował dzieło „Liber Abaci", dzięki któremu Europejczycy poznali cyfry arabskie i dziesiętny system liczenia. Właśnie w tym dziele Fibonacci zawarł rozważania na temat liczebności populacji królików…

n	F_n	F_{n+1}/F_n
1	1	1,000000
2	1	2,000000
3	2	1,500000
4	3	1,666667
5	5	1,600000
6	8	1,625000
7	13	1,615385
8	21	1,619048
9	34	1,617647
10	55	1,618182
11	89	1,617978
12	144	1,618056
13	233	1,618026
14	377	1,618037
15	610	1,618033

Tabela 1. *Ilorazy kolejnych liczb Fibonacciego*

Ciąg Fibonacciego, z pozoru prosty, ma wiele interesujących własności. Istnieje nawet towarzystwo Fibonacci Association, zajmujące się publikowaniem wyników licznych prac badawczych dotyczących tego zagadnienia. Ciekawą własność mają ilorazy kolejnych wyrazów ciągu Fibonacciego. W tabeli 1. zostały one przedstawione z dokładnością do sześciu miejsc po przecinku. Łatwo zaobserwować, jak zbliżają się do liczby

$1{,}61803398874989484820\ldots = \frac{1+\sqrt{5}}{2}$

zwanej **złotą średnią**

lub **złotym podziałem** (z łac. *divina proportio* – „boski podział").
Liczba ta jest proporcją najczęściej spotykaną w przyrodzie i stosowaną już przez starożytnych mistrzów. W ciele ludzkim – i to zarówno w całej jego postaci, jak i w wielu jego częściach – występują złote proporcje. Złoty podział można także odnaleźć chociażby w układzie listków na gałęzi.

Wokół Internetu i projektów

D

Moduł

Temat D1. Wybrane przepisy prawa dotyczące technologii informacyjno-komunikacyjnych

Temat D2. Komunikacja i wymiana informacji w Internecie

Temat D3. Tworzenie stron WWW z wykorzystaniem stylów i elementów programowania

Temat D4. Zadania projektowe



Temat D1. Wybrane przepisy prawa dotyczące technologii informacyjno--komunikacyjnych

1. Prawo autorskie
2. Korzystanie z cudzych utworów
3. Ochrona wizerunku
4. Zasady korzystania z programów komputerowych
5. Przestępczość komputerowa

Warto powtórzyć

1. Jakie znasz zasady korzystania z programów komputerowych?
2. Co określa licencja na program komputerowy?
3. Jakie znasz przykłady darmowych licencji?
4. Jakie znasz przejawy przestępczości komputerowej?

1. Prawo autorskie

Przepisy prawa autorskiego reguluje ustawa z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych. Najistotniejsze zasady prawa autorskiego omówimy na podstawie kilku wybranych artykułów tej ustawy.

§ Co jest przedmiotem prawa autorskiego?

Art. 1. *1. Przedmiotem prawa autorskiego jest każdy przejaw działalności twórczej o indywidualnym charakterze, ustalony w jakiejkolwiek postaci, niezależnie od wartości, przeznaczenia i sposobu wyrażenia (utwór).*
2. W szczególności przedmiotem prawa autorskiego są utwory:
1) wyrażone słowem, symbolami matematycznymi, znakami graficznymi (literackie, publicystyczne, naukowe, kartograficzne oraz programy komputerowe),
2) plastyczne,
3) fotograficzne,
4) lutnicze,
5) wzornictwa przemysłowego,
6) architektoniczne, architektoniczno-urbanistyczne i urbanistyczne,
7) muzyczne i słowno-muzyczne,
8) sceniczne, sceniczno-muzyczne, choreograficzne i pantomimiczne,
9) audiowizualne (w tym filmowe).

Prawo autorskie chroni m.in. programy komputerowe. **Utwór**, również program, jest przedmiotem prawa autorskiego od chwili **ustalenia**, nawet jeśli nie został skończony i **utrwalony**, czyli zapisany na kartce czy na nośniku pamięci masowej komputera.

Ustalenie utworu może polegać na przykład na ułożeniu (w głowie) ciągu poleceń tworzących program komputerowy i zapoznaniu z nimi choćby jednej osoby.

Kupując programy komputerowe, nie nabywamy praw autorskich do nich. W przypadku programów zapisanych na fizycznych nośnikach (np. na płytach CD) stajemy się jedynie właścicielami nośnika.

Prawo autorskie chroni twórców niezależnie od ich wieku. Utwory osób małoletnich również są chronione.

§ Co nie podlega prawu autorskiemu?

> ***Art. 4.*** *Nie stanowią przedmiotu prawa autorskiego:*
> *1) akty normatywne lub ich urzędowe projekty,*
> *2) urzędowe dokumenty, materiały, znaki i symbole,*
> *3)* ***opublikowane*** *opisy patentowe lub ochronne,*
> *4) proste informacje prasowe.*

We własnym opracowaniu, np. w dokumencie tekstowym czy na swojej stronie internetowej, można umieszczać informacje o prognozie pogody, proste informacje prasowe (np. o wypadkach, wydarzeniach politycznych lub sportowych), program radiowy czy telewizyjny. Można również umieszczać fragmenty aktów prawnych, np. treść wybranych artykułów różnych ustaw.

U **Utwór opublikowany**

Utwór, który za zezwoleniem twórcy został zwielokrotniony i którego egzemplarze udostępniono publicznie.

2. Korzystanie z cudzych utworów

§ Na czym polega „dozwolony użytek chronionych utworów”?

> ***Art. 23.*** *1. Bez zezwolenia twórcy wolno nieodpłatnie korzystać z już rozpowszechnionego utworu w zakresie własnego użytku osobistego. [...]*
> *2. Zakres własnego użytku osobistego obejmuje korzystanie z pojedynczych egzemplarzy utworów przez krąg osób pozostających w związku osobistym, w szczególności pokrewieństwa, powinowactwa lub stosunku towarzyskiego.*
> ***Art. 35.*** *Dozwolony użytek nie może naruszać normalnego korzystania z utworu lub godzić w słuszne interesy twórcy.*

Dopuszczalne jest wykonywanie kopii pojedynczych, legalnie **rozpowszechnionych** egzemplarzy utworu lub jego fragmentów jedynie na użytek własny i osób najbliższych.

U **Utwór rozpowszechniony**

Utwór, który za zezwoleniem twórcy został w jakikolwiek sposób udostępniony publicznie.

Możemy na przykład wykonać, do użytku osobistego, kopię zakupionej płyty CD z muzyką, aby słuchać jej w samochodzie. Przepisów o dozwolonym użytku nie stosuje się do programów komputerowych.

Co grozi za rozpowszechnianie cudzego utworu?

> ***Art. 116.*** *1. Kto bez uprawnienia albo wbrew jego warunkom rozpowszechnia cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie, fonogram, wideogram lub nadanie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.*

Nie wolno na przykład umieścić w Internecie plików muzycznych z zakupionej płyty CD, tak aby każdy mógł je pobrać. Nie można również umieścić w Internecie, np. na własnej stronie internetowej czy na stronie szkoły, podręcznika szkolnego lub jego większych fragmentów. Jeśli nie nabyliśmy prawa do dystrybucji tych utworów, takie działanie jest nielegalne.

Aby wykorzystać na własnej stronie internetowej cudzy utwór (np. tekst, grafikę, muzykę), należy dowiedzieć się, do kogo należą prawa autorskie do utworu, a następnie skontaktować się z uprawnioną osobą.

Kiedy można skorzystać z cudzego utworu bez pytania o zgodę?

> ***Art. 29.*** *Wolno przytaczać w utworach stanowiących samoistną całość urywki rozpowszechnionych utworów oraz rozpowszechnione utwory plastyczne, utwory fotograficzne lub drobne utwory w całości, w zakresie uzasadnionym celami cytatu, takimi jak wyjaśnianie, polemika, analiza krytyczna lub naukowa, nauczanie lub prawami gatunku twórczości.*
> ***Art. 29[1].*** *Wolno korzystać z utworów na potrzeby parodii, pastiszu lub karykatury, w zakresie uzasadnionym prawami tych gatunków twórczości.*

Jeśli chcemy zacytować, np. w wypracowaniu szkolnym lub w publikacji internetowej, fragment cudzego utworu, możemy skorzystać z tzw. **prawa cytatu** i zrobić to bez zgody autora. Musimy jednak podać autora i tytuł cytowanego utworu, a w przypadku utworu rozpowszechnionego w Internecie – dodatkowo adres URL strony źródłowej.

Poza cytatem tekstowym wyróżnia się m.in. cytaty graficzne, filmowe, muzyczne.

Stosowanie cytatu powinno być zawsze uzasadnione. Na przykład cytat może posłużyć jako teza do rozwinięcia tematu lub podsumowanie opracowania. Można też na przykładzie fragmentu cudzego utworu wyrazić własną opinię. Nie należy natomiast stosować cytatów jako „wypełniaczy" naszego wypracowania, tłumacząc się brakiem czasu lub pomysłu.

Bez ograniczeń można korzystać z utworów znajdujących się w tzw. **domenie publicznej** (ang. *public domain*), ponieważ autorskie prawa majątkowe do tych utworów wygasły. Autorskie prawa majątkowe wygasają po siedemdziesięciu latach od śmierci twórcy, a w przypadku utworu nieznanego autora – po siedemdziesięciu latach

od daty pierwszego rozpowszechnienia. Na przykład bez ograniczeń można umieścić na swojej stronie internetowej „Odę do młodości" Adama Mickiewicza, pamiętając jednak, że nie wolno pominąć nazwiska autora – nie wygasają bowiem autorskie prawa osobiste.

Do własnych celów można także wykorzystać utwory rozpowszechniane na zasadach Creative Commons (CC). Ten typ licencji pozwala twórcom zachować prawa autorskie, a jednocześnie dzielić się swoją twórczością z innymi.

W Wikipedii zbiory znajdujące się w domenie publicznej oznaczono symbolem: , a zbiory na licencji Creative Commons – symbolem .

Ćwiczenie 1. Wyszukujemy w Internecie obrazy objęte darmową licencją

Korzystając z Wikipedii, odszukaj obrazy: „Konstytucja 3 Maja 1791 roku" Jana Matejki, „Sobieski pod Wiedniem" Juliusza Kossaka, „Śpiący Staś" Stanisława Wyspiańskiego, oraz zdjęcia: komputera Eniac, pierwszej maszyny do pisania, superkomputera Cray-2 i telefonu z bakelitu z tarczą numerową (z 1931 roku). Sprawdź, które z odszukanych grafik znajdują się w zbiorach domeny publicznej, a które są objęte licencją Creative Commons.

W Internecie możemy znaleźć serwisy (strony internetowe), które udostępniają bezpłatnie grafikę (np. zdjęcia, animowane GIF-y, przyciski ekranowe, ikony, cliparty).

Korzystając z takiej grafiki, na przykład w celu umieszczenia jej na własnej witrynie lub wykorzystania we własnej publikacji, należy przestrzegać ograniczeń określonych w licencji podanej na stronie serwisu. Przykładowe ograniczenia to: zakaz likwidowania firmowych oznaczeń, które zostały umieszczone na grafice, zakaz darmowego udostępniania innym pobranej grafiki, obowiązek podawania za każdym razem źródła (adresu strony internetowej z darmowymi zbiorami grafiki). Niektóre serwisy udostępniają również grafikę za niewielką opłatą.

Ćwiczenie 2. Szukamy w Internecie serwisów z darmowymi zdjęciami

Znajdź w Internecie serwis udostępniający bezpłatnie zdjęcia. Podaj adres internetowy tego serwisu. Zapoznaj się z warunkami korzystania ze zdjęć umieszczonych w tym serwisie.

Popularne jest udostępnianie plików muzycznych i filmów poprzez programy typu *peer-to-peer* (w skrócie P2P). Dozwolone jest jednak wyłącznie dzielenie się utworami, których jesteśmy autorami.

Możemy również poszukać muzyki w serwisach, które udostępniają ją odpłatnie, gwarantując bezpieczeństwo i dobrą jakość nagrania.

Niestety, bardzo często mamy do czynienia z działalnością nielegalną. Wiele plików z muzyką i filmami, które są umieszczone w sieciach P2P lub w innych serwisach, to utwory chronione prawem autorskim, w stosunku do których licencja zabrania publicznego rozpowszechniania czy udostępniania. Jedynie właściciel praw

autorskich (twórca, wydawca) jest uprawniony do rozporządzania danym dziełem. Rozpowszechniając chroniony prawem utwór, nie możemy tłumaczyć się, że promujemy w ten sposób autora, ponieważ może on sobie nie życzyć, aby jego utwór był reklamowany na naszej stronie.

Muzykę legalnie udostępniają na swoich stronach internetowych sami artyści lub wytwórnie płytowe. Udostępniając próbki utworów, promują na przykład nową płytę lub najbliższy koncert.

W sieciach P2P nie wolno też udostępniać programów komercyjnych.

Nie wolno:

- rozpowszechniać cudzych utworów podlegających prawu autorskiemu (np. tekstu książki, muzyki, grafiki, filmów, programów komputerowych);
- zamieszczać na własnej stronie internetowej cudzych utworów i podpisywać ich własnym nazwiskiem;
- wstawiać do własnego tekstu fragmentów cudzych tekstów jako własnych;
- przerabiać i łączyć cudzych utworów bez zgody i wiedzy autora.

Ćwiczenie 3. Kiedy możemy skorzystać z cudzego materiału?

Czy poniższe stwierdzenia są poprawne? Uzasadnij odpowiedzi.

a. Mogę umieścić cytaty z książki Stanisława Lema „Solaris" w publikacji internetowej na temat literatury science fiction.
b. Mogę umieścić krótki wiersz Tadeusza Różewicza w wypracowaniu na temat jego poezji.
c. Kupiłem płytę CD z muzyką. Mogę umieścić na swojej stronie internetowej utwór muzyczny z tej płyty i udostępniać go bezpłatnie.
d. Mogę samodzielnie skomponowany utwór udostępnić w sieci P2P.

3. Ochrona wizerunku

§ Na czym polega ochrona wizerunku?

Art. 81. *1. Rozpowszechnianie wizerunku wymaga zezwolenia osoby na nim przedstawionej. W braku wyraźnego zastrzeżenia zezwolenie nie jest wymagane, jeżeli osoba ta otrzymała umówioną zapłatę za pozowanie.*
2. Zezwolenia nie wymaga rozpowszechnianie wizerunku:
1) osoby powszechnie znanej, jeżeli wizerunek wykonano w związku z pełnieniem przez nią funkcji publicznych, w szczególności politycznych, społecznych, zawodowych,
2) osoby stanowiącej jedynie szczegół całości takiej jak zgromadzenie, krajobraz, publiczna impreza.

Na przykład można na szkolnej witrynie internetowej umieszczać zdjęcia ze szkolnej uroczystości, takiej jak rozpoczęcie czy zakończenie roku szkolnego. Uczniowie uczestniczący w takiej uroczystości stanowią bowiem na fotografiach szczegół całości. Natomiast umieszczenie na szkolnej stronie zdjęć każdego ucznia danej klasy

z osobna czy nauczycieli uczących w tej klasie wymaga ich zgody na rozpowszechnianie wizerunku.

Ćwiczenie 4. Sprawdzamy wybrane przepisy prawa autorskiego

Czy poniższe stwierdzenia są poprawne? Uzasadnij odpowiedzi.

a. Mogę umieścić na szkolnej witrynie zdjęcia krajobrazów górskich ze szkolnej wycieczki w Bieszczady. Na zdjęciach widoczni są również koledzy i koleżanki z klasy. Rozpatrz dwie sytuacje: gdy osoby widoczne na zdjęciu pozują do fotografii i gdy zostały przypadkowo ujęte w kadrze.
b. Utworzona przez piętnastoletniego ucznia gra komputerowa jest chroniona przepisami prawa autorskiego.
c. Mogę zamieścić treść tematu D1 z podręcznika na swojej stronie internetowej.

4. Zasady korzystania z programów komputerowych

Do programów komputerowych nie stosuje się przepisów o dozwolonym użytku, a więc wykonywanie kopii programów na użytek własny i osób najbliższych może nie być dozwolone. Szczegółowe zasady korzystania z programu komputerowego określa **licencja**.



Na każdy program, z którego korzystamy, należy mieć licencję. Niektórych programów można używać bezpłatnie, ale muszą mieć odpowiedni status, np. **freeware**, **shareware**.

L **Licencja**

Umowa zawierana pomiędzy osobą (instytucją), która jest właścicielem majątkowych praw autorskich do utworu (np. producentem lub dystrybutorem programu), a odbiorcą utworu (np. użytkownikiem oprogramowania) określająca warunki korzystania z utworu.

Jeśli instalujemy program na swoim komputerze, wyrażamy równocześnie zgodę na warunki licencji. Licencja określa m.in.:

- prawa i obowiązki nabywcy, np. zakaz kopiowania programu, możliwość korzystania z programu w celach zarobkowych;
- liczbę komputerów, na których można instalować oprogramowanie, np. licencja jednostanowiskowa (na jednym komputerze), grupowa (w sieci lub w zestawie komputerów, np. w szkolnej pracowni);
- procedurę aktualizowania programu;
- czas trwania prawa do korzystania z programu.

Oprócz licencji komercyjnych (wymagających zapłaty za prawo do korzystania z programu) istnieją również inne rodzaje licencji:

- **Shareware** – umożliwia bezpłatne korzystanie z programu w celu jego testowania. Niektóre programy o tym statusie mogą mieć różne ograniczenia: limit liczby uruchomień lub limit czasowy (**trialware**). Po wyczerpaniu tych limitów za program trzeba zapłacić lub zrezygnować z korzystania z niego (usunąć z komputera). Innym ograniczeniem może być niedostępność części funkcji programu. Ograniczenie tego rodzaju posiadają programy w wersji demonstracyjnej

(**demoware**). Po zakupieniu licencji można korzystać z pełnej wersji. Najbardziej popularne są wersje demonstracyjne gier.

- **Postcardware** (lub **cardware**) – umożliwia bezpłatne korzystanie z programu w zamian za przesłanie kartki pocztowej do autora programu. Najlepiej, jeśli kartka jest z miejscowości, w której mieszka użytkownik. Wersja **emailware** wymaga przysłania e-maila.
- **Adware** – umożliwia bezpłatne korzystanie z programu, ale w programach są umieszczone reklamy. Można kupić wersję takiego programu bez reklam.
- **Freeware** – umożliwia bezpłatne używanie programu, bez ograniczeń czasowych. Prawa autorskie autora pozostają nadal w mocy. Program można rozpowszechniać tylko w niezmienionej formie (nie wolno go modyfikować) i nie może być on dystrybuowany odpłatnie przez osoby trzecie. Ewentualnie możliwe jest pobieranie opłat za produkty utworzone z wykorzystaniem takiego programu. W przypadku niektórych programów status freeware dotyczy indywidualnych użytkowników, natomiast od firm i instytucji wymagane jest uiszczenie opłaty licencyjnej.
- **Powszechna Licencja Publiczna GNU GPL** (z ang. GNU ***General Public License***) – ten rodzaj licencji, w odróżnieniu od licencji freeware, pozwala na przykład na modyfikowanie i udoskonalanie kodu źródłowego, który jest dla wszystkich dostępny. Poprawiony program możemy rozpowszechniać, ale pod warunkiem dołączenia do zmodyfikowanej wersji oryginalnego kodu źródłowego.
- **Public domain** (domena publiczna) – pozwala na bezpłatne korzystanie z programu, jego rozpowszechnianie, a także modyfikowanie, ponieważ umieszczenie programu w domenie publicznej oznacza, że twórca programu zrzeka się praw autorskich do niego.

Nazwy freeware, shareware, adware itd. stosuje się również do określenia programu, np. program typu freeware, program typu shareware.



Zawsze, gdy chcemy zainstalować program na komputerze, należy zapoznać się z warunkami licencji – zarówno w przypadku korzystania z darmowego programu, jak i zakupu oprogramowania.



Ćwiczenie 5. Sprawdzamy licencje na wybrane programy

Sprawdź, na podstawie jakich licencji są rozpowszechniane programy: WinRAR, GIMP, Inkscape, Tux Paint, komunikator Skype, system operacyjny Linux, pakiet programów LibreOffice.

5. Przestępczość komputerowa

Przestępstwo komputerowe to przestępstwo, w którym narzędziem służącym do jego popełnienia jest komputer lub inne urządzenie elektroniczne.

Przestępstwa komputerowe podlegają odpowiedzialności karnej określonej w przepisach *Kodeksu karnego*.

Przykłady przestępstw komputerowych

- **Piractwo komputerowe** – bezprawne kopiowanie, rozpowszechnianie lub publikowanie programów komputerowych prawnie chronionych, np. nielegalna produkcja nośników z programem, a następnie ich sprzedawanie, skopiowanie od kolegi programu komputerowego, a także **nielegalne rozpowszechnianie i sprzedaż utworów** muzycznych, filmów i oprogramowania za pośrednictwem Internetu.
- **Hacking** – uzyskanie nieuprawnionego dostępu do systemu komputerowego („włamanie się") poprzez naruszenie jego zabezpieczeń, jednak bez wyrządzania dalszych szkód.
- **Podsłuch komputerowy** – przechwytywanie informacji przesyłanych pomiędzy systemami komputerowymi (np. treści wiadomości e-mail, SMS-ów).
- **Bezprawne niszczenie informacji** – nieupoważnione zmienianie bądź usuwanie danych lub programów komputerowych z cudzego komputera.
- **Oszustwo komputerowe** – nieupoważnione wpływanie na automatyczne przetwarzanie, gromadzenie lub przesyłanie informacji lub zmienianie, usuwanie albo wprowadzanie nowego zapisu na komputerowym nośniku informacji, np. włamanie się do sieci komputerowej banku i przelanie określonej kwoty na własne konto, dokonywanie w imieniu innej osoby nieuczciwych transakcji, wysłanie w imieniu innej osoby e-maila z nieprawdziwymi informacjami.
- **Sabotaż komputerowy** – zakłócanie lub paraliżowanie funkcjonowania systemów informatycznych o istotnym znaczeniu dla bezpieczeństwa państwa i jego obywateli.
- **Szpiegostwo komputerowe** – włączanie się do sieci komputerowej w celu uzyskania wiadomości o charakterze tajemnicy państwowej lub służbowej, której udzielenie obcemu wywiadowi może wyrządzić szkodę państwu.
- **Podrzucanie wirusów i rozsyłanie innych szkodliwych programów:**
  - powodujących niszczenie danych czy umożliwiających dostęp do poufnych informacji (np. wirusy i robaki);
  - umożliwiających uzyskanie różnych danych (np. nazwisk, adresów, numerów kart kredytowych, zawartości folderów, wykazu używanego oprogramowania, a nawet zdjęć użytkownika, jeśli ma zamontowaną kamerę internetową) i przesłanie ich do Internetu (np. spyware);
  - umożliwiających przekazanie na zewnątrz informacji z „zarażonego" komputera, a nawet przejęcie całkowitej kontroli nad komputerem przez niepowołaną osobę (np. trojany, rootkity);
  - szyfrujących dane na nośnikach – aby je odzyskać, należy przestępcom zapłacić okup (ang. *ransomware*).

Piractwo komputerowe ma negatywny wpływ na osoby prywatne, instytucje oraz społeczeństwo. Na przykład:

- twórcy dzieł (programu komputerowego, filmu, utworu muzycznego) nie otrzymują wynagrodzenia za wykorzystywanie pirackich kopii utworów. Aby zrekompensować straty, muszą podnosić ceny programów, płyt CD i DVD czy biletów do kina;

- zainstalowanie pirackiego programu na domowym komputerze naraża na odpowiedzialność karną całą rodzinę; zainstalowanie pirackiego programu na komputerze w firmie naraża firmę na wysokie koszty (odszkodowanie dla twórcy programu);
- pirackie kopie programów komputerowych mogą zawierać wirusy i inne niepożądane programy.



Warto zapamiętać

- Utwory będące przejawem działalności twórczej (np. programy komputerowe, utwory muzyczne, plastyczne, filmowe, literackie, architektoniczne) podlegają ochronie wynikającej z prawa autorskiego.
- Prawu autorskiemu nie podlegają m.in.: teksty ustaw, proste informacje prasowe, np. prognoza pogody, kursy walut, program telewizyjny, repertuar kin.
- Dozwolony użytek chronionych utworów polega na tym, że możemy korzystać nieodpłatnie z już rozpowszechnionego utworu bez zezwolenia twórcy w zakresie własnego użytku osobistego.
- Przepisy o dozwolonym użytku nie mają zastosowania do programów komputerowych.
- Na każdy program, z którego korzystamy, trzeba mieć licencję, która określa warunki, na jakich odbiorca może korzystać z programu. Korzystając z programu, trzeba przestrzegać tych warunków. Licencje mogą być odpłatne i nieodpłatne.
- Jeśli wykorzystujemy cudze utwory, nie wolno naruszać praw autorskich do nich. Trzeba zawsze sprawdzić, w jakim zakresie można z danego utworu swobodnie korzystać.
- Przestępstwa komputerowe (np. włamywanie się do cudzego komputera, piractwo komputerowe, podrzucanie wirusów komputerowych i innych złośliwych programów) podlegają odpowiedzialności karnej.



Pytania i polecenia

1. Czym zajmuje się prawo autorskie?
2. Co nie podlega prawu autorskiemu?
3. Na czym polega „dozwolony użytek chronionych utworów”?
4. Kiedy rozpowszechnianie wizerunku nie wymaga zezwolenia?
5. Co oznacza stwierdzenie, że „do programów komputerowych nie stosuje się przepisów o dozwolonym użytku”?
6. Czym jest licencja na program komputerowy i co określa?
7. Omów podstawowe rodzaje darmowych licencji.
8. Kiedy można korzystać z cudzego utworu bez pytania o zgodę?
9. Podaj przykład zastosowania „prawa cytatu”.
10. Kiedy wygasają autorskie prawa majątkowe?
11. Podaj przynajmniej cztery zasady korzystania z cudzych materiałów.
12. Co grozi za bezprawne rozpowszechnianie w Internecie cudzego utworu?
13. Omów przykłady przestępstw komputerowych.
14. Jakie szkody mogą spowodować działania pirackie w sieci?

Zadania

1. Czy poniższe stwierdzenia są poprawne? Odpowiedzi uzasadnij.
   a. Mogę umieścić samodzielnie nagrany film w serwisie YouTube.
   b. Nie mogę umieścić w sieci P2P programu typu freeware.
   c. Mogę umieścić grafikę, zdjęcia, diagramy, schematy znalezione w Internecie w tworzonej na lekcję geografii prezentacji multimedialnej na temat zmian powierzchni lasów na świecie.
   d. Mogę pobrać wybrany utwór ze strony znanego artysty i udostępnić go na swojej stronie internetowej.
   e. Mogę obejrzeć film umieszczony w sieci P2P.
2. Znajdź w Internecie serwis legalnie udostępniający muzykę. Podaj adres internetowy znalezionego serwisu. Zapoznaj się z warunkami korzystania z tego serwisu, m.in. pobierania plików muzycznych.
3. Sprawdź, czy program CorelDRAW można pobrać za darmo w celu testowania przez określoną liczbę dni.
4. Przygotuj dla kolegi lub koleżanki trzy pytania podobne do postawionych w ćwiczeniach 3. i 4. oraz w zadaniu 1. Opracuj również wzorcowe odpowiedzi.

Dla zainteresowanych

5. Znajdź w Internecie tekst ustawy o prawach autorskich i zapoznaj się ze wszystkimi artykułami dotyczącymi dozwolonego użytku chronionych utworów.
6. Odszukaj więcej informacji na temat przestępstw komputerowych, w szczególności dotyczących kar, które grożą za wybrane przestępstwa (skorzystaj z tekstu ustawy – *Kodeks karny*, który można znaleźć w Internecie). Przygotuj prezentację multimedialną na ten temat.

Temat D2. Komunikacja i wymiana informacji w Internecie

1. Wybrane formy komunikacji i wymiany informacji
2. Działanie poczty elektronicznej
3. Zasady netykiety
4. Chmura obliczeniowa
5. Korzyści i zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnych

Warto powtórzyć

1. Jakie formy komunikacji internetowej są najbardziej popularne?
2. Jakie inne urządzenia, poza komputerem, umożliwiają komunikowanie się z wykorzystaniem Internetu?
3. Czy możemy rozpowszechniać cudze utwory podlegające prawu autorskiemu? Uzasadnij odpowiedź.

1. Wybrane formy komunikacji i wymiany informacji

Nazwa	Opis	Wymagania
Poczta elektroniczna (e-mail)	usługa internetowa służąca do przesyłania wiadomości tekstowych, tzw. listów elektronicznych, do których można dołączać załączniki (np. teksty, obrazy, muzykę, filmy)	posiadanie konta pocztowego (firmowego, darmowego lub komercyjnego); dostęp do konta pocztowego za pomocą specjalnego programu albo aplikacji na smartfon (tzw. klienta poczty elektronicznej) lub strony WWW
Czat	serwis internetowy umożliwiający prowadzenie rozmów w czasie rzeczywistym	wejście na odpowiednią stronę WWW; niekiedy zainstalowanie odpowiedniego oprogramowania (dodatkowej wtyczki)
Komunikator internetowy	program komputerowy lub aplikacja na smartfon umożliwiające natychmiastowe przesyłanie komunikatów tekstowych, głosowych i obrazu, np. zdjęcia rozmówcy	zainstalowanie odpowiedniego programu (zwykle darmowego, np. Gadu-Gadu, Tlen, Skype, WhatsApp); posiadanie głośników, mikrofonu i kamery internetowej (albo smartfona) w celu ułatwienia komunikacji

Grupa dyskusyjna	usługa internetowa, która umożliwia publikowanie (wysyłanie) wiadomości zwanych artykułami (ang. *post*) oraz czytanie artykułów innych użytkowników, ale nie w czasie rzeczywistym	zapisanie się do grupy (zwane subskrybowaniem grupy), której wiadomości chcemy czytać (i brać udział w dyskusji); można również zakładać własne grupy; do grupy możemy się zapisać w wybranym programie pocztowym; korzystanie z grup umożliwia też wyszukiwarka Google
Forum dyskusyjne	forma grup dyskusyjnych realizowana za pomocą stron WWW; umożliwia dzielenie się poglądami i dyskusję na różne tematy poprzez pisanie tzw. postów	wejście na odpowiednią stronę WWW; można uczestniczyć w rozpoczętej dyskusji lub inicjować własne tematy, zwane wątkami
Serwis społecznościowy	serwis internetowy, który umożliwia interakcję użytkowników polegającą na współtworzeniu zasobów serwisu	zazwyczaj wymaga zalogowania się na stronie wybranego serwisu, np. *Facebook*, *Instagram* czy *Twitter*
Blog	rodzaj strony internetowej – sieciowy dziennik; zawiera chronologicznie uporządkowane wpisy, których autorem jest właściciel bloga	założenie bloga w serwisie WWW z blogami lub na portalu internetowym
Kanał filmowy	zbiór filmów lub transmisji na żywo należących do jednej osoby (instytucji), udostępnionych w serwisie umożliwiającym oglądanie filmów	posiadanie konta w serwisie umożliwiającym oglądanie filmów lub transmisji na żywo, np. *YouTube*, *Twitch*
FTP	protokół przesyłania plików pomiędzy komputerami za pośrednictwem Internetu; pliki przechowywane są w jednym miejscu (na serwerze), a dostęp do nich może mieć wiele komputerów (klientów)	zalogowanie się do serwera FTP (podanie identyfikatora i hasła); niezbędny jest odpowiedni program, zwany klientem FTP; podstawowe funkcje klienta FTP wbudowane są w większość przeglądarek internetowych; dostęp do konta FTP wymaga wpisania adresu URL w postaci: *ftp://użytkownik:hasło@adres_serwera*
P2P	model komunikacji umożliwiający wymianę plików pomiędzy komputerami bez pośrednictwa dedykowanego serwera; każdy komputer, oprócz funkcji klienta, jednocześnie pełni funkcję serwera	niezbędny jest odpowiedni program (można pobrać darmowy program z Internetu)

Chmura obliczeniowa	model komunikacji umożliwiający wymianę plików pomiędzy komputerami za pośrednictwem serwisu internetowego; pliki przechowywane są w chmurze obliczeniowej, a dokładnie na kilku (kilkunastu) serwerach umieszczonych w centrach obliczeniowych na całym świecie	wystarczy przeglądarka internetowa lub odpowiednie oprogramowanie

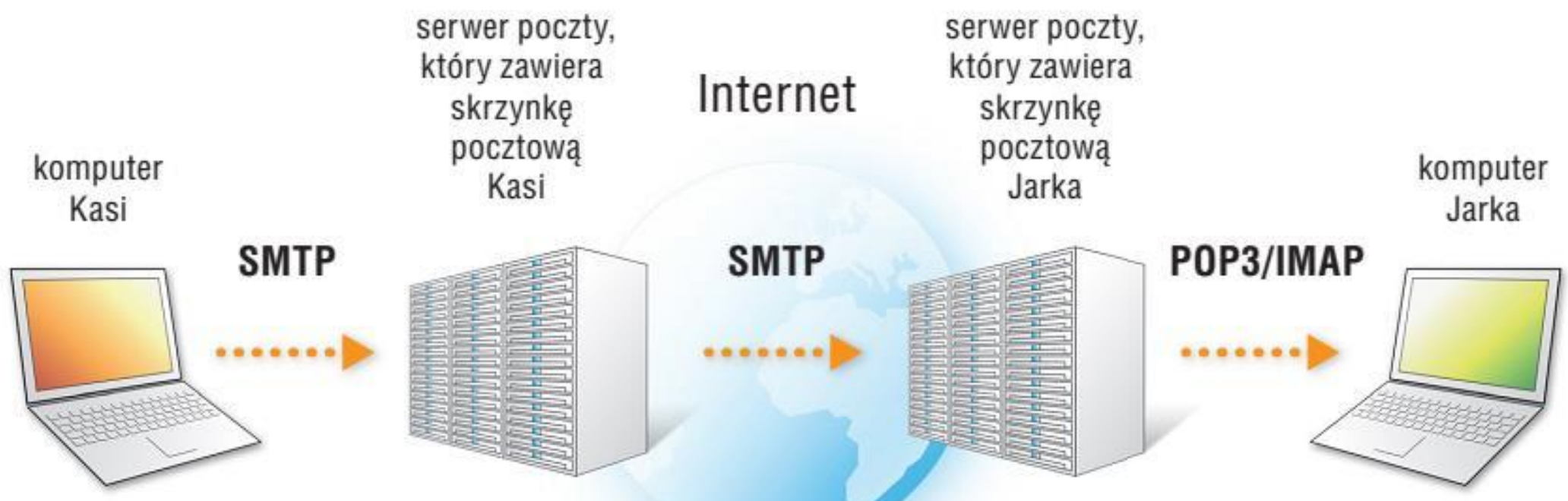
Tabela 1. *Zestawienie wybranych form komunikacji i wymiany informacji*



Pobierając z Internetu lub rozpowszechniając pliki chronione prawem autorskim, przestrzegaj przepisów prawa!

2. Działanie poczty elektronicznej

Do korzystania z poczty elektronicznej niezbędne jest posiadanie własnej skrzynki pocztowej – **konta poczty elektronicznej** (rys. 1.). Konto poczty może zostać przydzielone np. przez pracodawcę (konto firmowe), można założyć również konto prywatne, korzystając z rozwiązań darmowych (np. Gmail, Outlook) lub płatnych.



Rys. 1. *Schemat działania poczty elektronicznej. Kasia wysyła list do Jarka*

Z kontem pocztowym związany jest adres e-mail o postaci: *osoba@adres.komputera*. Adres ten oznacza, że użytkownik o nazwie *osoba* ma swój adres pocztowy na serwerze o adresie *adres.komputera*.

Dostęp do konta poczty elektronicznej możliwy jest poprzez odpowiednią stronę WWW (np. *www.gmail.com*, *www.wp.pl*, *www.onet.pl*, *www.o2.pl*) lub za pomocą tzw. klienta poczty elektronicznej (odpowiedniego programu lub aplikacji na smartfon). Program pocztowy (np. Microsoft Outlook, Microsoft Outlook Express,

Windows Live Mail, Mozilla Thunderbird) musi zostać zainstalowany na komputerze i odpowiednio skonfigurowany (trzeba wprowadzić parametry konta poczty elektronicznej).

Wysyłanie wiadomości za pomocą poczty elektronicznej obsługuje **protokół sieciowy SMTP** (z ang. *Simple Mail Transfer Protocol*), a odbieranie – protokół **POP3** (z ang. *Post Office Protocol 3*) lub protokół **IMAP** (z ang. *Internet Message Access Protocol*).

P **Protokół sieciowy**

Zbiór reguł, którym podlegają komunikujące się ze sobą komputery.

Ćwiczenie 1. Wyszukujemy zasady działania poczty elektronicznej

Znajdź w Internecie zasady korzystania z poczty elektronicznej. Zapoznaj się z nimi i odpowiedz na pytania:

a. Czym różni się format HTML od zwykłego tekstu?
b. Jak wysłać e-mail do kilkunastu odbiorców, aby żaden z nich nie widział pozostałych adresów?
c. O czym należy pamiętać, gdy korzysta się z opcji „odpowiedz autorowi”?
d. Czy za pomocą poczty elektronicznej można przesłać na przykład film lub wiele zdjęć z aparatu cyfrowego jednocześnie? Odpowiedź uzasadnij.
e. W jaki sposób można wysłać cały folder jako załącznik?

Dostęp poprzez stronę WWW	**Dostęp poprzez klienta poczty elektronicznej**
dostęp do konta uzyskujemy poprzez stronę WWW służącą do obsługi konta, każdorazowo musimy podać swój login i hasło	na komputerze lub telefonie komórkowym musi być zainstalowany i skonfigurowany odpowiedni program
poczta jest przechowywana na serwerze	poczta jest przechowywana na dysku lokalnym komputera lub w pamięci telefonu
bezpośredni dostęp do poczty z dowolnego komputera	dostęp do poczty z komputera lub telefonu, na którym zainstalowano program
wykonywanie operacji na wiadomościach (np. ich przeglądanie, przenoszenie między folderami) może być powolne	wykonywanie operacji na wiadomościach jest szybkie
brak narzędzi dla zaawansowanych użytkowników lub nieliczne narzędzia	liczne narzędzia dla zaawansowanych użytkowników, np. możliwość porządkowania wiadomości według zadanych kryteriów (filtry)

Tabela 2. *Porównanie metod dostępu do poczty elektronicznej*

3. Zasady netykiety



Z wszelkich form komunikacji w Internecie korzystaj w taki sposób, aby odbiorca widział, że potrafisz to robić profesjonalnie – będziesz wtedy traktowany poważnie.

Zasad netykiety nie trzeba uczyć się na pamięć – tylko ich przestrzegać.

We wszystkich formach komunikacji:

- starannie redaguj treść wiadomości – pisz poprawną polszczyzną, bez błędów ortograficznych, interpunkcyjnych czy redakcyjnych; używaj narzędzi sprawdzających pisownię, jeśli tylko są dostępne;
- nie obrażaj odbiorcy (rozmówcy) i zawsze go szanuj; w dyskusjach internetowych możesz zwracać się do rozmówcy, używając nicka lub imienia, jeśli rozmówca wyraża na to zgodę; nie używaj samego nazwiska;
- nie pisz całego tekstu wielkimi literami, nie używaj naprzemiennie wielkich i małych liter;
- do zapisu treści wiadomości używaj polskich liter, z wyjątkiem sytuacji, w której adresat wiadomości znajduje się za granicą;
- format listu (HTML lub zwykły tekst) dostosuj do zasad obowiązujących w danej formie komunikacji, np. na forach i w grupach dyskusyjnych często jest zalecany zwykły tekst;
- emotikonów używaj z umiarem (mają być dodatkiem do tekstu, a nie główną treścią);
- odpowiadaj zawsze pod cytowanym fragmentem; cytuj tylko te fragmenty, do których się odnosisz; nie cytuj całej wiadomości, jeśli nie ma takiej potrzeby;
- nie rozsyłaj reklam i innych informacji do osób, które wcześniej nie wyraziły na to zgody; nie rozsyłaj „internetowych łańcuszków" (jest to forma spamu).

Korzystając z poczty elektronicznej:

- nadawaj sensowny temat wiadomościom; nie umieszczaj w polu tematu treści wiadomości;
- zawsze rozważ, czy wysyłając e-mail jednocześnie do wielu osób, wybrać opcję z jawnymi adresami poczty elektronicznej, czy zastosować kopię ukrytą.

W grupach i na forach dyskusyjnych:

- odpowiedzi na proste pytania szukaj najpierw w wyszukiwarkach internetowych – grupy i fora dyskusyjne to miejsce na dyskusje i wymianę poglądów;
- przed wejściem do danej grupy dyskusyjnej zapoznaj się z jej regulaminem;
- zanim zaczniesz aktywnie uczestniczyć w danej grupie, najpierw poćwicz wysyłanie wiadomości, korzystając z grupy testowej;
- przed zadaniem pytania zapoznaj się z najczęściej zadawanymi pytaniami (wykazem FAQ); przejrzyj archiwum grupy;
- jeśli chcesz rozpocząć dyskusję na konkretny temat, utwórz nowy wątek; w przeciwnym przypadku dołącz do istniejącego wątku;
- pisz zawsze na temat omawiany w danym wątku forum, a do grupy wysyłaj wiadomości związane z jej tematem.

Na blogu:

- dyskutuj rozważnie; możesz również kontynuować przemyślenia autora, przekazując własne spostrzeżenia;
- jeśli krytykujesz, rób to konstruktywnie – nie krytykuj autora blogu, tylko wpisy.

W serwisach społecznościowych:

- nie publikuj zdjęć bez zgody znajdujących się na nich osób, najlepiej skontaktuj się wcześniej z danymi osobami i poczyń stosowne ustalenia;
- nie umieszczaj prywatnych informacji o innych osobach bez ich zgody;
- nie umieszczaj w profilach innych osób obraźliwych komentarzy i wystrzegaj się niepoważnych treści i niemądrych komentarzy w swoim profilu.

4. Chmura obliczeniowa

Internet umożliwia przechowywanie oraz **przetwarzanie danych** (w tym wykonywanie obliczeń) w chmurze obliczeniowej, a dokładnie – poprzez **serwery internetowe** firm oferujących taką usługę.

S **Serwer internetowy**

Komputer, który świadczy usługi w Internecie, np. przechowuje i udostępnia strony internetowe WWW.



Aby zacząć korzystać z chmury obliczeniowej, należy założyć konto w wybranej usłudze internetowej (trzeba podać m.in. login i hasło), która udostępnia przestrzeń dyskową na swoich serwerach internetowych.

Uzyskanie przestrzeni w chmurze obliczeniowej umożliwiają rozwiązania takie, jak: Microsoft OneDrive, Dropbox, Dysk Google.

Operacje na danych (tj. umieszczanie w chmurze, przeglądanie i pobieranie z chmury) można wykonywać za pomocą przeglądarki internetowej lub specjalnej aplikacji, którą należy zainstalować na komputerze.

Z danych zapisanych w chmurze można korzystać na wielu komputerach lub urządzeniach mobilnych. Oprogramowanie dba o to, aby dane na wszystkich komputerach były identyczne (proces ten nazywa się **synchronizacją danych**).

Możliwości wynikające z korzystania z chmury obliczeniowej:

- umieszczanie w chmurze plików: zdjęć, obrazów i innych dokumentów komputerowych;
- dostęp do umieszczonych danych z dowolnego komputera lub urządzenia mobilnego;
- zapraszanie innego użytkownika do korzystania z naszych danych; można m.in.:
  - udostępnić plik;
  - umożliwić współdzielenie danego folderu (co oznacza, że inny użytkownik może nie tylko przeglądać folder, pobierać pliki, lecz także umieszczać swoje pliki w tym folderze);
  - udostępnić link do wybranego folderu.

Dane umieszczane w chmurze obliczeniowej można traktować jako kopię zapasową naszych dokumentów.

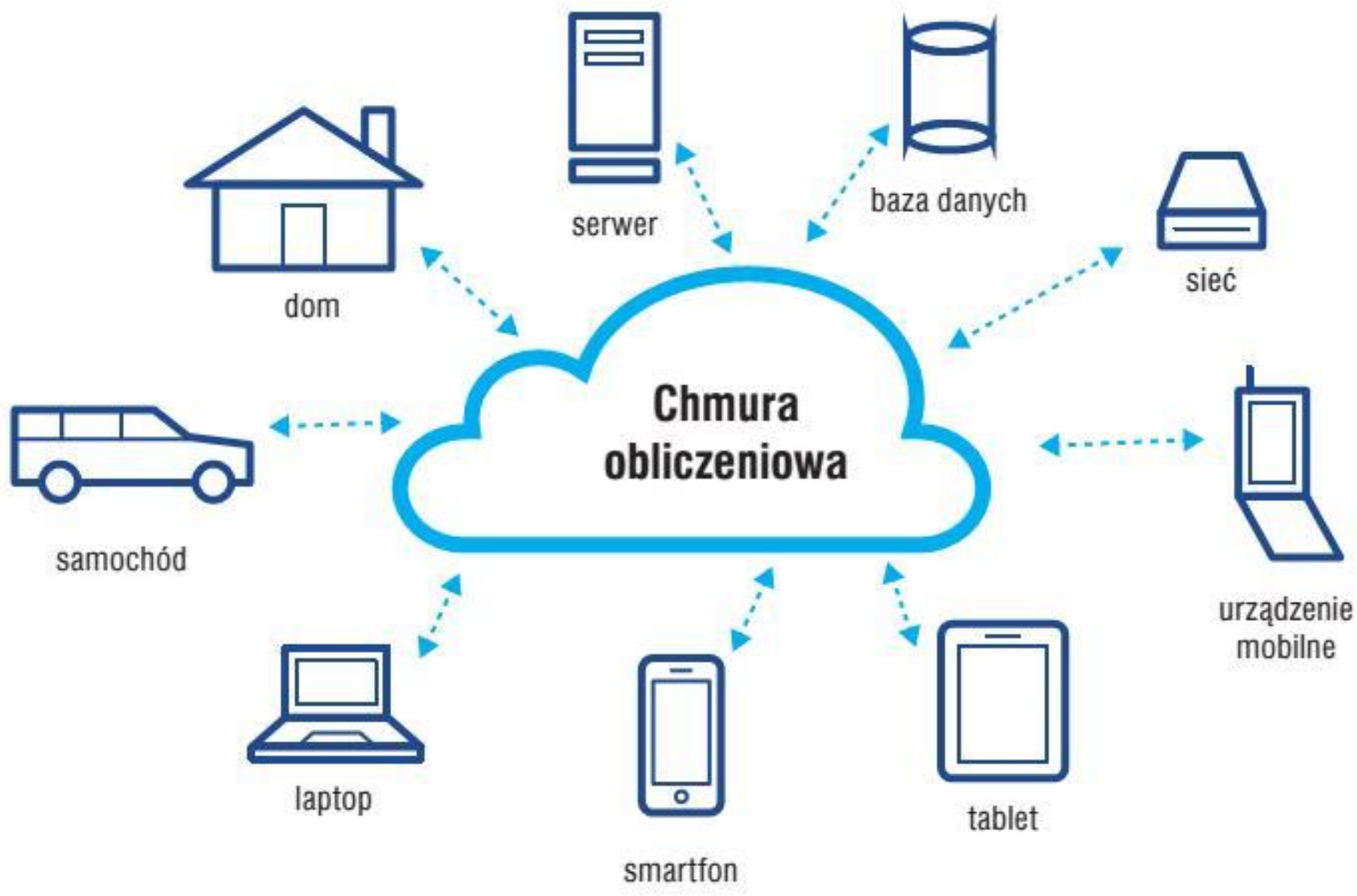
Korzystanie z chmury w najprostszym zakresie jest zwykle bezpłatne. Dane możemy umieszczać i pobierać za darmo, ale udostępniana jest określona ilość miejsca na dane – zwykle za powiększenie tego obszaru należy zapłacić. Warto wcześniej zapoznać się z warunkami oferowanymi przez daną firmę.

Poza umieszczaniem plików w chmurze możemy również korzystać z niektórych programów za pomocą przeglądarki internetowej. Programy nie są uruchamiane na komputerze użytkownika, ale przechowywane na serwerze internetowym i dostępne przez Internet. 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. Dokumenty tworzone w takich aplikacjach są umieszczane w chmurze (można je pobrać na komputer w postaci plików). Możemy też dokumenty współdzielić z innymi użytkownikami. Zmiany wprowadzane w dokumentach są natychmiast widoczne dla współużytkowników, z którymi je dzielimy.

W programach pakietu Microsoft Office (np. w wersji 2013 i nowszych) istnieje możliwość współtworzenia (współdzielenia) dokumentów przez wielu użytkowników pracujących na różnych komputerach w różnych miejscach.

Z chmurą obliczeniową mogą być połączone różne urządzenia znajdujące się w domach lub samochodach. Np. zainstalowany w domu system bezpieczeństwa może przesyłać zapisy z kamer do chmury. Domowi asystenci (Amazon Echo, Google Home) to sterowane głosem urządzenia, które pomagają wykonywać ludziom różne czynności. Urządzenie zainstalowane w samochodzie może przesyłać do chmury pozyskane za pomocą systemu GPS informacje o położeniu samochodu, prędkości jazdy itp.



Rys. 2. *Schematyczny rysunek chmury obliczeniowej*

Ćwiczenie 2. Zapoznajemy się z możliwościami chmury

Zapoznaj się z możliwościami wybranej usługi umożliwiającej przechowywanie danych w chmurze oraz korzystanie z oprogramowania w chmurze.

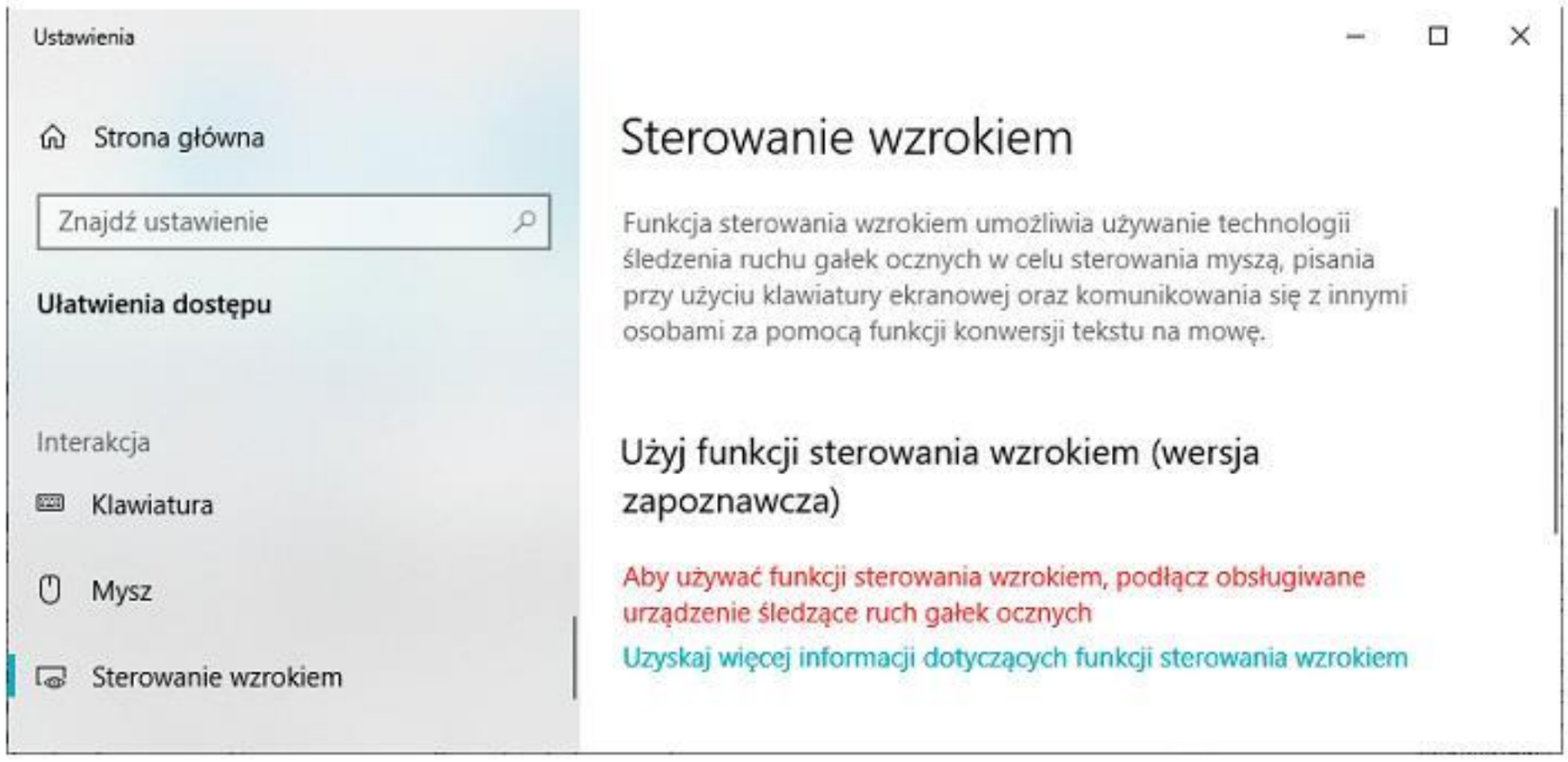
5. Korzyści i zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnych

Technologia informacyjno-komunikacyjna, czyli połączenie zastosowań informatyki z technikami komunikacji, wkroczyła w prawie każdą dziedzinę działalności człowieka (m.in. usługi, pracę, naukę, kulturę i rozrywkę), umożliwiając ich dynamiczny rozwój.

Dzięki technologii informacyjno-komunikacyjnej ułatwiono wyszukiwanie, przechowywanie, przetwarzanie i przesyłanie informacji, a także nieograniczone czasem i przestrzenią komunikowanie się między ludźmi.

Komunikacja i wymiana informacji z wykorzystaniem Internetu przynoszą wiele korzyści, zwłaszcza ludziom o specjalnych potrzebach (np. z ograniczonymi możliwościami poruszania się, chorym, z niepełnosprawnością). Są to różne rozwiązania techniczne, m.in.:

- specjalna sekcja w ustawieniach systemu Windows (dostępnych w aplikacji Ustawienia lub w Panelu sterowania), pozwalająca na dostosowanie pracy komputera do potrzeb niepełnosprawnych, np. powiększanie czcionki dla słabowidzących,
- syntezatory mowy, umożliwiające odczytywanie treści dokumentów lub stron internetowych niewidomym,
- generatory tekstu, umożliwiające niesłyszącym zobaczenie treści komunikatów dźwiękowych (np. serwis YouTube pozwala na włączenie napisów),
- śledzenie ruchu oka osób sparaliżowanych (rys. 3.).



Rys. 3. *Okno aplikacji* ***Ustawienia systemu Windows*** *(Windows 10)*

Ćwiczenie 3. Sprawdzamy ustawienia systemu Windows dla osób o specjalnych potrzebach

Sprawdź, jakie są możliwości dostosowywania dla osób niedowidzących używanej przez ciebie wersji systemu Windows. Jakie inne dostosowania dla osób o specjalnych potrzebach oferuje system?

Wskazówka: Aby otworzyć okno, w którym możemy dostosowywać ustawienia systemu, należy z menu **Start** uruchomić aplikację **Ustawienia** i w otwartym oknie **Ustawienia systemu Windows** wybrać opcję **Ułatwienia dostępu**.

Korzystając z technologii informacyjno-komunikacyjnej, trzeba zdawać sobie sprawę z zagrożeń, z jakimi możemy się spotkać.

Uzależnienie od komputera i Internetu

Wiele godzin spędzonych na przeglądaniu stron internetowych czy komunikowaniu się przez Internet może źle wpływać na zachowania społeczne, psychikę oraz zdrowie fizyczne (m.in. może powodować wady wzroku, zwyrodnienie kręgosłupa).

Osoby uzależnione od korzystania z telefonu komórkowego lub Internetu często nie potrafią zorganizować sobie czasu. Brak dostępu do Internetu wywołuje u nich zniechęcenie, czasem załamanie nerwowe, które może doprowadzić do depresji. Szczególnie niebezpieczne jest stopniowe zanikanie granic pomiędzy rzeczywistością a fikcją. Negatywne wzorce zachowań są często przenoszone ze świata wirtualnego do rzeczywistości. Na przykład uzależnienie od brutalnych gier komputerowych może pobudzać do agresywnych zachowań lub doprowadzić do braku wrażliwości na ludzką krzywdę.

Należy pamiętać o równoważeniu czasu spędzanego przy komputerze z czasem spędzanym na wykonywaniu innych zajęć takich, jak: spotkania z przyjaciółmi, przebywanie na świeżym powietrzu, czynna rekreacja, np. uprawianie sportu, spacery, wycieczki.

Zanik związków międzyludzkich

Korzystanie z komputera oraz z Internetu ogranicza bezpośrednie kontakty międzyludzkie.

Dzięki nowoczesnym technologiom zamiast spotykać się ze znajomymi, rozmawiamy z nimi przez Internet. Zaletą takiego sposobu komunikowania się jest możliwość poznawania mieszkańców najodleglejszych zakątków świata oraz prowadzenia równoczesnej rozmowy z kilkoma osobami przebywającymi w różnych miejscach. Wadą jest natomiast odizolowanie się od rzeczywistego świata, problemy w nawiązywaniu bezpośrednich kontaktów.

Anonimowość kontaktów

W Internecie łatwo udawać kogoś innego. Nie można mieć pewności, że osoba, z którą korespondujemy lub rozmawiamy na czacie, jest tą, za którą się podaje. Dotyczy to szczególnie umawiania się na spotkania z ludźmi poznanymi w Internecie.

Mimo że w prosty sposób można nadać sobie dowolną liczbę pseudonimów czy założyć kilka kont pocztowych, nie zwalnia to nikogo z właściwego zachowania. Niestety, zbyt często podczas korzystania z różnych form komunikacji internetowej jesteśmy atakowani wypowiedziami agresywnymi, wulgarnymi czy uderzającymi w naszą godność.

Należy jednak pamiętać, że działania w Internecie zostawiają ślad. Można zidentyfikować nadawcę wiadomości, nawet jeśli nie podpisuje się imieniem i nazwiskiem.

Zagrożenia prywatności

Wielu użytkowników portali społecznościowych czy komunikatorów chętnie dzieli się prywatnymi informacjami o sobie, nie zastanawiając się nad konsekwencjami.

Jeśli nie jest to konieczne, nie należy podawać wszystkich danych o sobie – w szczególności nie należy ujawniać swojego nazwiska, adresu zamieszkania i numeru telefonu. Na podstawie wybranych danych obca osoba może np. łatwo odgadnąć hasło do naszego konta pocztowego (niektóre osoby w haśle umieszczają datę urodzenia czy numer telefonu).

Nieroztropne jest umieszczanie w Internecie zbyt wielu zdjęć z życia prywatnego, na których pokazujemy np. uroczystości rodzinne, rozmieszczenie pomieszczeń w domu i ich wyposażenie, co może być (w połączeniu z podanym adresem) cenną wskazówką dla oszustów i złodziei. Takich przykładów można podawać wiele.

Bardzo często nasza nieuwaga i beztroska stanowią źródło zagrożenia, należy więc zachować rozsądek w kwestii umieszczania prywatnych danych w Internecie.

Cyberprzemoc

W Internecie można spotkać się z cyberprzemocą (ang. *cyberbullying*), która polega na nękaniu, straszeniu, wyśmiewaniu się z innych osób, podszywaniu się pod kogoś w Sieci.

Do tych działań wykorzystywane są m.in.: poczta elektroniczna, czaty, komunikatory, strony internetowe, blogi, serwisy społecznościowe i telefony komórkowe. Napastnicy, czując się bezkarni i anonimowi, prześladują swoje ofiary złośliwymi i obraźliwymi SMS-ami, telefonami lub e-mailami, mogą rozsyłać również niechciane zdjęcia lub filmy.

Osoby, które publikują w Internecie agresywne, obraźliwe lub skrajnie krytyczne komentarze pod adresem innej osoby (osób), nazywamy **hejterami**, a ich działanie **hejtem** (z ang. *hate* – nienawidzić). Stosowanie hejtu (hejtowanie) podlega odpowiedzialności – od dyscyplinarnej (kary szkolne), poprzez cywilną (zadośćuczynienie za znieważenie) do karnej.

Niewiarygodność informacji

W Internecie znajdziemy zarówno informacje rzetelne, profesjonalnie opracowane, jak i niewiarygodne, zawierające błędy. Nie należy ufać bezkrytycznie wszystkim informacjom wyszukanym w Internecie, m.in. umieszczonym na prywatnych stronach internetowych. Ostrożnie trzeba też traktować informacje zamieszczane na forach

i w grupach dyskusyjnych. Należy pamiętać, że są to przede wszystkim subiektywne opinie, które nie zawsze muszą być zgodne z prawdą.

Rzetelnym źródłem informacji są profesjonalnie opracowane portale edukacyjne, internetowe encyklopedie i słowniki. Jeśli są nam potrzebne ważne informacje prawne, wybierajmy fachowe strony poświęcone prawu, np. rządowe, a nie amatorskie opracowania przepisów.

Nieprawdziwe informacje, tworzone i rozpowszechnianie specjalnie, noszą ogólną nazwę **fake news** (ang. *fake* – fałszywy, nieprawdziwy, *news* – wiadomości).

Od wielu lat dostępne są narzędzia do tworzenia realistycznych fotomontaży, także filmowych. Specjalny program umożliwia zastąpienie twarzy wybranej osoby wizerunkiem kogoś innego. Zachowane zostają przy tym wszystkie elementy oryginalnego filmu (kąt nachylenia twarzy, mimika, oświetlenie itp.). Można również naśladować czyjś głos, wypowiadając dowolne zdania. W ten sposób można np. stworzyć nagranie polityka wygłaszającego tezy nie jego autorstwa.

Taki fotomontaż filmowy nazywa się *deepfake* (z ang. *deep* – głęboki, *fake* – fałszywy, nieprawdziwy).

Warto zapamiętać

- Internet umożliwia różne sposoby komunikowania się i wymiany informacji, m. in. poprzez: pocztę elektroniczną, czat, komunikatory internetowe, grupy i fora dyskusyjne, portale społecznościowe, blogi.
- Przesyłanie plików w Internecie umożliwia m.in. protokół FTP.
- Do wymiany plików pomiędzy komputerami bez pośrednictwa serwera wykorzystuje się model komunikacji P2P.
- Korzystając z różnych form internetowej komunikacji, należy przestrzegać zasad netykiety.
- Internet umożliwił korzystanie z chmury obliczeniowej: przechowywanie danych w chmurze i łatwy do nich dostęp z dowolnego komputera lub urządzenia mobilnego, korzystanie z oprogramowania, które jest dostępne w Internecie, bez instalowania go na swoim komputerze, a także współtworzenie z innymi użytkownikami dokumentów komputerowych.
- Rozwój technologii informacyjno-komunikacyjnych przyczynia się do postępu w wielu dziedzinach działalności człowieka, ale musimy być równocześnie świadomi zagrożeń, z jakimi możemy się spotkać.

Pytania i polecenia

1. Jakie wydarzenie w historii rozwoju usług internetowych stanowiło ważny przełom w rozwoju komunikacji między ludźmi?
2. Jakie trzy zasady korzystania z poczty elektronicznej warto podać osobie, która właśnie założyła konto i chce wysłać pierwszy e-mail?
3. Omów schemat działania poczty elektronicznej.
4. Czy zgadzasz się z twierdzeniem, że w komunikacji internetowej panuje nieograniczona swoboda? Uzasadnij odpowiedź.

5. Jakie są różnice pomiędzy następującymi formami internetowej komunikacji: czat, komunikator internetowy i forum dyskusyjne?
6. Na czym polega model komunikacji P2P? Co umożliwia protokół FTP?
7. Dlaczego niektóre osoby wolą rozmawiać z innymi w rzeczywistości wirtualnej? Podaj zalety i wady takiego sposobu kontaktu między ludźmi.
8. Przedstaw rolę Internetu, w tym różnych form komunikacji, w życiu osób niepełnosprawnych lub samotnych.
9. Omów zasady netykiety, które powinno się stosować we wszystkich formach komunikacji.
10. Jakie zasady obowiązują na blogu, w portalach społecznościowych, w grupach i na forach dyskusyjnych oraz podczas wysyłania e-maili?
11. Wyjaśnij, czym jest chmura obliczeniowa i do czego może być wykorzystana.
12. Jakie możliwości daje umieszczanie danych w chmurze obliczeniowej?
13. Omów zagrożenia, z jakimi możemy się spotkać, korzystając z technologii informacyjno-komunikacyjnej. Przed jakimi zagrożeniami należy szczególnie chronić dzieci?
14. Jak sobie wyobrażasz przyszłość poszczególnych form komunikowania się z wykorzystaniem Internetu? Które z nich się rozwiną, a które ulegną zapomnieniu?



Zadania

1. Przedstaw w tabeli podobieństwa i różnice między listem tradycyjnym a elektronicznym.
2. Rozważ problem: *Gdyby Adam Mickiewicz żył w erze Internetu, czy wysyłałby listy elektroniczne zamiast odręcznej korespondencji*? Punktem wyjścia do dyskusji może być trwałość listów tradycyjnych i elektronicznych.
3. Podaj argumenty za lub przeciw tezie *Listy elektroniczne mogą całkowicie zastąpić listy tradycyjne* w odniesieniu do różnych dziedzin życia. Uzasadnij swoje zdanie.

Dla zainteresowanych

4. Wyszukiwarka Google umożliwia tworzenie grup dyskusyjnych i zapisywanie się do istniejących grup. Zapoznaj się z zasadami korzystania z grup dyskusyjnych Google.
5. Znajdź w Internecie przykłady filmów, w których zastosowano technikę *deepfake*.

Temat D3. Tworzenie stron WWW z wykorzystaniem stylów i elementów programowania

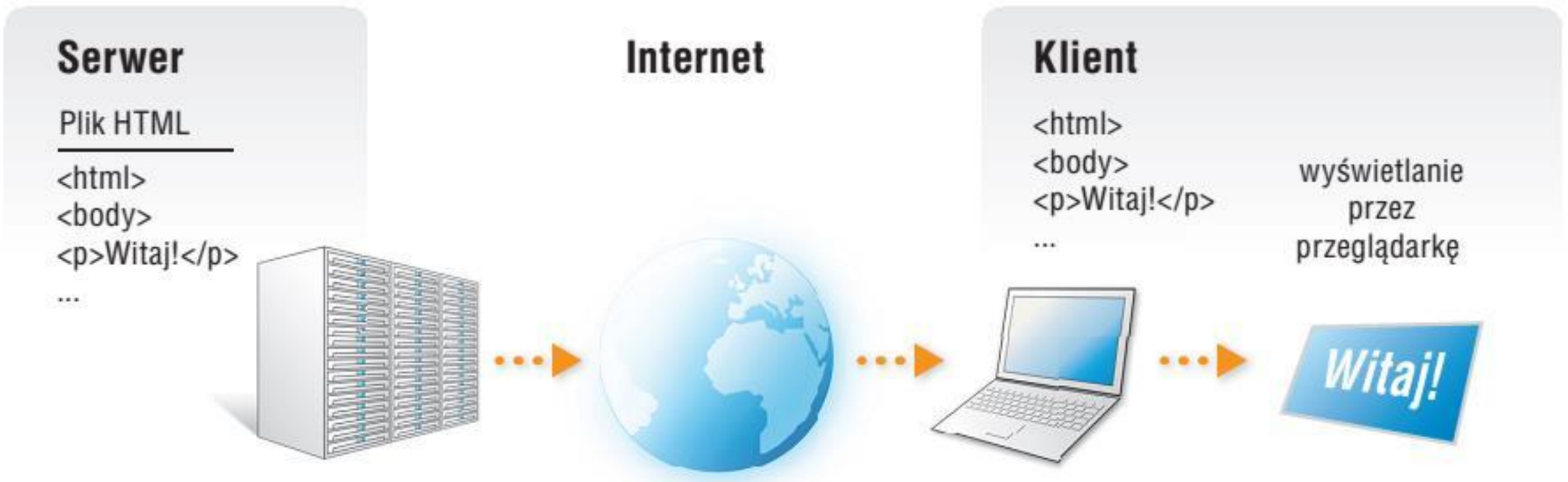
1. Jak zapisana jest strona WWW?
2. Przykład tworzenia strony w języku HTML
   - 2.1. Stosowanie podstawowych znaczników języka HTML
   - 2.2. Tworzenie łączy hipertekstowych
   - 2.3. Tworzenie podstrony
   - 2.4. Wstawianie obrazów
   - 2.5. Wstawianie tabel
   - 2.6. Stosowanie kolorów
3. Kaskadowe arkusze stylów CSS
   - 3.1. Najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości
   - 3.2. Formatowanie hiperłączy
   - 3.3. Tło strony
   - 3.4. Klasy elementów
4. Elementy dynamiczne na stronie – skrypty w języku JavaScript
5. Publikowanie i promowanie stron internetowych w Internecie
   - 5.1. Publikowanie strony WWW
   - 5.2. Promowanie strony WWW

Warto powtórzyć

1. Przypomnij pojęcia: strona WWW, witryna internetowa, adres WWW.
2. Czym są hiperłącza?
3. Jakie znasz znaczniki języka HTML?
4. W jaki sposób tworzyliśmy i jak wykorzystywaliśmy style w edytorze tekstu?

1. Jak zapisana jest strona WWW?

Do opisu stron WWW stosuje się specjalny język – **HTML** (ang. *Hypertext Markup Language*, czyli język znaczników hipertekstowych). Serwer WWW przesyła do przeglądarki internetowej kod HTML. Przeglądarka interpretuje otrzymany kod i wyświetla stronę (rys. 1.).



Rys. 1. *Schemat działania statycznej strony WWW*

Plik w języku HTML to plik tekstowy, zapisany z rozszerzeniem *htm* bądź *html*. Jeden plik w języku HTML opisuje jedną stronę **WWW**, natomiast **witryna internetowa** składa się najczęściej z wielu (od kilkudziesięciu do nawet kilkuset tysięcy) stron, wraz z dodatkowymi danymi: obrazami, animacjami, dźwiękami itp.

Oprócz języka HTML do opisywania wyglądu i funkcjonowania witryn internetowych wykorzystuje się **kaskadowe arkusze stylów** (CSS – ang. *Cascading Style Sheet*) oraz skryptowy język programowania **JavaScript**. Niektóre strony internetowe posiadają też dodatkowe (coraz rzadziej spotykane) elementy (np. odtwarzacz multimediów, interaktywne gry), wykonane w technologiach Java lub Adobe Flash.

Do utworzenia prostej witryny wystarczą: edytor plików tekstowych lub edytor tekstu, edytor grafiki oraz program umożliwiający przesyłanie na serwer WWW (klient FTP) plików składających się na witrynę. Rozbudowane witryny tworzy się zwykle z wykorzystaniem specjalistycznych narzędzi. Witryny takie korzystają również z baz danych i języków skryptowych.

Przykładowe narzędzia	Opis, przeznaczenie
edytory plików tekstowych	wymagają znajomości znaczników HTML, m.in.: • dostępny w systemie Windows program Notatnik; • dostępne bezpłatnie w Internecie (np. Notepad++, Programmer's, Notepad, PSPad, BlueFish), wyróżniają (różnymi kolorami) znaczniki i inne elementy języka HTML
edytor tekstu Microsoft Word	tworzenie strony bez znajomości języka HTML (w oknie edytora widzimy stronę tak, jak będzie wyglądać w przeglądarce); umożliwia zapis dokumentu w formacie HTML (opcja **Zapisz jako/Strona sieci Web**); zawiera narzędzia pozwalające na przygotowanie strony WWW na podstawie szablonu lub istniejącego dokumentu (udostępnia na pasku narzędzi specjalne polecenia, które automatycznie tworzą odpowiednie znaczniki HTML)

systemy zarządzania treścią (CMS)	pozwalają na utworzenie witryny internetowej i zarządzanie nią za pomocą przeglądarki internetowej, bez konieczności poznawania języka HTML; nie wymagają szczegółowej wiedzy dotyczącej technologii internetowych; rozpowszechniane na licencji GNU GPL, Joomla!, Drupal, WordPress
serwisy przeznaczone do tworzenia stron	rozbudowane serwisy, pozwalające tworzyć profesjonalne strony
specjalistyczne narzędzia	przeznaczone dla osób zajmujących się profesjonalnie tworzeniem stron internetowych; podczas tworzenia witryny wymagana jest zazwyczaj znajomość baz danych i skryptowych języków programowania

Tabela 1. *Wybrane narzędzia do tworzenia stron*

Etapy wykonywania strony

1. Określenie tematyki strony internetowej.
2. Planowanie, projektowanie:
   - treści, w tym elementów grafiki (rysunków, zdjęć), które chcemy zamieścić na stronie,
   - ustalenie sposobu rozmieszczenia grafiki względem tekstu,
   - ustalenie kolorystyki strony (tła, tekstu, elementów graficznych),
   - parametrów formatowania tekstu,
   - numeracji, wyliczeń, tabel,
   - systemu nawigacji (wstawienie hiperłączy do innych stron, w tym podstron).
3. Wykonanie strony – utworzenie kodu w języku HTML w oparciu o projekt.
4. Sprawdzenie poprawności tekstów i funkcjonowania strony.

Ćwiczenie 1. Porównujemy wygląd witryn

Porównaj kilka najczęściej odwiedzanych przez ciebie witryn. Zwróć uwagę na strukturę strony: tytuł, rozmieszczenie tekstu, przejrzystość, grafikę, sposób przechodzenia do innych stron.

Ćwiczenie 2. Oglądamy źródło strony

Wejdź na wybraną stronę internetową, np. swojej szkoły. Obejrzyj źródło strony, korzystając z odpowiedniej opcji przeglądarki (dostępnej zwykle w menu kontekstowym strony).

2. Przykład tworzenia strony w języku HTML

Zamierzamy utworzyć stronę internetową o Wyspach Kanaryjskich. Na stronie chcemy umieścić fotografie i informacje o najciekawszych zabytkach lub ważnych obiektach, wskazówki dla turystów dotyczące godzin otwarcia muzeów oraz opłat za zwiedzanie, ciekawostki historyczne, przyrodnicze itp. Jak wykonać taką stronę, korzystając z edytora plików tekstowych?

2.1. Stosowanie podstawowych znaczników języka HTML

Aby utworzyć stronę w języku HTML, należy zapisać jej kod w pliku z rozszerzeniem *htm* lub *html*. Taki plik możemy następnie otworzyć w przeglądarce internetowej, aby zobaczyć efekt naszej pracy.

Do utworzenia przykładowej strony internetowej o Wyspach Kanaryjskich wykorzystamy program Notepad++. Aby znaczniki HTML były wyróżnione kolorem, należy w menu **Składnia** wybrać z listy wyboru literę **H**, a następnie **HTML**.

C:\Strona\index.html - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Okno ? X

index.html

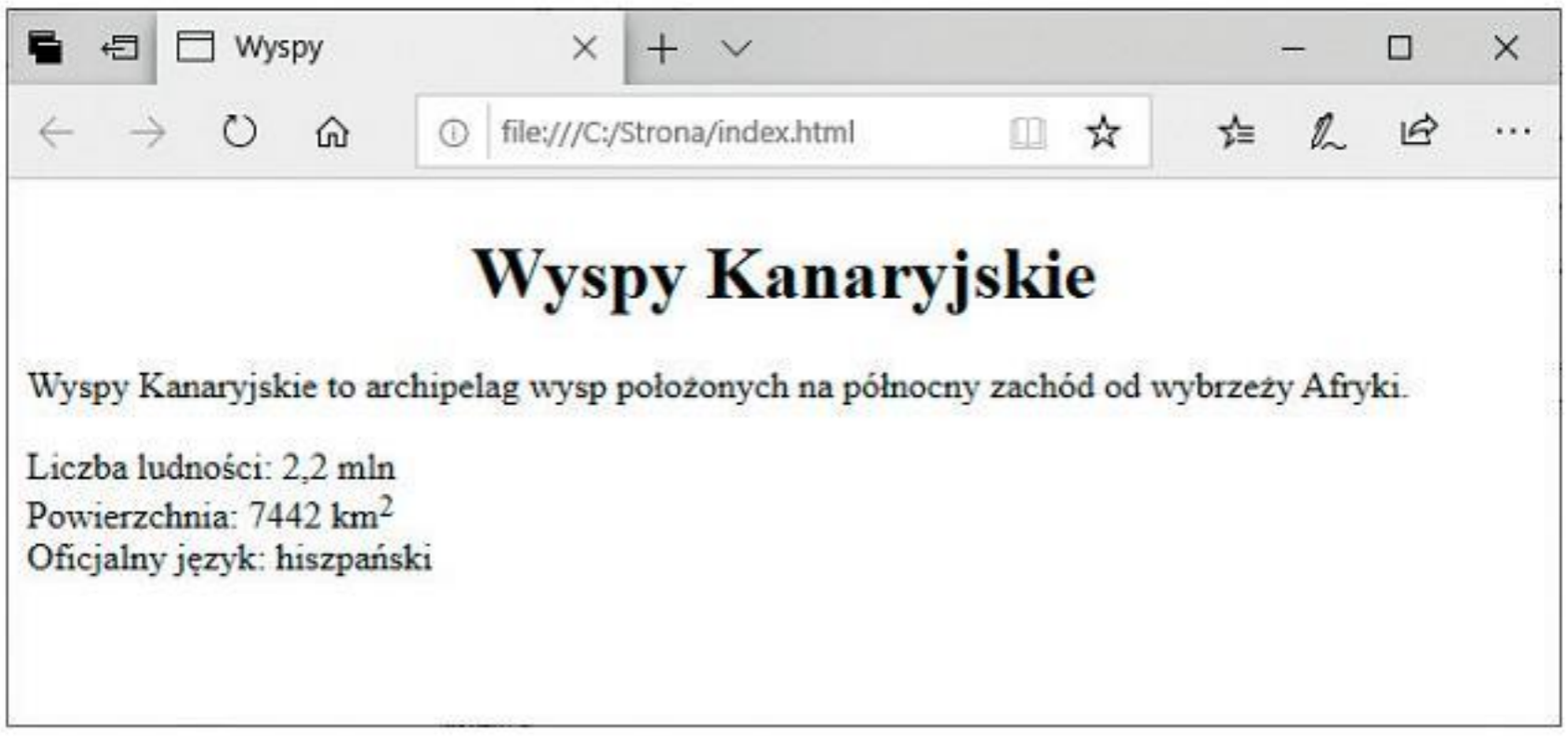
```
1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Wyspy</title>
6      </head>
7
8      <body>
9      <h1 align="center">Wyspy Kanaryjskie</h1>
10     <p>Wyspy Kanaryjskie to archipelag wysp położonych na północny zachód
11     od wybrzeży Afryki.</p>
12     <p></p>
13     <p>Liczba ludności: 2,2 mln<br>
14     Powierzchnia: 7442 km<sup>2</sup><br>
15     Oficjalny język: hiszpański</p>
16     </body>
17
18 </html>
19
20
21
```

length : 400 lines : 21 | Ln : 1 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

informacja dla przeglądarki o standardzie, w jakim wykonano dokument – dla standardu HTML5 stosuje się `<!DOCTYPE html>`

informacja o kodowaniu znaków w standardzie kodowania UTF-8

Rys. 2a. *Kod źródłowy strony widoczny w oknie programu Notepad++ – ćwiczenie 3.*



Rys. 2b. *Strona widoczna w przeglądarce – ćwiczenie 3.*

Tworząc akapit, można określić sposób jego wyrównania (podobnie jak w edytorze tekstu: do lewej, do środka, do prawej, wyjustowany). Służy do tego atrybut `align` (z ang. *wyrównywać*), przyjmujący jedną z wartości wymienionych w tabeli 2. Atrybut `align` jest wykorzystywany także w innych znacznikach języka HTML.

Postać atrybutu `align`	Funkcja
`align="left"`	wyrównanie do lewej
`align="center"`	wyrównanie do środka
`align="right"`	wyrównanie do prawej
`align="justify"`	justowanie

Tabela 2. *Wartości atrybutu* `align`



Przykład 1. Stosowanie atrybutu `align`

```
<p align="center">Wyrównanie akapitu do środka strony</p>
```

W języku HTML mamy do dyspozycji sześć poziomów tytułów, w kolejności od najważniejszych do najmniej ważnych.

```
<h1>Tytuł pierwszego poziomu</h1>
…
<h6>Tytuł szóstego poziomu</h6>
```

`<h1 align="center">Tytuł</h1>` spowoduje wyśrodkowanie tytułu względem szerokości strony.

Uwagi:

- Domyślnie (jeżeli nie użyjemy atrybutu `align`) akapity i tytuły wyrównywane są do lewej strony.
- W języku HTML wartości atrybutów wpisuje się w znakach cudzysłowu.

Jeżeli strona ma zostać umieszczona w Internecie, należy pamiętać o następujących zasadach nazywania plików:

- nie używamy w nazwach plików polskich znaków diakrytycznych,
- nie używamy w nazwach plików spacji,
- używamy tylko małych liter (większość serwerów WWW oparta jest na systemie operacyjnym Linux, który w nazwach plików rozróżnia małe i wielkie litery).

Ćwiczenie 3. Zapisujemy kod źródłowy strony w programie Notepad++

1. Utwórz folder o nazwie *Strona*, w którym zapiszesz swoją stronę WWW.
2. Uruchom program Notepad++ i przepisz kod strony podany na rysunku 2a.
3. Wyjaśnij znaczenie znaczników umieszczonych w kodzie, wykorzystując wiadomości z lekcji informatyki ze szkoły podstawowej.
3. Zapisz kod strony w pliku o nazwie *index.html* i otwórz go w przeglądarce internetowej.

Uwaga

Strona o Wyspach Kanaryjskich jest tylko przykładem – możesz na jej wzór od razu wykonać podobną stronę na wybrany przez siebie temat.

Wskazówki:

- Aby w programie Notepad++ zastosować kodowanie UTF-8, należy z menu **Format** wybrać opcję **Koduj w UTF-8**.
- Przy zapisywaniu pliku w programie Notepad++ w polu **Zapisz jako typ** należy wybrać opcję **All types** (**Wszystkie pliki**).

Czasami zachodzi potrzeba rozdzielenia fragmentów tekstu poziomą linią. Do utworzenia takiej linii służy znacznik `<hr>`. Znacznik ten posiada także dodatkowe atrybuty.

Nazwa atrybutu	Funkcja
`align="`*`wyrównanie`*`"`	Określa położenie linii względem strony. Możliwe wartości: `left`, `right`, `center`.
`size="`*`grubość`*`"`	Określa grubość linii (w pikselach).
`width="`*`szerokość`*`"`	Określa szerokość linii. Szerokość może być określona w pikselach lub procentowo, względem szerokości elementu nadrzędnego (np. całej strony).

Tabela 3. *Atrybuty znacznika* `<hr>`

HTML pozwala na tworzenie list uporządkowanych (numerowania) lub nieuporządkowanych (wypunktowania). Przy tworzeniu list numerowanych komputer wstawia automatycznie kolejne numery punktów. Listy można też zagnieżdżać wewnątrz siebie.

Lista uporządkowana ma postać:

```
<ol>
  <li>Element listy</li>
  <li>Element listy</li>
  ...
</ol>
```

Lista nieuporządkowana ma postać:

```
<ul>
  <li>Element listy</li>
  <li>Element listy</li>
  ...
</ul>
```

C:\Strona\index.html - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Okno ? X

index.html index.html

```
1   <!DOCTYPE HTML>
2   <html>
3       <head>
4           <meta charset="utf-8">
5           <title>Wyspy</title>
6       </head>
7
8       <body>
9           <h1 align="center">Wyspy Kanaryjskie</h1>
10          <p>Wyspy Kanaryjskie to archipelag wysp położonych na północny zachód
11          od wybrzeży Afryki.</p>
12          <p></p>
13          <ul>
14              <li><b>Liczba ludności:</b> 1,99 mln</li>
15              <li><b>Powierzchnia:</b> 7442 km<sup>2</sup></li>
16              <li><b>Oficjalny język:</b> hiszpański</li>
17          </ul>
18          <p></p>
19          <hr width="80%">
20          <h2>Wyspy wchodzące w skład archipelagu:</h2>
21          <ol>
22              <li>Lanzarote</li>
23              <li>Fuerteventura</li>
24              <li>Gran Canaria</li>
25              <li>Teneryfa</li>
26              <li>El Hierro</li>
27              <li>La Palma</li>
28              <li>La Gomera</li>
29          </ol>
30      </body>
31
32  </html>
33
```

Rys. 3a. *Kod źródłowy strony widoczny w oknie programu Notepad++ – ćwiczenie 4.*

Wyspy

file:///C:/Strona/index.html

Wyspy Kanaryjskie

Wyspy Kanaryjskie to archipelag wysp położonych na północny zachód od wybrzeży Afryki.

- **Liczba ludności:** 1,99 mln
- **Powierzchnia:** 7442 km^2
- **Oficjalny język:** hiszpański

Wyspy wchodzące w skład archipelagu:

1. Lanzarote
2. Fuerteventura
3. Gran Canaria
4. Teneryfa
5. El Hierro
6. La Palma
7. La Gomera

Rys. 3b. *Strona widoczna w przeglądarce – ćwiczenie 4.*

Ćwiczenie 4. Dodajemy do strony listę nieuporządkowaną, listę uporządkowaną i poziomą linię

1. Otwórz plik *index.html*, zapisany w ćwiczeniu 3. w folderze *Strona*.
   Dodaj do strony o Wyspach Kanaryjskich:
   – listę nieuporządkowaną, zawierającą dane o liczbie ludności, powierzchni i języku oficjalnym (urzędowym),
   – listę uporządkowaną, zawierającą nazwy wysp wchodzących w skład archipelagu.
   Listy oddziel od siebie linią poziomą o szerokości 80% strony.
2. Zapisz plik pod tą samą nazwą.

2.2. Tworzenie łączy hipertekstowych

Najważniejszą cechą języka HTML, decydującą o jego popularności i wszechstronności, jest możliwość tworzenia **hiperłączy**. Przez hiperłącze rozumiemy wyróżniony w pewien sposób obiekt (fragment tekstu, rysunek), po którego wybraniu (np. przez kliknięcie myszą) w oknie przeglądarki zostanie wyświetlony nowy dokument.

Do tworzenia łącz hipertekstowych służy znacznik `<a>`. Istnieją dwie różne składnie tego polecenia, o odmiennych zastosowaniach.

Tworzenie hiperłącza do innego dokumentu

Znacznik przyjmuje postać:

```
<a href="obiekt_docelowy">Treść hiperłącza</a>
```

Przez `treść hiperłącza` należy rozumieć tekst lub rysunek aktywujący hiperłącze (po kliknięciu myszą). Natomiast *`obiekt_docelowy`* to nazwa bądź adres URL pliku, który ma zostać otwarty przez przeglądarkę.

Dodatkowy atrybut `target="`*`nazwa_ramki`*`"` umożliwia otwarcie strony docelowej w innym oknie (na innej karcie) przeglądarki.



Przykład 2. Tworzenie hiperłączy

```
<a href="index2.html">Dalej</a>
```

Po kliknięciu tekstu *Dalej* w oknie przeglądarki zostanie wyświetlona zawartość pliku *index2.html*.

```
<a href="https://www.gov.pl/web/edukacja" target="_blank">
Ministerstwo Edukacji Narodowej</a>
```

Po kliknięciu tekstu *Ministerstwo Edukacji Narodowej* otworzy się nowe okno (nowa karta) przeglądarki, w którym zostanie wyświetlona strona Ministerstwa Edukacji Narodowej.

```
<a href="index2.html"><img src="rys.gif"></a>
```

Aktywowanie hiperłącza następuje poprzez kliknięcie rysunku *rys.gif*.

Tworzenie kotwic wewnętrznych

Przy długich dokumentach zdarza się, że potrzebujemy wyświetlić określone miejsce takiego dokumentu (np. jeżeli tekst podzielony jest na rozdziały – początek rozdziału).

Język HTML pozwala na zaznaczanie wybranych miejsc za pomocą kotwic (ang. *anhor*). Kotwicę tworzymy poleceniem: `<a name="`*`nazwa_kotwicy`*`"></a>`, gdzie *`nazwa_kotwicy`* to dowolny, wybrany przez nas ciąg znaków.

Aby następnie przejść do zaznaczonego w ten sposób miejsca, tworzymy hiperłącze poleceniem: `<a href="plik.html#`*`nazwa_kotwicy`*`">Treść hiperłącza</a>`, gdzie `plik.html` to nazwa pliku, w którym wstawiliśmy kotwicę.

Jeżeli odnośnik ma przenieść nas do kotwicy w tym samym dokumencie, nazwę pliku można pominąć: `<a href="#nazwa_kotwicy">Treść hiperłącza</a>`.

2.3. Tworzenie podstrony

Na stronie głównej umieścimy ogólne informacje o Wyspach Kanaryjskich, a szczegółowe o poszczególnych wyspach przedstawimy na podstronach. Na stronie głównej musi być umieszczone hiperłącze prowadzące do podstrony, a na podstronie – hiperłącze przenoszące z powrotem do strony głównej. Podstronę najlepiej zapisać w pliku z rozszerzeniem *html* w tym samym folderze, co strona główna.

Ćwiczenie 5. Tworzymy podstronę i dodajemy hiperłącze do strony głównej

1. Utwórz w programie Notepad++ podstronę o wybranej wyspie, np. Lanzarote. Umieść na stronie w nagłówku tytuł, a pod nim trzy zdania opisujące wyspę (możesz utworzyć trzy akapity lub zastosować listę).
2. Zapisz podstronę w pliku pod nazwą *lanzarote.html* w folderze *Strona*.
3. Na stronie głównej umieść hiperłącze na nazwie „Lanzarote", prowadzące do podstrony *lanzarote.html*, a na podstronie – w lewym górnym rogu ikonę z linkiem powrotu do strony głównej.
4. Zapisz obydwa pliki: stronę główną i podstronę. Odśwież stronę w przeglądarce. Sprawdź poprawność działania hiperłączy.

2.4. Wstawianie obrazów

Podstawowe formaty obrazów wykorzystywanych na stronach WWW to: GIF, PNG, JPEG.

Aby umieścić na stronie rysunek, należy zastosować znacznik `<img src="źródło_pliku">`. Znacznik ten nie ma postaci zamykającej.
Na przykład: `<img src="zdjecie.jpg">`

Na stronie można także umieścić dowolny rysunek z Internetu, pod warunkiem, że znamy jego adres URL. Wykorzystując na swojej stronie obrazy pochodzące z innych stron internetowych, należy pamiętać o przestrzeganiu praw autorskich (temat D1).

Znacznik `<img>` posiada wiele dodatkowych atrybutów, które można wykorzystać do określenia położenia obrazu na stronie. Najważniejsze z nich zebrano w tabeli 4.

Nazwa atrybutu	Funkcja
`align="wyrównanie"`	Określa położenie rysunku względem strony. Możliwe wartości: `left`, `right`, `center`.
`alt="tekst"`	Określa tekst pokazywany w sytuacji, gdy niemożliwe jest wyświetlenie obrazu.
`width="szerokość"`	Określa szerokość rysunku. Szerokość powinna być podana w pikselach.
`height="wysokość"`	Określa wysokość rysunku. Wysokość może być podana w pikselach (względem wysokości oryginalnego rysunku).

***Tabela 4.** Wybrane atrybuty znacznika* `<img>`

Na przykład:

```
<img src="teneryfa.jpg" align="left" width="300" height="200">
```

wyświetla obraz wyrównany do lewej strony, o szerokości 300 i wysokości 200 pikseli.

Ćwiczenie 6. Dodajemy obrazy do podstrony

1. Otwórz plik *lanzarote.html*, zapisany w ćwiczeniu 5. Dodaj do strony przed tekstem opisującym wyspę dwa wybrane obrazy (rysunki, zdjęcia).
2. Zapisz plik pod tą samą nazwą.

Wskazówka: Aby mieć pewność, że tekst będzie zaczynał się pod obrazem, należy poprzedzić tekst znacznikiem `<br>` z atrybutem `clear="all"`: `<br clear="all">`.



Rys. 4. *Przykładowy wygląd podstrony z wstawionymi zdjęciami i ikoną powrotu do strony głównej*

2.5. Wstawianie tabel

W języku HTML **tabele** składają się z **wierszy**, a wiersze – z **komórek**. Granice tabeli określają znaczniki `<table>` ... `</table>`, granice wierszy tabeli – znaczniki `<tr>` ... `</tr>`, komórek tabeli – znaczniki `<td>` ... `</td>`.

Przykład 3. Tworzenie tabeli

Utworzymy tabelę składającą się z dwóch wierszy – każdy z dwóch komórek.

```
<table>
   <tr>
      <td>Wiersz 1, komórka 1</td>
      <td>Wiersz 1, komórka 2</td>
   </tr>
   <tr>
      <td>Wiersz 2, komórka 1</td>
      <td>Wiersz 2, komórka 2</td>
   </tr>
</table>
```

Dodatkowe atrybuty pozwalają określić sposób wyświetlania tabeli. Najważniejsze z nich zebrano w tabeli 5.

Nazwa atrybutu	Funkcja
Znacznik `<table>`	
`align="`*`wyrównanie`*`"`	Określa położenie tabeli względem strony. Możliwe wartości: `left`, `right`, `center`.
`width="`*`szerokość`*`"`	Określa szerokość tabeli. Szerokość może być podana w pikselach lub procentowo, względem szerokości elementu nadrzędnego (np. całej strony).
`border="`*`grubość`*`"`	Określa grubość (w pikselach) obramowania tabeli i komórek tabeli. Jeżeli atrybut nie zostanie użyty lub będzie miał wartość 0, linie nie będą wyświetlane.
`cellspacing="`*`odstęp`*`"`	Określa odstęp (w pikselach) pomiędzy komórkami tabeli.
`cellpadding="`*`odstęp`*`"`	Określa odstęp (w pikselach) pomiędzy krawędziami komórek tabeli a zawartością komórek.
Znacznik `<td>`	
`align="`*`wyrównanie`*`"`	Określa położenie zawartości komórki względem komórki. Możliwe wartości: `left`, `right`, `center`, `justify`.
`width="`*`szerokość`*`"`	Określa szerokość komórki. Szerokość może być podana w pikselach lub procentowo (względem szerokości tabeli). Raz określona szerokość komórki obowiązuje dla kolejnych wierszy.
`colspan="`*`liczba_komórek`*`"`	Powoduje połączenie komórki z komórkami po prawej stronie komórki; `liczba_komórek` określa całkowitą liczbę połączonych komórek.
`rowspan="`*`liczba_komórek`*`"`	Powoduje połączenie komórki z komórkami umieszczonymi poniżej komórki; `liczba_komórek` określa całkowitą liczbę połączonych komórek.

Tabela 5. *Atrybuty znaczników* `<table>` *i* `<td>`

Dane statystyczne Lanzarote		
Powierzchnia	845,9 km^2	
Populacja	Gęstość zaludnienia	227,6 os/km^2
	Liczba ludności	142 132

Rys. 5. *Przykładowy wygląd tabeli na podstronie*

Ćwiczenie 7. Dodajemy tabelę do podstrony

1. Otwórz plik podstrony *lanzarote.html*, zapisany w ćwiczeniu 6. Dodaj tabelę, w której wierszach umieścisz podstawowe informacje o wyspie: dane dotyczące powierzchni, populacji itp. – jak pokazano na rysunku 5.
2. Zapisz plik z podstroną pod tą samą nazwą.

Wskazówka: Aby zapisać km^2, użyj znacznika `<sup> ... </sup>`: `km<sup>2</sup>`.

2.6. Stosowanie kolorów

Wielu elementom strony WWW można nadawać odrębne kolory. Kolor można określić albo za pomocą angielskiej nazwy, albo podając jego wartość w systemie RGB. Na wartość w systemie RGB składają się, poprzedzone krzyżykiem, trzy liczby zapisane w systemie szesnastkowym, odpowiadające składowym koloru (czerwonej, zielonej i niebieskiej). W ten sposób można opisać dowolną z ponad 24 milionów barw dostępnych na ekranie monitora.

Przykład 4. Stosowanie kolorów

```
<font color="red">czerwony tekst</font>
<font color="#ff0000">czerwony tekst</font>
```

Ćwiczenie 8. Stosujemy kolory na stronie

1. Znajdź w Internecie w specyfikacji języka HTML pełen wykaz kolorów i opis tworzenia kolorów w systemie RGB.
2. Na stronie głównej o Wyspach Kanaryjskich, zapisanej w ćwiczeniu 5. i na podstronie zapisanej w ćwiczeniu 7., zmień według własnego pomysłu kolory nagłówków. Zapisz pliki pod tymi samymi nazwami.

3. Kaskadowe arkusze stylów CSS

Kaskadowe arkusze stylów pozwalają opisywać sposób wyświetlania stron WWW. Stanowią uzupełnienie znaczników języka HTML, udostępniając jednocześnie znacznie szerszy zakres możliwości w zakresie kształtowania wyglądu strony.

Przykładowa reguła zapisana w języku CSS może mieć następującą postać:

```
p { font-family: Verdana; font-size: 11px; }
```

Przed nawiasami klamrowymi umieszczamy **selektor**, określający elementy strony, których dotyczy dana definicja. Jeżeli definicja ma dotyczyć kilku różnych typów elementów, ich nazwy oddzielamy przecinkami. W nawiasach klamrowych podajemy nazwy **atrybutów** i ich wartości (po wartości atrybutu należy umieścić średnik). Podana przykładowa reguła dotyczy akapitów (znaczniki `<p>`) i oznacza, że będą one wyświetlane z użyciem czcionki Verdana o rozmiarze 11 pikseli. Przykład 5. pokazuje, w jaki sposób można dołączyć podaną wyżej regułę do dokumentu HTML.

Przykład 5. Dołączanie stylów CSS do dokumentów HTML

Istnieją dwa sposoby dołączania stylów CSS do dokumentów HTML. Pierwszy z nich polega na umieszczeniu stylów bezpośrednio w dokumencie, wewnątrz znacznika `<style>`, dodanego w nagłówku dokumentu HTML.

```
<head>
  <style type="text/css">
    p { font-family: Verdana; font-size: 11px; }
  </style>
</head>
```

Drugi sposób polega na umieszczeniu reguł CSS w osobnym pliku. W nagłówku dokumentu HTML dodajemy wtedy odwołanie do pliku stylów CSS.

```
<head>
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
```

W przypadku takiego odwołania reguły CSS powinny być umieszczone w pliku o nazwie *style.css*, zapisanym w tym samym katalogu co dokument HTML. Nazwa i położenie pliku z arkuszem stylów mogą być dowolne – określamy je za pomocą atrybutu `href` w podobny sposób, w jaki wskazywaliśmy elementy docelowe hiperłączy.

Reguły CSS zawarte wewnątrz znacznika `<style>` umieszczonego w dokumencie mają pierwszeństwo przed regułami dołączanymi z zewnętrznych plików za pomocą znacznika `<link>`.

Ćwiczenie 9. Stosujemy style do zmiany parametrów czcionki

1. Używając jednego ze sposobów opisanych w przykładzie 5., dodaj do strony o Wyspach Kanaryjskich regułę CSS określającą czcionkę akapitów. Wypróbuj różne wielkości i kroje czcionek. Następnie zmodyfikuj regułę tak, aby zamiast akapitów dotyczyła całego dokumentu.
2. Zapisz plik pod tą samą nazwą.

Wskazówki:

- Aby zmienić czcionkę w akapitach i listach, można wpisać polecenie w taki sposób:

```
p, ul, ol { font-family: Arial; font-size: 12 px; }
```

  Polecenie to zmieni czcionkę w akapitach, listach uporządkowanych i nieuporządkowanych na Arial o rozmiarze 12 pikseli.
- Aby zmienić czcionkę w całym dokumencie, należy dodać regułę dotyczącą znacznika `<body>`.

3.1. Najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości

Nazwa atrybutu	Funkcja
`width:` *`szerokość`*	Określa szerokość elementu.
`height:` *`wysokość`*	Określa wysokość elementu.
`margin:` *`wielkość_marginesów`*	Określa wielkość marginesów dla elementu, w kolejności: górny, prawy, dolny, lewy.
`border:` *`grubość styl kolor`*	Określa grubość, styl i kolor obramowania.
`font-family:` *`rodzina_krojów`*	Określa rodzinę krojów pisma dla tekstu.
`font-size:` *`wielkość`*	Określa wielkość pisma dla tekstu.
`font-style:` *`lista_atrybutów`*	Określa atrybuty tekstu. Możliwe wartości: `normal`, `italic`, `oblique`.
`font-weight:` *`pogrubienie`*	Określa pogrubienie tekstu. Możliwe wartości: `normal`, `bold`, `bolder`, `lighter` lub liczba określająca grubość pisma.
`color:` *`kolor`*	Określa kolor elementu.
`background-color:` *`kolor_tła`*	Określa kolor tła elementu.

Tabela 6. *Najczęściej wykorzystywane atrybuty CSS*

Sposób określania wartości atrybutów	
`px`	Określa wielkość w pikselach. **Przykład:** `{ font-size: 10px; }`
`pt`	Określa wielkość w punktach. **Przykład:** `{ font-size: 12pt; }`
`%`	Określa wielkość w procentach (względem elementu nadrzędnego). **Przykład:** `{ width: 50%; }`
`in, cm, mm`	Określa wielkość odpowiednio w calach, centymetrach lub milimetrach. **Przykład:** `{ margin: 2cm 4cm 3cm 4cm; }`
`em`	Określa wielkość pisma relatywnie do aktualnie ustawionej wielkości pisma. **Przykład:** `{ font-size: 2em; }`

Tabela 7. *Sposoby określania wartości atrybutów*

3.2. Formatowanie hiperłączy

Jednym z elementów, których wygląd często modyfikuje się za pomocą stylów CSS, jest hiperłącze. Arkusze stylów CSS pozwalają na osiągnięcie efektów niemożliwych do uzyskania przy zastosowaniu jedynie znaczników HTML.

Przykład 6. Zmiana koloru hiperłączy

Arkusze stylów CSS pozwalają łatwo zmienić wygląd hiperłączy znajdujących się na stronie. Aby zmienić kolory hiperłączy, wystarczy dodać do arkusza stylów następujące trzy reguły:

```
a { color: #000080; }
a:hover { color: #ff0000; }
a:visited { color: #808080; }
```

Oznaczają one, że domyślnie hiperłącza będą koloru granatowego (`#000080`). Po najechaniu (ang. *hover*) na hiperłącze kursorem jego kolor zmieni się na czerwony (`#ff0000`). Odwiedzone (ang. *visited*) hiperłącza będą koloru ciemnoszarego (`#808080`).

Przykład 7. Ustawianie podświetlania hiperłączy po najechaniu kursorem myszy

Ciekawy efekt można uzyskać, zmieniając kolor tła hiperłącza, na które najedzie kursor. Na przykład reguła:

```
a:hover { background-color: #fffecc; }
```

oznacza, że gdy kursor najedzie na hiperłącze, będzie ono podświetlane na żółto (`#fffecc`).

Ćwiczenie 10. Zmieniamy wygląd hiperłączy

1. Korzystając z przykładów 6. i 7., zmień wygląd hiperłączy na stronie głównej o Wyspach Kanaryjskich. Dobierz dla hiperłączy wybrane przez siebie kolory.
2. Zapisz plik pod tą samą nazwą.

3.3. Tło strony

Przykład 8. Zmienianie tła strony

Aby zmienić tło strony na jasnoniebieskie, dodajemy do arkusza stylów CSS definicję:

```
body { background-color: #d5f3ff; }
```

Możliwe jest także ustawienie jako tła strony obrazka.

```
body { background-image: url(obrazek.gif); }
```

W podanej formule `obrazek.gif` to nazwa obrazka, który ma zostać użyty jako tło. Warto zauważyć, że jeżeli obrazek tła jest mniejszy od okna przeglądarki, będzie on powielany w pionie i poziomie.

Ćwiczenie 11. Zmieniamy tło strony

1. Zmień tło strony głównej o Wyspach Kanaryjskich na wybrany przez siebie kolor lub obrazek.
2. Zapisz plik pod tą samą nazwą.

3.4. Klasy elementów

Przy tworzeniu stron internetowych często zachodzi potrzeba zmiany wyglądu jedynie wybranych elementów strony – na przykład, gdy chcemy, aby niektóre akapity były zapisane inną czcionką od pozostałych. W takiej sytuacji jednym z rozwiązań jest zastosowanie znacznika `<font>`, jednak znacznik ten musielibyśmy powtarzać dla każdego akapitu, co spowodowałoby wydłużenie kodu HTML i utrudniłoby jego późniejszą modyfikację. Znacznie lepszym rozwiązaniem jest zastosowanie klas CSS.

Przykład 9. Stosowanie klas CSS

W arkuszu stylów CSS definiujemy wygląd obiektów danej klasy, tworząc regułę, w której jako selektor podajemy kropkę i nazwę klasy, na przykład:

```
.mojaKlasa { font-size: 11px; font-weight: bold; }
```

Od tej pory wszystkie elementy tej klasy będą używały pogrubionej czcionki o rozmiarze 11 pikseli.
Aby określić klasę elementu, dodajemy do znacznika określającego element atrybut `class="nazwa_klasy"`, na przykład:

```
<p class="mojaKlasa">Ten akapit będzie używał czcionki określonej
w definicji klasy mojaKlasa.</p>
```

Ta sama klasa może być na stronie używana wielokrotnie do formatowania elementów różnych typów. Aby sformatować jedynie fragment tekstu, zawarty na przykład wewnątrz akapitu, można użyć znacznika `<span> ... </span>`:

```
<p>Ten <span class="mojaKlasa">fragment tekstu</span> będzie używał
czcionki określonej w definicji klasy mojaKlasa.</p>
```

4. Elementy dynamiczne na stronie – skrypty w języku JavaScript

Strony WWW nie muszą być statycznymi dokumentami wyświetlającymi tekst. Dzięki zastosowaniu programów komputerowych (skryptów) możliwe jest uzyskanie wielu interesujących efektów, takich jak dynamiczne generowanie treści (np. na podstawie informacji pobranych z bazy danych) czy dodanie interaktywnych elementów reagujących na działania użytkownika.

Wyróżniamy dwa rodzaje skryptów:

- **skrypty działające po stronie serwera** – skrypty takie uruchamiane są na serwerze za każdym razem, gdy użytkownik chce otworzyć stronę; zajmują się generowaniem kodu strony, przetwarzaniem danych przekazanych przez użytkownika w formularzach itp.;
- **skrypty działające po stronie przeglądarki** – umożliwiają dodanie do strony elementów interaktywnych, np. przycisków zmieniających wygląd po najechaniu na nie kursorem myszy, a także modyfikację treści i wyglądu strony w zależności od działań użytkownika.

Najczęściej używane języki skryptowe, działające po stronie serwera, to: VBScript (technologia ASP), Visual Basic, C# (technologia ASP.NET), PHP, Python, JavaScript (technologia Node.js).

Językiem skryptowym działającym po stronie przeglądarki jest JavaScript. Składnia poleceń tego języka jest, w podstawowym zakresie, zbliżona do składni języka C++.

W przykładzie 10. pokazujemy przykład skryptu działającego po stronie przeglądarki, napisanego w języku JavaScript. Wyświetla on aktualną datę i godzinę przy każdym otwarciu strony.

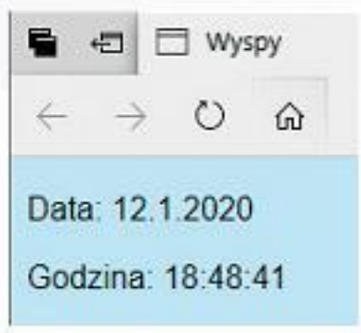
Przykład 10. Tworzenie skryptu w języku JavaScript

Aby wyświetlić aktualną datę i godzinę, korzystamy z następującego skryptu:

- za pomocą słowa kluczowego `var` tworzymy zmienną o nazwie `data` (zmienne w języku JavaScript, podobnie jak w języku Python, nie posiadają z góry określonego typu),
- za pomocą instrukcji przypisania = przypisujemy zmiennej `data` aktualną datę i godzinę,
- używamy instrukcji wyprowadzania danych `document.writeln` do wypisania w treści strony tekstu (ujętego w pojedyncze lub podwójne cudzysłowy) albo wartości: aktualnego dnia, miesiąca, roku (`date.getDate()`, `date.getMonth()`, `date.getFullYear()`) oraz aktualnej godziny. minuty i sekundy (`date.getHours()`, `date.getMinutes()`, `date.getSeconds()`),
- każdą instrukcję kończymy średnikiem.

```
<script type="text/javascript">
var data = new Date();
   document.writeln('<p>Data: ');
   document.writeln(data.getDate() + '.' + (data.getMonth()+1) + '.' +
   data.getFullYear());
   document.writeln('</p>');
   document.writeln('<p>Godzina: ');
   document.writeln(data.getHours() + ':' + data.getMinutes() + ':' +
   data.getSeconds());
   document.writeln('</p>');
</script>
```

Podany skrypt wyświetli datę i godzinę w następującej postaci:



Ćwiczenie 12. Dodajemy do strony wyświetlanie daty i godziny

1. Dodaj do strony głównej *index.html* o Wyspach Kanaryjskich skrypt wyświetlający aktualną datę i godzinę.
2. Zmodyfikuj skrypt podany w przykładzie 10., aby data i godzina wyświetlane były w jednej linii, rozdzielone przecinkiem.
3. Zapisz plik pod tą samą nazwą.

Wskazówka: Skrypt umieść w sekcji `<body>`.

Język JavaScript posiada wszelkie cechy typowe dla języków programowania, np. możliwość zaprogramowania iteracji.

Przykład 11. Wyświetlenie kwadratów dwudziestu początkowych liczb naturalnych

```
<script type="text/javascript">
   for(var i=1; i<=10; i++)
      document.writeln(i*i);
</script>
```

Ćwiczenie 13. Dodajemy do strony wyświetlanie daty i godziny

1. Sprawdź działanie skryptu z przykładu 11., tworząc prostą stronę w programie Notepad++.
2. Utwórz folder *Matematyka* i zapisz w nim stronę w pliku pod nazwą *index.html*. Wyświetl stronę w przeglądarce.
3. Dodaj skrypt wyświetlający w drugim akapicie wartość sześcianów dziesięciu początkowych liczb naturalnych. Zapisz plik pod tą samą nazwą i odśwież stronę.

5. Publikowanie i promowanie stron internetowych w Internecie

5.1. Publikowanie strony WWW

Aby nasza witryna była widoczna dla innych, musimy składające się na nią pliki umieścić na serwerze WWW.

Mamy do wyboru cztery rozwiązania:

1. możemy uruchomić serwer WWW na naszym własnym komputerze (na przykład korzystając z bezpłatnego oprogramowania Apache HTTP, dostępnego między innymi dla systemów Windows i Linux);
2. jeżeli pracujemy w sieci lokalnej podłączonej do Internetu i serwer tej sieci jest odpowiednio skonfigurowany, możemy opublikować witrynę na tym serwerze;
3. możemy umieścić witrynę na powszechnie dostępnym, darmowym serwerze WWW;
4. możemy umieścić witrynę na komercyjnym serwerze WWW.

Każde z tych rozwiązań ma swoje wady i zalety (tabela 8.).

1.	2.	3.	4.
⊕ Zalety			
⊕ możliwość samodzielnego skonfigurowania serwera w zależności od potrzeb ⊕ objętość strony WWW ograniczona jedynie pojemnością dysku twardego ⊕ łatwe zarządzanie plikami strony internetowej (znajdują się na lokalnym dysku komputera) ⊕ brak konieczności ponoszenia kosztów utrzymania strony WWW	⊕ w większości przypadków brak konieczności ponoszenia kosztów utrzymania strony WWW ⊕ wygodny dostęp do serwera (połączenie przez sieć lokalną) ⊕ witryna dostępna przez 24 godziny na dobę	⊕ rozwiązanie darmowe ⊕ witryna dostępna przez 24 godziny na dobę ⊕ stały adres strony	⊕ witryna dostępna przez 24 godziny na dobę ⊕ stały adres strony ⊕ możliwość korzystania z pomocy technicznej dostawcy usług ⊕ szybkie serwery, zwykle z gwarancją dostępności ⊕ możliwość korzystania z zaawansowanych funkcji serwera (np. bazy danych, skrypty uruchamiane po stronie serwera)
⊖ Wady			
⊖ witryna dostępna jest tylko wtedy, gdy nasz komputer jest włączony ⊖ konieczność posiadania publicznego, najlepiej stałego, adresu IP ⊖ konieczność samodzielnego nadzorowania zainstalowanego oprogramowania serwera (wymagana jest wiedza fachowa) ⊖ pracujący serwer obciąża łącze, procesor i pamięć naszego komputera	⊖ objętość witryny ograniczona ilością miejsca udostępnionego przez administratora serwera ⊖ łącza w sieciach lokalnych są zwykle w ciągu dnia mocno obciążone przez użytkowników, przez co dostęp do strony WWW może być utrudniony ⊖ najczęściej brak gwarancji bezawaryjnego działania serwera	⊖ funkcjonalność darmowych serwerów WWW jest zwykle ograniczona (np. brak obsługi skryptów uruchamianych po stronie serwera) ⊖ objętość witryny ograniczona jest zwykle do kilkudziesięciu megabajtów ⊖ ograniczony transfer danych, który może generować nasza witryna ⊖ brak gwarancji bezawaryjnego działania serwera ⊖ wielu dostawców darmowych usług wymaga umieszczania reklam na naszych stronach	⊖ koszty utrzymania strony (zwykle kilkaset złotych rocznie) ⊖ ograniczony transfer danych, który może generować nasza witryna (jeżeli witryna generuje duży ruch, zwykle trzeba ponosić dodatkowe koszty)

Tabela 8. *Wady i zalety czterech sposobów publikowania strony w Internecie*

5.2. Promowanie strony WWW

Jeżeli uważamy, że nasza witryna została profesjonalnie przygotowana i może zainteresować innych, warto rozreklamować ją w Internecie. W tym celu można podjąć następujące działania:

- dodać witrynę do jednego z internetowych katalogów stron WWW;
- wykonać optymalizację witryny pod kątem wyszukiwarek internetowych (opisać poszczególne strony za pomocą słów kluczowych, odpowiednio dobrać tytuły stron i zmodyfikować publikowane treści tak, aby zawierały słowa kluczowe związane z tematyką strony, często wyszukiwane przez internautów);
- jeżeli witryna jest interesująca i nowatorska, można wysłać informacje o niej do mediów;
- promować stronę w mediach społecznościowych;
- wykupić płatne reklamy, na przykład bannery czy odnośniki w wyszukiwarkach.

Jeżeli nasza witryna cieszy się popularnością, można spróbować na niej zarabiać. Najprostsza metoda to uczestnictwo w jednym z programów reklamowych (np. Google AdSense). W ramach tego rodzaju programów zgadzamy się na prezentowanie na naszej witrynie materiałów reklamowych, w zamian za co otrzymujemy niewielką kwotę za każdą odsłonę lub kliknięcie reklamy.

Ćwiczenie 14. Sprawdzamy możliwość dodania strony do katalogu internetowego

Znajdź kilka katalogów internetowych. Sprawdź, jak są w nich zorganizowane kategorie – do której z nich pasowałaby najlepiej strona o Wyspach Kanaryjskich? Dla wybranego katalogu sprawdź również, jakie dane o stronie trzeba podać, jeśli chcemy ją dodać do katalogu, i czy dodawanie stron jest płatne.

Warto zapamiętać

- W Internecie możemy publikować materiały, tworząc własne strony internetowe.
- Do opisu stron stosujemy język znaczników hipertekstowych HTML.
- Aby utworzyć prostą witrynę, wystarczy edytor tekstu, edytor grafiki oraz program umożliwiający przesyłanie na serwer WWW (klient FTP) plików składających się na witrynę.
- Arkusze stylów CSS pozwalają w wygodny sposób modyfikować wygląd dokumentów HTML, skracając jednocześnie długość kodu i eliminując konieczność wielokrotnego powtarzania tych samych znaczników formatujących.
- Dzięki zastosowaniu skryptów jest możliwe m.in. dynamiczne generowanie treści. Wyróżniamy dwa rodzaje skryptów: działające po stronie serwera i po stronie przeglądarki.
- W języku JavaScript możemy tworzyć skrypty działające po stronie przeglądarki.
- W języku JavaScript możemy tworzyć skrypty realizujące iterację. Język ten umożliwia również obsługę zdarzeń.
- Aby wysłać pliki na zewnętrzny serwer WWW, korzystamy najczęściej z protokołu FTP.
- Aby wypromować stworzoną stronę WWW, można, między innymi: dodać informacje o stronie do katalogów internetowych, wymienić się odnośnikami z innymi witrynami o podobnej tematyce czy zoptymalizować witrynę pod kątem wyszukiwarek internetowych.

Pytania i polecenia

1. Za pomocą jakich narzędzi można tworzyć strony WWW?
2. Czym charakteryzuje się język HTML?
3. Jaka jest ogólna struktura pliku w języku HTML?
4. Czym się różni tworzenie stron internetowych za pomocą języka HTML od tworzenia stron z wykorzystaniem innych narzędzi?
5. Do czego służą arkusze stylów CSS?
6. Jakie są sposoby dołączania arkuszy stylów CSS do dokumentów HTML?
7. Jakie korzyści płyną ze stosowania skryptów? Jakie wyróżniamy rodzaje skryptów?
8. Jakie możliwości oferuje język JavaScript?
9. Jakie są możliwości umieszczenia własnej strony w Internecie? Porównaj ich wady i zalety.

Zadania

1. Dodaj do strony o Wyspach Kanaryjskich podstrony opisujące pozostałe sześć wysp i połącz je hiperłączami ze stroną główną. Zapisz wszystkie pliki w folderze *Strona*.
2. Przygotuj stronę internetową o miejscu, które chciałbyś odwiedzić podczas wakacji. Strona powinna przypominać stronę o Wyspach Kanaryjskich utworzoną w tym temacie.
3. Korzystając z wybranego przez siebie narzędzia, zaprojektuj i utwórz stronę internetową związaną z twoimi zainteresowaniami (np. zespoły muzyczne, zwierzęta świata, ulubiona dyscyplina sportowa, modelarstwo, moda). Odszukaj w Internecie podobne strony i postaraj się, aby twoja strona wyróżniała się oryginalnymi i błyskotliwymi treściami (własnymi), ciekawymi ilustracjami (np. częściowo wykonanymi samodzielnie, a nie tylko pobranymi z Internetu). Pamiętaj o zasadach korzystania z cudzych materiałów (temat D1).
4. Odszukaj w Internecie informacje o tym, w jaki sposób za pomocą arkusza stylów CSS można wyłączyć wyświetlanie podkreśleń pod hiperłączami umieszczonymi na stronie.

Dla zainteresowanych

5. Wraz z kolegami i koleżankami stwórzcie stronę internetową o waszej miejscowości (dzielnicy, osiedlu). Strona powinna zawierać informacje o geografii i historii miejsca, ciekawostkach z nim związanych, sławnych ludziach z regionu. Zadbajcie o szatę graficzną strony – możecie np. zamieścić samodzielnie wykonane zdjęcia ciekawych obiektów.
6. Napisz skrypt, który będzie wyświetlał na stronie aktualną godzinę. Wykorzystaj funkcję `setTimeout()` do cyklicznego wywoływania funkcji aktualizującej wyświetlany czas.
7. Zapoznaj się z informacjami na temat formularzy w języku HTML. Utwórz stronę wykorzystującą język JavaScript, która będzie umożliwiać obliczanie pierwiastków równania kwadratowego $ax^2 + bx + c = 0$ dla podanych przez użytkownika współczynników a, b i c.
8. Znajdź w Internecie informacje na temat języka PHP.

Temat D4. Zadania projektowe

1. Przygotowanie projektu
2. Zadania projektowe
   Zadanie 1. Rozpowszechnianie programów komputerowych
   Zadanie 2. Przestępczość komputerowa
   Zadanie 3. Szanse i zagrożenia związane z rozwojem informatyki i technologii informacyjno-komunikacyjnych

Warto powtórzyć

1. Jakie są etapy tworzenia projektu grupowego?
2. Co jest przedmiotem prawa autorskiego?
3. Jakie są zasady korzystania z cudzych materiałów?
4. Po ilu latach od śmierci twórcy wygasają autorskie prawa majątkowe?
5. Jakie są zasady korzystania z programów komputerowych?
6. Podaj przykłady przestępczości komputerowej.
7. Jakie formy komunikacji i wymiany informacji oraz usługi rozwinęły się wraz z rozpowszechnieniem Internetu?
8. Jakie są zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnych?
9. Jak się przeprowadza debatę ZA i PRZECIW?

1. Przygotowanie projektu

Zagadnienia dotyczące wybranych norm etycznych i prawnych, rozwoju informatyki i technologii informacyjno-komunikacyjnych oraz związanych z tym rozwojem szans i zagrożeń, a także przestępczości komputerowej przedstawimy w formie projektów do samodzielnego wykonania (indywidulanego lub grupowego).

Nad projektem grupowym pracuje zespół uczniów. Należy wyznaczyć koordynatorów grup, którzy przydzielą zadania szczegółowe (w tym odpowiednie czynności) każdemu członkowi grupy oraz będą czuwali nad przebiegiem realizacji projektu (nawet w przypadku grup dwuosobowych).

W każdej grupie (pod kierunkiem koordynatora) ustalamy sposób zapisywania dokumentów komputerowych w plikach (tworzenia nazw) oraz sposoby przechowywania i udostępniania plików (również z wykorzystaniem chmury).

Przed przystąpieniem do wykonania zadań szczegółowych należy zgromadzić odpowiednie materiały. Poza tematami z podręcznika można skorzystać z fachowej literatury, odpowiednich aktów prawnych (w przypadku przepisów prawa), Internetu, w tym z zasobów udostępnianych na portalach edukacyjnych. 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. Natomiast poszczególne zadania szczegółowe – w postaci dokumentów tekstowych, obrazów, filmów, odręcznych szkiców na kartkach.

Wybrane projekty można realizować, przeprowadzając debatę ZA i PRZECIW.

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.

W temacie omawiamy trzy przykładowe zadania projektowe. Inne zadania projektowe można wykonać w podobny sposób.

2. Zadania projektowe

Zadanie 1. Rozpowszechnianie programów komputerowych

Celem projektu jest przedstawienie podstawowych przepisów prawa, dotyczących rozpowszechniania programów komputerowych. Analizę problemu należy rozpocząć od zapoznania się z przepisami prawa (odpowiednimi fragmentami). Przepisy prawa autorskiego reguluje ustawa o prawie autorskim i prawach pokrewnych – tekst tej ustawy (ujednolicony, czyli z naniesionymi zmianami) powinien stanowić podstawowe źródło informacji (teksty ustaw można znaleźć np. na stronie Internetowego Systemu Aktów Prawnych Sejmu RP: *http://isap.sejm.gov.pl/*).

Zaczynamy od wyjaśnienia, co jest przedmiotem prawa autorskiego i co nie podlega prawu autorskiemu, kiedy utwór określamy jako rozpowszechniony, a kiedy jako opublikowany.

Omówienia wymagają przepisy dotyczące korzystania z cudzych materiałów, ze szczególnym uwzględnieniem utworów rozpowszechnionych (tu: programów komputerowych). Należy też przedstawić rodzaje licencji na programy komputerowe.

Zadania szczegółowe

1. Terminologia prawa autorskiego, m.in.: *prawo autorskie*, *przedmiot prawa autorskiego*, *autor*, *utwór*, *autorskie prawa osobiste*, *autorskie prawa majątkowe*, *utwór rozpowszechniony*, *utwór opublikowany*, *licencja*.
2. Korzystanie z cudzych utworów, w tym programów komputerowych. Wyjaśnienie, kiedy można skorzystać z cudzego utworu bez pytania o zgodę i jakie kary grożą za rozpowszechnianie cudzego utworu bez zgody autora.
3. Zasady korzystania z programów komputerowych, w tym różne typy licencji na programy komputerowe.

Uwagi dotyczące realizacji projektu

- Terminologię opracowaną w pierwszym zadaniu szczegółowym można przedstawić w formie słowniczka terminów prawniczych, umieszczonego na ostatnich stronach (lub slajdach) projektu. W miejscu pierwszego pojawienia się danego hasła w dokumencie wystarczy umieścić hiperłącze do jego definicji umieszczonej w tym samym dokumencie. Takie rozwiązanie jest możliwe w przypadku przygotowywania projektu w postaci prezentacji multimedialnej, jak i w formie dokumentu tekstowego, tworzonego w edytorze tekstu.
- Opracowując drugie zadanie szczegółowe, należy wyjść od odpowiednich przepisów prawnych (wybranych z tematu D1). Przepisy należy zilustrować przykładami.
- Zagadnienie licencji na programy komputerowe (trzecie zadanie szczegółowe) trzeba omówić, podając nie tylko podstawowe rodzaje licencji, ale także wyjaśniając, na czym polega korzystanie z nich. Wskazane jest posłużenie się konkretnymi przykładami.
- Proponowana liczba osób w grupie: 3-5; w przypadku pięciu osób: zadanie 1. opracowuje jedna osoba, a zadania 2. i 3. – po dwie osoby.

Zadanie 2. Przestępczość komputerowa

Celem projektu jest omówienie zagadnienia przestępczości komputerowej, w tym piractwa komputerowego i nielegalnych transakcji w Sieci.

Realizację projektu należy rozpocząć od określenia, czym jest przestępstwo komputerowe w rozumieniu przepisów prawa. Następnie należy zapoznać się z rodzajami takich przestępstw i wyszukać najnowsze informacje na temat przestępczości komputerowej.

Należy też spróbować odszukać, jakie kary grożą za wybrane przestępstwa. W tym celu można skorzystać z odpowiednich przepisów Kodeksu karnego lub informacji umieszczonych na fachowych stronach w Internecie.

Zadania szczegółowe

1. Przykłady przestępstw komputerowych.
2. Wybrane przepisy prawa, dotyczące kar za przestępstwa komputerowe, na przykład za piractwo komputerowe czy podrzucanie wirusów.

Uwagi dotyczące realizacji projektu

- Projekt można przygotować w postaci prezentacji multimedialnej i połączyć (w miarę możliwości) opis danego przestępstwa z przewidywaną karą za jego popełnienie.
- Jeśli są takie możliwości, warto odnieść się do kilku rzeczywistych przykładów łamania prawa w zakresie przestępczości komputerowej. Konkretne przykłady bardziej przemawiają do słuchaczy niż same definicje.
- Proponowana liczba osób w grupie: 2-4.

Zadanie 3. Szanse i zagrożenia związane z rozwojem informatyki i technologii informacyjno-komunikacyjnych

Celem projektu jest przedstawienie szans i zagrożeń dla rozwoju społecznego, gospodarczego oraz dla obywateli, związanych z rozwojem informatyki i TIK.

W projekcie należy pokazać, w jaki sposób rozwój informatyki i TIK ma wpływ na rozwój gospodarczy kraju i jakie daje możliwości społeczeństwu.

Rozwój informatyki i technologii informacyjno-komunikacyjnych zmienił też w wielu przypadkach standard życia osób o specjalnych potrzebach (w tym niepełnosprawnych), m.in. umożliwił zdobycie pracy, a przede wszystkim ułatwił kontakt z innymi ludźmi.

Analizując problem, należy ocenić zagrożenia i ograniczenia, z jakimi możemy się spotkać, korzystając z technologii informacyjno-komunikacyjnych. Szczególną uwagę należy zwrócić na problem uzależnienia się od komputera i Internetu. W związku z bardzo dynamicznym rozwojem informatyki i TIK, trzeba zwrócić uwagę na pojawiające się nowe zagrożenia, m.in. uzależnienie od komputera i Internetu, zagrożenia prywatności, cyberprzemoc.

Projekt zaprezentujemy w postaci debaty ZA i PRZECIW i przygotujemy go na podstawie tematu D2 i dodatkowych informacji znalezionych w Internecie.

Zadania szczegółowe

1. Przygotowanie głównych zagadnień, wokół których ma toczyć się debata. Na przykład:
   a. Kontakty między ludźmi.
   b. Dostęp do informacji.
   c. (Nie)wiarygodność informacji.
2. Wstępne przygotowanie tezy ZA i argumentów ZA daną e-usługą.
3. Wstępne przygotowanie tezy PRZECIW i argumentów PRZECIW danej e-usłudze, czyli za usługą w wersji tradycyjnej.
4. Wstępne przygotowanie końcowych komunikatów.

Uwagi dotyczące realizacji projektu

- Ze względu na bardzo rozległą tematykę zadania należy wybrać tylko kilka zagadnień, skupiając się na wybranych przykładach.
- Klasę dzielimy na dwie grupy. Koordynator projektu przydziela grupom wybrane zagadnienia. W każdej grupie muszą znaleźć się zwolennicy i przeciwnicy danego zagadnienia. Pozostała część klasy będzie publicznością.
- Debata powinna być wcześniej wstępnie przygotowana. Jeśli w trakcie jej trwania zostaną postawione nowe argumenty, 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 (propozycja w pierwszej kolumnie tabeli 1.).
- Proponowana liczba osób w grupie: 4-6.

- Podczas wykonywania tego projektu można wykorzystać również materiały zgromadzone do przygotowania zadania 2., ponieważ przestępczość komputerowa, stanowiąca zagrożenie, wynika z rozwoju informatyki i TIK.
- Proponowana liczba osób w grupie: 3-5.

W tabeli 1. pokazano przykładowe argumenty ZA (określające pozytywne aspekty rozwoju technologii informacyjno-komunikacyjnej) i PRZECIW (określające zagrożenia). W trakcie przeprowadzania debaty modyfikujcie i uzupełniajcie na bieżąco tę tabelę.

Zagadnienie	Argumenty	
	ZA, czyli szanse	**PRZECIW, czyli zagrożenia**
Kontakty międzyludzkie	ułatwianie kontaktów międzyludzkich	zanik kontaktów bezpośrednich
	szanse dla osób niepełnosprawnych	anonimowość kontaktów
	możliwość poznawania wielu ludzi z odległych miejsc świata	izolacja ludzi od społeczeństwa
Dostęp do informacji	możliwość znajdowania wielu rzetelnych informacji, z różnorodnych źródeł	znajdowanie informacji niewiarygodnych, stron WWW niezgodnych z prawem, obrażających ludzką godność

Tabela 1. *Przykładowa tabela do zadania 3.*



Przykłady innych zadań projektowych

1. Budowa domowej sieci komputerowej.
2. Bezpieczeństwo i ochrona danych w sieciach komputerowych.
3. Wybrane zastosowania arkusza kalkulacyjnego.
4. Grafika rastrowa a grafika wektorowa.
5. Algorytmy porządkowania: metoda bąbelkowa a metoda przez wstawianie.
6. Wybrane przepisy prawa dotyczące TIK.
7. Nowe sposoby komunikacji i wymiany informacji.
8. Style CSS w tworzeniu stron WWW.

Dla zainteresowanych

9. Liczby Fibonacciego i złoty podział – występowanie w przyrodzie.
10. Wybierz (wybierzcie) własny problem do opracowania i wykonaj (wykonajcie) projekt.

Słownik pojęć

K

Klient
Komputer, który korzysta z zasobów (str. 10).

L

Licencja
Umowa zawierana pomiędzy osobą (instytucją), która jest właścicielem majątkowych praw autorskich do utworu (np. producentem lub dystrybutorem programu), a odbiorcą utworu (np. użytkownikiem oprogramowania) określająca warunki korzystania z utworu (s. 149).

Liczba pierwsza
Liczba naturalna większa od 1, która ma dokładnie dwa dzielniki naturalne: jedynkę i siebie samą, np.: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 (str. 130).

M

Mapowanie
Przyporządkowanie oznaczenia dysku komputera lokalnego wybranemu zasobowi na zdalnym komputerze (s. 22).

Mbps
(z ang. *Megabit per second*)
Jednostka szybkości przesyłania danych; liczba megabitów przesyłanych w ciągu sekundy (str. 12).

O

Oktet
Oznacza osiem bitów. W postaci dziesiętnej przedstawiany jest jako liczba z zakresu 0-255. Teoretycznie za pomocą czterech oktetów można przydzielić 4 294 967 296 adresów, bo 256 x 256 x 256 x 256 = $= 256^4 = 4\ 294\ 967\ 296$ (str. 14).

P

Parametr
Wartość przekazywana do funkcji (str. 111).

Podprogram
(procedura lub funkcja)
Wyodrębniona część programu, mająca unikatową nazwę i ustalony sposób wymiany danych z innymi częściami programu (s. 110).

Protokół sieciowy
Zbiór reguł, którym podlegają komunikujące się ze sobą komputery (str. 157).

R

Router
Urządzenie sieciowe umożliwiające przesyłanie pakietów danych pomiędzy różnymi sieciami komputerowymi (str. 14).

S

Serwer
Komputer, który udostępnia zasoby (str. 10).

Serwer internetowy
Komputer, który świadczy usługi w Internecie, np. przechowuje i udostępnia strony internetowe WWW (s. 159).

Ś

Ścieżka
Sekwencja prostych odcinków i/lub krzywych Béziera. W odróżnieniu od innych kształtów, może być edytowana poprzez swobodne przeciąganie dowolnego węzła, nie tylko wcześniej ustalonych uchwytów. Jak inne kształty może mieć wypełnienie i kontur (str. 93).

U

Udostępnianie
Zezwolenie na korzystanie z zasobów lokalnego komputera przez innych użytkowników sieci (str. 21).

Utwór opublikowany
Utwór, który za zezwoleniem twórcy został zwielokrotniony i którego egzemplarze udostępniono publicznie (str. 145).

Utwór rozpowszechniony
Utwór, który za zezwoleniem twórcy został w jakikolwiek sposób udostępniony publicznie (str. 145).

W

Wartość zwracana
Wartość przekazywana z funkcji do miejsca wywołania (np. programu głównego) (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 180 ćwiczeń i zadań sprawdzających, z tego ponad 60 z programowania
- ponad 100 pytań sprawdzających
- 13 zadań projektowych

ISBN 978-83-953551-2-7

