

Grażyna Koba

# **Program nauczania**

Informatyka  
dla liceum ogólnokształcącego

**MIGRA 2003**

**Konsultacja:**

Marta Skała-Kowalczyk

Michał Łętowski

**Redakcja i korekta:** Marcin Czerwiński**Opracowanie graficzne i skład:** Roman Jankowski

Program nauczania dopuszczony do użytku szkolnego  
przez ministra właściwego do spraw oświaty i wychowania  
i wpisany do wykazu programów nauczania w kształceniu ogólnym do nauczania informatyki  
na poziomie liceum ogólnokształcącego – kształcenie w zakresie rozszerzonym,  
na podstawie recenzji rzeczoznawców:  
prof. dr. hab. Andrzeja Stępnika (z rekomendacji Uniwersytetu Marii Skłodowskiej-Curie w Lublinie)  
oraz  
mgr. Jacka Stańdo (z rekomendacji Uniwersytetu Łódzkiego).

Numer dopuszczenia:

DKOS-5002-33/03

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

ISBN 83-916848-5-7

Copyright © by MIGRA Sp. z o.o., Wrocław 2003

Wydanie I

Druk i oprawa: Zakłady Graficzne MOMAG SA, Tomaszów Mazowiecki

Dystrybutor: FAMA Sp. z o.o., ul. Ściegiennego 7, 40-114 Katowice  
tel./faks (32) 259 71 60, tel. (32) 258 47 56

# Spis treści

<b>I. Założenia dydaktyczne i wychowawcze programu</b>	4
<b>II. Ogólne cele edukacyjne oraz treści nauczania.</b>	
<b>Relacja z podstawą programową</b>	
1. Podstawa programowa informatyki	5
2. Cele edukacyjne: wychowania i kształcenia	6
3. Treści nauczania	8
<b>III. Szczegółowe cele edukacyjne oraz związane z nimi treści nauczania</b>	
1. Metody rozwiązywania problemów algorytmicznych	8
2. Realizacja algorytmów w wybranym języku programowania	10
3. Zasady działania komputera i sieci komputerowych	11
4. Przetwarzanie danych w bazach danych	13
5. Wśród multimediów	14
<b>IV. Uwagi o realizacji programu</b>	15
<b>V. Przykładowy rozkład materiału</b>	16
<b>VI. Procedury osiągnięcia szczegółowych celów edukacyjnych</b>	
<b>– czynności nauczyciela i uczniów</b>	
1. Metody rozwiązywania problemów algorytmicznych	18
2. Realizacja algorytmów w wybranym języku programowania	19
3. Zasady działania komputera i sieci komputerowych	21
4. Przetwarzanie danych w bazach danych	22
5. Wśród multimediów	23
<b>VII. Opis założonych osiągnięć ucznia</b>	
<b>– przykłady wymagań na poszczególne oceny szkolne</b>	
1. Metody rozwiązywania problemów algorytmicznych	25
2. Realizacja algorytmów w wybranym języku programowania	26
3. Zasady działania komputera i sieci komputerowych	27
4. Przetwarzanie danych w bazach danych	28
5. Wśród multimediów	29
<b>VIII. Propozycje sposobów oceny osiągnięć ucznia</b>	32

# I. Założenia dydaktyczne i wychowawcze programu

*Wiedzieć, że wiemy, co wiemy,  
i że nie wiemy tego, czego nie wiemy  
– oto prawdziwa wiedza.*

Konfucjusz

W programie ujęte są w pełni cele i treści nauczania zawarte w *Podstawie programowej* przedmiotu informatyka, realizowanego w zakresie rozszerzonym. Dla autorki programu ważne jest, aby możliwa była dzięki niemu rzetelna realizacja założonego cyklu nauczania, a wiedza była prawidłowo przekazywana i skutecznie utrwalana.

Program jest tak przygotowany, aby wydatnie pomagał nauczycielom w nauczaniu informatyki w zakresie rozszerzonym. Informatyka jest dziedziną rozwijającą się tak szybko, że nie jest możliwe przekazanie uczniom wszystkich informacji z jej wielu obszarów zastosowania. Obszary te są wskazane w programie, ale pełną wiedzę na ich temat uczniowie zdobędą na studiach wyższych.

Jednym z celów *Podstawy* jest „przygotowanie do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego”. Na przykład uczeń poznaje podstawy algorytmiki i programowania w szkole ponadgimnazjalnej, ale tylko w takim stopniu, aby mógł je w pełni zrozumieć i świadomie wybrać kierunek dalszego kształcenia, np. informatycznego.

Materiał nauczania jest dostosowany do poziomu kształcenia ponadgimnazjalnego, a nie studiów wyższych. Zagadnienia programowe na tym etapie powinny inspirować uczniów i zachęcić ich do samodzielnego zgłębiania interesującej ich wiedzy. Zbyt trudne zagadnienia programowe na tym etapie mogą uczniów zniechęcić.

Autorka w swym pragmatycznym podejściu do nauczania zakłada realną realizację prezentowanego programu w każdych warunkach szkolnych.

Dążąc do wszechstronnego rozwoju ucznia należy pamiętać, że kształcenie i wychowanie powinny być ze sobą zharmonizowane. Powinny być one również konsekwentną kontynuacją założeń edukacyjnych realizowanych w poprzednich etapach edukacyjnych. Każdy nauczyciel powinien zapoznać się z nimi, zanim rozpocznie planowanie zajęć. W przypadku informatyki należy również poznać szczegółowe założenia programowe technologii informacyjnej dla szkół ponadgimnazjalnych.

Wyjątkowo istotnym założeniem edukacyjnym jest odpowiednie przygotowanie uczniów do podjęcia studiów i pracy zawodowej. Realizując cele kształcenia i wychowania wskazujemy, jak w praktyce wykorzystać wiedzę ogólną i nabyte umiejętności oraz jak w poszukiwaniu nowej wiedzy pogodzić zasady etyczne z otworzeniem się na świat.

Uczniowie poznają informatykę jako dziedzinę wiedzy (jest to szczególnie ważne dla tych, którzy zamierzają kontynuować naukę na studiach informatycznych). Uczą się podstaw algorytmiki i programowania. Szczególną uwagę zwracamy na podstawowe zasady programowania, posługując się przykładem wybranego języka programowania (jednego lub dwóch). Takie podejście jest uzasadnione metodycznie. Jeśli uczeń zrozumie metody programowania na przykładzie jednego języka, to bez większych trudności potrafi je zastosować w innych.

Uczniowie powinni sprawnie korzystać z komputera i sieci komputerowych. Dlatego w programie zakładamy poznawanie logicznego działania komputera i schematu działania sieci (w tym Internetu) oraz zagrożeń, jakie można spotkać w sieci, jak też zabezpieczenia się przed nimi.

Znacznie pogłębione zostały zagadnienia dotyczące grafiki komputerowej – edycji, przekształcania i fotomontażu obrazu. Poszerzone są zasady przygotowywania prezentacji multimedialnych. Uczniowie poznają metody tworzenia stron internetowych z wykorzystaniem nowoczesnych narzędzi oraz zasady publikowania w Internecie.

Poszerzone są wiadomości dotyczące przetwarzania danych w bazach danych w zakresie samodzielnego przygotowania relacyjnej bazy danych i wyszukiwania informacji z użyciem języka zapytań. Autorka

zakłada, że z poprzednich etapów edukacyjnych uczniowie wiedzą już, na czym polega przetwarzanie danych, i potrafią wykonywać na nich podstawowe operacje, korzystając z gotowych baz danych.

Na przykładzie projektowania systemów informatycznych uczniowie mogą dowiedzieć się, na czym polega podejmowanie indywidualnych i grupowych decyzji, skuteczne działanie w grupie, umiejętne planowanie pracy oraz nadzór nad wykonywanym projektem.

Cele edukacyjne i treści nauczania (rozdział III) oraz procedury osiągnięcia celów edukacyjnych (rozdział VI) przygotowane są (podobnie jak program technologii informacyjnej, tej samej autorki) w przejrzystej formie tabelarycznej. Szczegółowe treści nauczania przedstawione w rozdziale III odpowiadają pięciu grupom zagadnień wymienionych w tabeli (rozdział II). Cele edukacyjne, które umieszczone są po lewej stronie danej części odpowiadają treściom nauczania wymienionym po prawej. Aby umożliwić pogłębianie wiedzy uczniom bardziej zainteresowanym, treści, które można omówić dodatkowo, ujęte są jako zagadnienia nadprogramowe.

Zakładane osiągnięcia ucznia są przedstawione w postaci szczegółowych wymagań na poszczególne oceny szkolne – od 2 do 6 (rozdział VII). W programie podane są propozycje siatek godzin oraz przykładowy rozkład materiału (rozdział V).

### Uwagi

Zgodnie z zapisem w odpowiednim rozporządzeniu „dla technologii informacyjnej przedmiotem ujętym w podstawie programowej w zakresie rozszerzonym jest informatyka”. Zatem podstawową wiedzę i umiejętności uczeń zdobywa na technologii informacyjnej. Program informatyki nie ujmuje więc zagadnień podstawowych z poszczególnych działów tematycznych wynikających z podstawy programowej TI, a tylko je rozszerza i uzupełnia. Jedynym wyjątkiem są zagadnienia dotyczące algorytmiki i programowania. Tematy te nie występują bowiem w podstawie programowej technologii. W tym przypadku wyjaśniamy również (lub przypominamy z gimnazjum) materiał podstawowy.

Podstawa programowa informatyki dla zakresu rozszerzonego została określona tylko dla liceum ogólnokształcącego. Zgodnie z zapisem w odpowiednim rozporządzeniu „podstawa programowa zakłada w liceum ogólnokształcącym, liceum profilowanym i technikum kształcenie w zakresie podstawowym, natomiast w liceum ogólnokształcącym przewiduje dla wybranych, zgodnie z odrębnymi przepisami, przedmiotów dodatkowo kształcenie w zakresie rozszerzonym”.

W ramowych planach nauczania dla liceum ogólnokształcącego przewidziano dodatkowe godziny na kształcenie w zakresie rozszerzonym (10 godzin w całym cyklu kształcenia), natomiast w pozostałych typach szkół ponadgimnazjalnych nie ma takich godzin. Zapisano jednak, że dyrektor może przeznaczyć część godzin na kształcenie w zakresie rozszerzonym – z godzin do swojej dyspozycji. Ponieważ nie określono podstaw programowych do kształcenia w zakresie rozszerzonym dla szkół innych niż liceum ogólnokształcące, można więc korzystać w nich z podstawy programowej dla liceum, a także z programu oraz podręcznika opracowanego na jego podstawie.

## II. Ogólne cele edukacyjne oraz treści nauczania. Relacja z podstawą programową

### 1. Podstawa programowa informatyki

#### Cele edukacyjne

1. Przygotowanie do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego.
2. Zdolność do samodzielnego korzystania z komputera dla realizacji części zadań edukacyjnych oraz innych celów poznawczych.

### Zadania szkoły

1. Stworzenie warunków do poznania wybranych zagadnień, pojęć i metod informatyki jako dyscypliny naukowej oraz jej najważniejszych zastosowań.
2. Kształcenie samodzielności intelektualnej, odpowiedzialności za własny rozwój, gotowości do podejmowania i rozwiązywania złożonych zadań, z uwzględnieniem środków i metod informatyki.
3. Rozwijanie umiejętności pracy zespołowej przez realizację projektów grupowych.

### Treści nauczania

1. Algorytmika i programowanie:
  - 1) metodyczna analiza i modelowanie umiarkowanie złożonych problemów i procesów z różnych dziedzin,
  - 2) przegląd algorytmów klasycznych,
  - 3) wybrane techniki projektowania algorytmów i struktur danych: programowanie strukturalne, zstępujące, abstrakcja danych, metoda kolejnych uściśleń,
  - 4) elementy analizy algorytmów,
  - 5) indywidualna i zespołowa realizacja projektów programistycznych w wybranym języku wysokiego poziomu.
2. Bazy danych:
  - 1) podstawowe formy organizacji informacji w bazach danych,
  - 2) budowa relacyjnych baz danych,
  - 3) wyszukiwanie informacji w relacyjnych bazach danych z użyciem języka zapytań,
  - 4) projektowanie prostych relacyjnych baz danych.
3. Multimedia. Sieci komputerowe:
  - 1) sprawne i świadome korzystanie z multimediiów i tworzenie własnych materiałów multimedialnych,
  - 2) przetwarzanie informacji w różnej postaci (w tym wizualnej i dźwiękowej),
  - 3) budowa i działanie sieci komputerowych,
  - 4) tworzenie i publikowanie własnych materiałów w sieci.
4. Tendencje w rozwoju informatyki i jej zastosowań.

### Osiągnięcia

1. Formułowanie sytuacji problemowej, jej modelowanie i rozwiązywanie z użyciem metod informatycznych.
2. Ocenianie poprawności i efektywności rozwiązań i ich testowanie. Tworzenie dokumentów rozwiązań.
3. Wyszukiwanie informacji w bazach danych i projektowanie prostych baz danych.
4. Tworzenie opracowań multimedialnych.
5. Sprawne korzystanie z usług sieci komputerowych w pracy z informacjami własnymi i obcymi.
6. Planowanie pracy i nadzór nad przebiegiem wykonywania projektów realizowanych zespołowo z wykorzystaniem programów komputerowych.

**Źródło:** Rozporządzenie Ministra Edukacji Narodowej i Sportu z dnia 26 lutego 2002 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół (Dz.U. nr 51, poz. 458).

## 2. Cele edukacyjne: wychowania i kształcenia

Przedstawione poniżej ogólne cele edukacyjne są w rozdziale III uszczegółowione. Odniesiono je również do szczegółowych treści nauczania. Na cele edukacyjne składają się cele wychowania oraz cele kształcenia.

## Cele wychowania

### *Przygotowanie do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego:*

1. Dążenie do wszechstronnego rozwoju osobowego i społecznego oraz do osiągania wyznaczonych celów życiowych.
2. Kształcenie samodzielności w podejmowaniu decyzji dotyczących własnego rozwoju intelektualnego.
3. Rozwijanie umiejętności funkcjonowania w skomputeryzowanym świecie. Traktowanie wiadomości z informatyki w sposób integralny z innymi naukami, jako wartości poznawczej samej w sobie, prowadzącej do lepszego rozumienia świata, ludzi i siebie.

### *Zdolność do samodzielnego korzystania z komputera dla realizacji części zadań edukacyjnych oraz innych celów poznawczych:*

1. Rozwijanie osobistych zainteresowań i dociekliwości poznawczej, ukierunkowanej na poszukiwanie informacji w różnych źródłach.
2. Przewidywanie użyteczności informatyki dla życia zawodowego i społecznego.
3. Przygotowywanie do podejmowania i rozwiązywania złożonych zadań, z uwzględnieniem środków i metod informatyki.
4. Organizowanie i ocenianie własnej nauki, przyjmowanie odpowiedzialności za jej poziom.
5. Kształtowanie w sobie postawy dialogu, umiejętności współdziałania w zespole, słuchania innych, rozumienia i uwzględniania ich poglądów. Przygotowywanie prezentacji własnych poglądów.
6. Podejmowanie wyzwań współczesnego świata związanych z ogromną ekspansją środków i narzędzi informatyki.
7. Postrzeganie wpływu informatyki na zachowania społeczne. Rozumienie zalet i zagrożeń, wynikających z korzystania z nowoczesnych technologii.
8. Uświadomienie wagi prawnych i społecznych aspektów zastosowań informatyki.
9. Przestrzeganie zasad regulaminu pracowni komputerowej. Poszanowanie mienia.

## Cele kształcenia

### *Przygotowanie do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego:*

1. Poznanie pojęć, metod informatyki i zdobywanie niezbędnych umiejętności umożliwiających podjęcie pracy, studiów lub innych form doksztalcenia.
2. Poznanie wybranych zagadnień informatyki jako dyscypliny naukowej.
3. Rozumienie, a nie tylko pamięciowe opanowanie zagadnień z dziedziny informatyki.
4. Poznawanie obszarów zastosowania informatyki i zasad funkcjonowania społeczeństwa informacyjnego.
5. Poznawanie przeobrażeń w dziedzinie informatyki w perspektywie europejskiej i światowej.

### *Zdolność do samodzielnego korzystania z komputera dla realizacji części zadań edukacyjnych oraz innych celów poznawczych:*

1. Opanowanie zasad posługiwania się komputerem i sprawne korzystanie z usług sieci komputerowych.
2. Poprawne posługiwanie się językiem informatyki oraz swobodne wypowiadanie się na temat wybranych zagadnień z dziedziny informatyki.
3. Rozwijanie zdolności myślenia algorytmicznego oraz dostrzegania różnego rodzaju związków i zależności między problemem, algorytmem a programem komputerowym.
4. Wykorzystywanie umiejętności efektywnego posługiwania się urządzeniami i środkami informatyki w rozwiązywaniu zadań z innych przedmiotów.
5. Stosowanie przyswojonej wiedzy w praktyce oraz zdobywanie potrzebnych doświadczeń i tworzenie odpowiednich nawyków.
6. Rozwiązywanie problemów w sposób twórczy.

### 3. Treści nauczania

W tabeli przedstawione są główne zagadnienia treści nauczania, które w następnym rozdziale są uszczegółowione. W kolumnie obok umieszczono odpowiadające im treści nauczania z *Podstawy programowej*.

Zagadnienia	Treści nauczania z <i>Podstawy programowej informatyki</i>
1. Metody rozwiązywania problemów algorytmicznych. 1.1. Sposoby prezentacji algorytmów. 1.2. Przegląd technik algorytmicznych i algorytmów klasycznych. 1.3. Elementy analizy algorytmów.	<b>Algorytmika:</b> <ul style="list-style-type: none"> <li>• metodyczna analiza i modelowanie umiarkowanie złożonych problemów i procesów z różnych dziedzin,</li> <li>• przegląd algorytmów klasycznych,</li> <li>• elementy analizy algorytmów.</li> </ul>
2. Realizacja algorytmów w wybranym języku programowania. 2.1. Zasady programowania. 2.2. Dobór struktur danych do rozwiązywanego problemu.	<b>Programowanie:</b> <ul style="list-style-type: none"> <li>• wybrane techniki projektowania algorytmów i struktur danych, programowanie strukturalne, zstępujące, abstrakcja danych, metoda kolejnych uściśleń,</li> <li>• indywidualna i zespołowa realizacja projektów programistycznych w wybranym języku wysokiego poziomu.</li> </ul>
3. Zasady działania komputera i sieci komputerowych. 3.1. System komputerowy. 3.2. Sieci komputerowe, w tym Internet. 3.3. Kierunki w rozwoju informatyki i jej zastosowań.	<b>Sieci komputerowe:</b> <ul style="list-style-type: none"> <li>• budowa i działanie sieci komputerowych.</li> </ul> <b>Tendencje w rozwoju informatyki i jej zastosowań.</b>
4. Przetwarzanie danych w bazach danych. 4.1. Projektowanie i tworzenie relacyjnej bazy danych. 4.2. Wyszukiwanie informacji z użyciem języka zapytań. 4.3. Realizacja projektu programistycznego, w tym przygotowanie dokumentów i raportów.	<b>Bazy danych:</b> <ul style="list-style-type: none"> <li>• podstawowe formy organizacji informacji w bazach danych,</li> <li>• budowa relacyjnych baz danych,</li> <li>• wyszukiwanie informacji w relacyjnych bazach danych z użyciem języka zapytań,</li> <li>• projektowanie prostych relacyjnych baz danych.</li> </ul>
5. Wśród multimediiów. 5.1. Przetwarzanie informacji w różnej postaci, m.in. graficznej, dźwiękowej. 5.2. Tworzenie prezentacji multimedialnych. 5.3. Tworzenie stron internetowych.	<b>Multimedia:</b> <ul style="list-style-type: none"> <li>• sprawne i świadome korzystanie z multimediiów i tworzenie własnych materiałów multimedialnych,</li> <li>• przetwarzanie informacji w różnej postaci (w tym wizualnej i dźwiękowej),</li> <li>• tworzenie i publikowanie własnych materiałów w sieci.</li> </ul>

## III. Szczegółowe cele edukacyjne oraz związane z nimi treści nauczania

### 1. Metody rozwiązywania problemów algorytmicznych

#### 1.1. Sposoby prezentacji algorytmów

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
Posługiwanie się algorytmami w rozwiązywaniu zadań szkolnych i problemów życia codziennego.  Rozumienie zależności między problemem, algorytmem a programem komputerowym.  Poznanie różnych sposobów prezentacji algorytmów.  Przestrzeganie zasad zapisu algorytmów w wybranej postaci (notacji).	Związki i zależności między problemem, algorytmem i programem komputerowym.  Formułowanie sytuacji problemowej i jej modelowanie. Określenie specyfikacji problemu (zadania).  Postać prezentowania algorytmów: <ul style="list-style-type: none"> <li>• wypowiedź ustna,</li> <li>• lista kroków,</li> <li>• schemat blokowy,</li> <li>• symboliczny język programowania (pseudojęzyk).</li> </ul>



<p>Zapisywanie rozwiązania dowolnego zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie postaci (notacji).</p> <p>Dobieranie sposobu prezentacji do algorytmu.</p> <p>Stosowanie poznanych metod prezentacji algorytmów do opisywania zadań z innych przedmiotów szkolnych oraz sytuacji z różnych dziedzin życia.</p>	<p>Zapis algorytmu w postaci listy kroków.</p> <p>Tworzenie schematu blokowego algorytmu (wykorzystywanie Autokształtów w edytorze tekstu, rysowanie odręczne w zeszytach lub rysowanie w programie graficznym). Rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym.</p> <p>Zastosowanie oprogramowania edukacyjnego do graficznej prezentacji i analizy algorytmów.</p> <p>Analiza i symulacja działania algorytmu. Konstruowanie własnego schematu blokowego.</p> <p>Zapis algorytmu w symbolicznym języku programowania (pseudojęzyku).</p> <p>Przykłady instrukcji pseudojęzyka.</p> <p>Struktura programu i podstawowe zasady programowania.</p> <p><b>Pojęcia:</b> algorytm, specyfikacja zadania, lista kroków, schemat blokowy, pseudojęzyk, instrukcja.</p>
---	---

## 1.2. Przegląd technik algorytmicznych i algorytmów klasycznych

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Rozumienie różnic między algorytmem liniowym a algorytmem z warunkami oraz algorytmem iteracyjnym.</p> <p>Stosowanie technik algorytmicznych (sortowania, iteracji i rekurencji) do rozwiązywania zadań szkolnych i problemów życia codziennego.</p> <p>Poprawne stosowanie sposobów zakończenia iteracji. Rozumienie, na czym polega pętla nieskończona. Radzenie sobie z sytuacjami zapętlenia.</p> <p>Rozumienie różnic między iteracją i rekurencją.</p> <p>Stosowanie algorytmów klasycznych w typowych sytuacjach problemowych.</p> <p>Stosowanie iteracji w algorytmach sortowania.</p> <p>Stosowanie algorytmów porządkowania w typowych sytuacjach problemowych.</p> <p>Stosowanie techniki iteracji w algorytmach sortowania.</p> <p>Stosowanie algorytmów szukania minimum i maksimum w sortowaniu.</p>	<p>Wybrane rodzaje algorytmów: liniowe, z warunkami (rozgałęzieniami), iteracyjne, rekurencyjne. Zasady zapisu ww. algorytmów w liście kroków, schemacie blokowym, pseudojęzyku.</p> <p>Przykłady realizacji niektórych algorytmów w arkuszu kalkulacyjnym.</p> <p>Algorytmy z warunkami: prosta sytuacja warunkowa, warunki zagnieżdżone. Iteracja. Określanie kroku iteracji. Sposoby zakończenia iteracji: liczba powtórzeń z góry określona lub zależna od warunku. Rekurencja. Rekurencyjny zapis algorytmu.</p> <p>Przykłady algorytmów w materiale nauczania różnych przedmiotów (np. matematyki, fizyki).</p> <p>Algorytmy w życiu codziennym. Określanie, kiedy dowolny przepis (np. kucharski) jest algorytmem.</p> <p>Przykłady algorytmów z rozgałęzieniami: obliczanie wartości bezwzględnej liczby, rozwiązywanie równania liniowego i kwadratowego.</p> <p>Przykłady algorytmów iteracyjnych: dodawanie liczb, silnia, algorytm Euklidesa (NWD), NWW, szukanie minimum lub maksimum wśród danych elementów; metoda „dziel i zwyciężaj”, przeszukiwanie ciągu w poszukiwaniu wyróżnionego elementu, generowanie liczb Fibonacciego, schemat Hornera.</p> <p>Przykłady algorytmów rekurencyjnych: silnia, algorytm Euklidesa (NWD), generowanie liczb Fibonacciego, schemat Hornera.</p> <p>Przykłady algorytmów sortowania: przez wybór, bąbelkowe, przez wstawianie, kubełkowe.</p> <p>Przykłady algorytmów na liczbach naturalnych: generowanie liczb pierwszych, pozycyjne reprezentacje liczb.</p> <p>Przykłady algorytmów numerycznych: obliczanie wartości pierwiastka kwadratowego, wyznaczanie miejsca zerowego funkcji.</p> <p>Przykłady algorytmów szyfrowania i kompresji danych.</p> <p><b>Pojęcia:</b> algorytm liniowy, algorytm z rozgałęzieniami, iteracja, krok iteracji, rekurencja.</p>

### 1.3. Elementy analizy algorytmów

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Ocenianie poprawności działania algorytmu, czyli umiejętność określania, czy algorytm działa zgodnie ze specyfikacją zadania.</p> <p>Określanie uniwersalności algorytmu.</p> <p>Ocenianie złożoności obliczeniowej algorytmu.</p> <p>Stosowanie oprogramowania edukacyjnego do analizy algorytmów.</p>	<p>Analiza działania algorytmu (zapisanego w wybranej postaci) dla przykładowych danych.</p> <p>Poprawność, uniwersalność algorytmu.</p> <p>Złożoność obliczeniowa algorytmu (pracochołność) – określanie liczby operacji wykonywanych na danych, np. określanie liczby porównań elementów w algorytmie sortowania.</p> <p>Porównywanie złożoności różnych algorytmów tego samego zadania dla tych samych danych, np. porównanie dwóch algorytmów sortowania.</p>
	<p><b>Pojęcia:</b> Poprawność, uniwersalność, złożoność (pracochołność).</p>

#### Zagadnienia nadprogramowe

Prezentacja algorytmu w postaci drzewa algorytmu; drzewo wyrażenia.

Inne przykłady algorytmów sortowania, np. sortowanie szybkie.

Inne algorytmy rekurencyjne, np. problem ośmiu hetmanów, wieże Hanoi, algorytm trwałych małżeństw.

Złożoność czasowa i pamięciowa algorytmu.

## 2. Realizacja algorytmów w wybranym języku programowania

### 2.1. Zasady programowania

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Poznanie struktury wybranego języka programowania.</p> <p>Rozumienie roli procesora i pamięci operacyjnej w wykonywaniu programu.</p> <p>Umiejętność stosowania procedur w programowaniu strukturalnym. Wykorzystanie tych samych procedur w różnych programach.</p> <p>Stosowanie wybranych technik algorytmicznych w programowaniu – strukturalny zapis programu na przykładzie algorytmów porządkowania (sortowania) elementów i algorytmów rekurencyjnych.</p> <p>Wykorzystanie zasad pracy zespołowej przy projektowaniu złożonych programów.</p>	<p>Język programowania. Elementy języka (słowa kluczowe, instrukcje, wyrażenia, zasady składni).</p> <p>Klasyfikacja języków programowania.</p> <p>Zapis algorytmu w postaci programu. Struktura programu.</p> <p>Etapy programowania – implementacja, kompilacja, uruchomienie, testowanie. Błędy kompilacji i błędy wykonania.</p> <p>Rola procesora i pamięci operacyjnej w realizowaniu instrukcji i uruchamianiu (wykonywaniu) programu.</p> <p>Podstawowe zasady prawidłowego programowania.</p> <p>Wprowadzanie danych i wyprowadzanie wyników.</p> <p>Instrukcja przypisania.</p> <p>Realizacja algorytmów z warunkami w postaci programu (instrukcja warunkowa, instrukcja wyboru).</p> <p>Realizacja algorytmów iteracyjnych w postaci programu (instrukcje pętli).</p> <p>Programowanie strukturalne. Metoda zstępująca. Deklaracja procedur i funkcji. Sposoby przekazywania parametrów. Zasięg działania zmiennej. Zastosowanie procedur i funkcji w zadaniach.</p> <p>Realizacja algorytmów rekurencyjnych w języku programowania wysokiego poziomu.</p> <p>Grafika w języku programowania. Podstawowe procedury graficzne. Procedury graficzne (zaznaczanie punktów, rysowanie prostych, krzywych, płaskich figur geometrycznych). Kolorowanie obrazu. Wykres funkcji liniowej. Elementy grafiki animowanej.</p>
	<p><b>Pojęcia:</b> program, język programowania, instrukcja, słowo kluczowe, program źródłowy, implementacja, kompilacja, kompilator, translator, interpreter, zmienna, pętla, zapętlenie programu.</p>

## 2.2. Dobór struktur danych do rozwiązywanego problemu

Szczegółowe cele kształcenia	Szczegółowe treści nauczania
<p>Rozumienie, czym jest zmienna w programie komputerowym (m.in. co oznacza przypisanie zmiennej konkretnej wartości).</p> <p>Poznanie zastosowań wybranych struktur danych i właściwego ich doboru do rozwiązywanego problemu.</p> <p>Uświadomienie, jakie są konsekwencje deklaracji zmiennych określonego typu (zajętość pamięci komputera).</p>	<p>Pojęcie zmiennej. Nazwa zmiennej. Deklaracja zmiennych. Typ zmiennej.</p> <p>Ogólna klasyfikacja struktur (typów) danych.</p> <p>Przykłady typów prostych – porządkowe (całkowite, logiczne, znakowe, wyliczeniowe) i rzeczywiste.</p> <p>Metody reprezentowania ciągu znaków – typ łańcuchowy.</p> <p>Metody reprezentowania danych złożonych – typy strukturalne (typ tablicowy, rekordowy, plikowy). Zmienne indeksowane, rekordowe.</p> <p>Metody doboru struktur danych do algorytmu.</p> <p>Sposoby wprowadzania danych, wprowadzanie danych do pliku i z pliku.</p> <p>Szacowanie wielkości pamięci zajmowanej przez dane określonego typu.</p> <p><b>Pojęcia:</b> zmienna i jej typy: prosty, strukturalny, łańcuchowy.</p>

### Zagadnienia nadprogramowe

Programowanie dynamiczne. Dynamiczne struktury danych. Zmienne typu wskaźnikowego. Przykłady struktur listowych: stos, kolejka, lista (jedno- i dwukierunkowa).

Programowanie obiektowe. Przykład języka programowania zorientowanego obiektowo. Pojęcie obiektu. Definicja typu obiektowego. Dziedziczenie. Definiowanie metod. Polimorfizm i metody wirtualne.

## 3. Zasady działania komputera i sieci komputerowych

### 3.1. System komputerowy

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Poznanie, jak działa komputer. Analizowanie modelu logicznego komputera.</p> <p>Rozumienie, jaka rolę w systemie komputerowym odgrywają systemy pozycyjne: dwójkowy i szesnastkowy.</p> <p>Rozróżnianie rodzajów pamięci komputera, określanie ich własności i przeznaczenia.</p> <p>Poznanie organizacji pamięci komputerowych.</p> <p>Poprawne zarządzanie plikami i folderami.</p> <p>Rozumienie potrzeby ochrony plików.</p>	<p>Reprezentacja danych w komputerze.</p> <p>Zapis liczb dziesiętnych w systemie dwójkowym (liczb naturalnych, liczb ze znakiem, liczb rzeczywistych).</p> <p>Operacje arytmetyczne na liczbach binarnych (dodawanie i odejmowanie). Przesunięcia bitowe. Operacje logiczne.</p> <p>System szesnastkowy. Konwersje liczb z jednego systemu pozycyjnego na inny. Kompresja i szyfrowanie danych.</p> <p>Logiczny model komputera. Działanie procesora.</p> <p>Kod ASCII. System operacyjny. Najważniejsze funkcje systemu.</p> <p>Przykłady systemów operacyjnych. Zestaw komputerowy.</p> <p>Typy komputerów, w tym superkomputery.</p> <p>Ważne operacje wykonywane na plikach.</p> <p>Nadawanie plikom atrybutów. Wyszukiwanie plików.</p> <p>Odzyskiwanie utraconych plików. Ochrona plików.</p> <p><b>Pojęcia:</b> bit, bajt, system pozycyjny, kod ASCII, procesor, system operacyjny, atrybuty plików.</p>

### 3.2. Sieci komputerowe, w tym Internet

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Poznanie schematu działania sieci komputerowych. Rozróżnianie topologii sieci komputerowych.</p> <p>Uświadomienie pożytku z różnorodnego zastosowania sieci.</p> <p>Dostrzeganie zagrożeń płynących z sieci. Poznanie metod ochrony. Stosowanie podstawowych zabezpieczeń.</p> <p>Poznanie różnych sposobów połączenia z Internetem.</p>	<p>Sieć komputerowa. Praca w sieci komputerów. Korzyści wynikające z łączenia komputerów w sieć. Topologie i technologie sieciowe. Zasady wymiany informacji w sieci komputerowej. Urządzenia i elementy sieciowe. Usługi i protokoły sieciowe. Przegląd sieciowych systemów operacyjnych. Typy sieci komputerowych.</p> <p>Realizacja prostych sieci komputerowych (kryteria wyboru, niezbędne składniki). Prosta sieć domowa.</p> <p>Konfiguracja komputerów sieciowych. Sieć domowa z wydzielonym routerem. Szkolna pracownia internetowa z serwerem sieciowym. Logowanie do sieci.</p> <p>System Linux:</p> <ul style="list-style-type: none"> <li>na stacjach roboczych (zasady instalacji systemu, oprogramowanie użytkowe),</li> <li>jako serwer (serwery sieciowe Linuksa, linuksowy serwer plików).</li> </ul> <p>Współpraca z MS Windows – serwer Samba. Konfiguracja Samby.</p> <p>Bezpieczeństwo w sieci. Rodzaje zagrożeń. Przykłady programów szpiegujących. Ochrona i zabezpieczenia. Zapory sieciowe – sprzęt i programy. Konfiguracja przykładowego programu. Postępowanie przy włamaniu.</p> <p>Działanie Internetu.</p> <p>Połączenia z Internetem: modemy analogowe i cyfrowe, sieci kablowe i energetyczne, SDI i Neostrada Plus. Inne sposoby połączenia.</p> <p><b>Pojęcia:</b> sieć komputerowa, topologia sieci, magistrała, gwiazda, pierścień, hub, switch, karta sieciowa, router, protokół sieciowy, peer-to-peer, client-serwer.</p>

### 3.3. Tendencje w rozwoju informatyki i jej zastosowań

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Rozumienie wpływu rozwoju informatyki na zachowania społeczne i gospodarkę kraju.</p> <p>Dostrzeganie zalet i zagrożeń, wynikających z korzystania z nowoczesnych technologii.</p> <p>Dostrzeganie przeobrażeń w dziedzinie informatyki w kraju i na świecie.</p> <p>Uświadomienie wagi prawnych i społecznych aspektów zastosowania informatyki.</p>	<p>Nowoczesne spojrzenie na Internet. Nowości w zakresie usług internetowych.</p> <p>Historia komputerów. Pierwsze języki programowania. Najnowsze pomysły na komputery (np. komputery chemiczne, biologiczne). Sztuczna inteligencja.</p> <p>Nowoczesne zastosowania urządzeń komputerowych. Sieci przyszłości. Rozwój multimediów. Prawne i społeczne aspekty zastosowań informatyki.</p>

#### Zagadnienia nadprogramowe

Model warstwowy sieci. Systemy sieciowe Windows i NetWare. Usługi katalogowe. Uruchamianie własnych usług sieciowych – serwer pocztowy, serwer WWW, serwer FTP. Sieci bezprzewodowe Wi-Fi i BlueTooth, serwer webowy w szkole.

## 4. Przetwarzanie danych w bazach danych

### 4.1. Projektowanie relacyjnej bazy danych

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Stosowanie techniki komputerowej do gromadzenia i przetwarzania informacji.</p> <p>Poznanie zasad działania i obszarów zastosowań programów baz danych oraz korzyści z pozyskiwania i wymiany informacji.</p> <p>Rozumienie, na czym polega przetwarzanie danych.</p> <p>Uświadomienie, na czym polega relacja w bazach danych (porównanie z pojęciem relacji z matematyki).</p> <p>Rozróżnianie mechanizmów porządkowania informacji w bazie danych: sortowanie a indeksowanie.</p> <p>Poznanie podstawowych zasad ochrony danych w bazach danych.</p>	<p>Obszary zastosowania baz danych – na przykładach z najbliższego otoczenia – szkoły, zakładów pracy, instytucji naukowych i gospodarczych.</p> <p>Przykłady programów do tworzenia baz danych.</p> <p>Metody przetwarzania danych na przykładzie gotowej bazy danych. Podstawowe operacje wykonywane na danych (wprowadzanie, redagowanie, sortowanie, indeksowanie, wyszukiwanie, prezentacja).</p> <p>Projektowanie relacyjnej bazy danych, składającej się z dwóch (ewentualnie trzech) tabel połączonych relacją.</p> <p>Zasady przygotowania tabel. Określanie typu danych.</p> <p>Definiowanie kluczy podstawowych. Określanie relacji.</p> <p>Projektowanie formularzy i raportów. Kwerendy wybierające.</p> <p>Łączenie informacji z bazy danych z dokumentami tekstowymi – zasady tworzenia korespondencji seryjnej.</p> <p>Możliwości drukowania bazy danych i jej elementów.</p> <p>Drukowanie wybranych rekordów, formularzy i raportów.</p> <p><b>Pojęcia:</b> tabela, rekord, pole, typ pola, formularz, kwerenda, raport, klucz podstawowy, indeks.</p>

### 4.2. Wyszukiwanie informacji z użyciem języka zapytań

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Wykorzystywanie metod programowania w języku SQL do wyszukiwania informacji.</p> <p>Szukanie podobieństw w zastosowaniu języka zapytań SQL w strukturach poznanych języków programowania.</p>	<p>Podstawy języka SQL – zasady składni języka, podstawowe instrukcje.</p> <p>Wybrane metody tworzenia kwerend z wykorzystaniem języka SQL. Zastosowanie instrukcji SELECT do wybierania kolumn z tabel bazy danych.</p> <p>Główne klauzule instrukcji SELECT – FROM, WHERE, GROUP BY, HAVING, ORDER BY, JOIN.</p> <p>Przykład wykorzystania klauzuli JOIN do łączenia informacji z wielu tabel i kwerend oraz przedstawiania wyników jako jednego logicznego połączenia rekordów.</p> <p>Warunki wyszukiwania – operatory logiczne.</p> <p><b>Pojęcia:</b> SQL, klauzula.</p>

### 4.3. Realizacja projektu programistycznego, w tym przygotowanie dokumentów i raportów

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Poznanie pracy informatyków i organizacji pracy zespołowej.</p> <p>Umiejętność współdziałania w zespole, wypowiedzenia własnego zdania, ale również rozumienia i uwzględniania poglądów innych.</p> <p>Stosowanie zasady dokładnego wykonywania swojej części pracy zgodnie z wcześniejszymi ustaleniami.</p>	<p>Zasady projektowania systemów informatycznych.</p> <p>Etapy projektowania: analiza systemu informacyjnego, opracowanie założeń, wykonanie projektu technicznego, wykonanie projektu informatycznego, testowanie systemu, wdrożenie.</p> <p>Realizacja projektu na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania.</p> <p>Podział zadania na zadania szczegółowe.</p> <p>Zastosowanie szablonów (w dokumentach tekstowych) do przygotowywania założeń.</p> <p><b>Pojęcia:</b> analiza systemu, założenia, system informacyjny i informatyczny, projekt techniczny, szablon.</p>

## Zagadnienia nadprogramowe

Projektowanie bazy zawierającej więcej niż trzy tabele. Zaawansowane tworzenie kwerend z wykorzystaniem języka SQL. Przykłady kwerend złożonych. Przykłady innych narzędzi do projektowania systemów baz danych.

## 5. Wśród multimediiów

### 5.1. Przetwarzanie informacji w różnej postaci, m.in. graficznej i dźwiękowej

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Uświadomienie twórczego charakteru tworzenia grafiki komputerowej.</p> <p>Rozróżnianie grafiki wektorowej i rastrowej.</p> <p>Poznanie przykładowych możliwości programów przeznaczonych do edycji obrazu, tworzenia animacji i obróbki filmu.</p> <p>Umiejętność wykorzystywania multimediiów w projektowaniu stron internetowych i prezentacji multimedialnych.</p> <p>Rozumienie zasad łączenia poszczególnych elementów multimedialnych, np. obrazu z dźwiękiem.</p> <p>Stosowanie właściwych formatów do zapisu plików graficznych.</p>	<p>Przykłady urządzeń multimedialnych.</p> <p>Reprezentacja obrazu i dźwięku w komputerze.</p> <p>Możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo.</p> <p>Zasady działania animacji. Przykład tworzenia animacji.</p> <p>Źródła danych wideo. Przechwytywanie obrazu pochodzącego z różnych źródeł. Możliwości programów komputerowych w zakresie montażu wideo. Łączenie wideo, dźwięku, animacji i obrazów statycznych. Optymalizacja plików dla potrzeb Internetu.</p> <p>Grafika rastrowa a wektorowa. Praca z zaawansowanym programem do obróbki grafiki rastrowej.</p> <p>Narzędzia malarskie i korekcyjne. Tryby pracy narzędzi.</p> <p>Wybór fragmentów obrazu – praca z maskami. Fotorealizm.</p> <p>Metody definiowania barw i operacje na barwach.</p> <p>Przekształcenia obrazu – filtry i przekształcenia geometryczne.</p> <p>Tworzenie fotomontażu. Praca z warstwami obrazu.</p> <p>Retuszowanie. Formaty zapisu danych graficznych ze szczególnym uwzględnieniem Internetu.</p> <p><b>Pojęcia:</b> grafika rastrowa i wektorowa, maska, filtry, fotomontaż, warstwa obrazu, kanał barwny, histogram, krzywa barw; modele RGB, CMYK i HSV; sygnał wideo analogowy i cyfrowy; klatka obrazu (ramka), przechwytywanie (<i>capturing</i>); MPEG, AVI, DivX, FireWire.</p>

### 5.2. Tworzenie prezentacji multimedialnych

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Stosowanie poprawnych zasad tworzenia prezentacji.</p> <p>Rozróżnianie sposobów przygotowania prezentacji wspomagającej wystąpienie prelegenta od prezentacji do samodzielnego przeglądania przez odbiorcę.</p> <p>Umiejętność dostosowania rodzaju i formy prezentacji do jej tematyki i odbiorcy.</p> <p>Stosowanie zasad prezentowania pokazu slajdów.</p> <p>Stosowanie zasad prawnych aspektów korzystania z cudzych materiałów.</p>	<p>Zasady tworzenia prezentacji. Rodzaje prezentacji.</p> <p>Projektowanie prezentacji. Przygotowanie scenariusza, materiałów (tekstów, grafiki, dźwięku, animacji).</p> <p>Projektowanie układu slajdów. Animowanie slajdów.</p> <p>Metody przygotowania prezentacji wspomagającej wystąpienie prelegenta. Przygotowanie pokazu slajdów. Organizowanie prezentacji w widoku konspektu. Posługiwanie się widokiem sortowania slajdów. Odtwarzanie prezentacji. Zasady referowania konkretnego tematu wspomaganej prezentacją.</p> <p>Metody przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę. Nagrywanie narracji, zasady prezentowania.</p> <p>Testowanie chronometrażu. Hiperłącza.</p> <p>Dodawanie efektów multimedialnych: animacji, grafiki, dźwięków, podkładu muzycznego. Tworzenie schematu organizacyjnego.</p> <p>Możliwości tworzenia wykresów.</p> <p>Publikowanie prezentacji w Internecie.</p> <p>Dopasowywanie parametrów konwersji do formatu HTML.</p> <p><b>Pojęcia:</b> slajd, widok konspektu, widok sortowania slajdów, przejście slajdu, chronometraż.</p>

### 5.3. Tworzenie stron internetowych

Szczegółowe cele edukacyjne	Szczegółowe treści nauczania
<p>Dbłość o redakcyjną i merytoryczną poprawność oraz zgodność z prawem umieszczanych w Internecie tekstów i materiałów.</p> <p>Poznanie przykładowego programu do tworzenia stron internetowych.</p> <p>Poznanie nowoczesnych technologii dynamicznego generowania stron internetowych.</p>	<p>Metody tworzenia stron internetowych z wykorzystaniem nowoczesnych narzędzi.</p> <p>Planowanie zawartości strony (teksty, rysunki, dźwięki, animacje). Umieszczanie ww. elementów na stronie.</p> <p>Korzystanie z multimediów – Flash.</p> <p>Metody projektowania układu strony, ze szczególnym uwzględnieniem jej czytelności i przejrzystości.</p> <p>Posługiwanie się tabelami, ramkami i stylami dla efektywnego rozmieszczenia elementów strony.</p> <p>Dynamiczne strony HTML – wykorzystywanie JavaScriptu.</p> <p>Możliwości niektórych narzędzi (programów) do tworzenia stron oraz zasady publikowania w Internecie.</p> <p>Możliwości serwerów WWW wspomagające tworzenie dużych witryn internetowych – SSI (Server-Side Includes), języki skryptowe.</p>
	<p><b>Pojęcia:</b> HTTP, HTML, tabele, ramki (<i>frames</i>), style kaskadowe, Flash, JavaScript, Dynamic HTML, edytor HTML.</p>

#### Zagadnienia nadprogramowe

Skrypty po stronie serwera – CGI, PHP. Wykorzystywanie baz danych – MySQL. Oprogramowanie wspomagające tworzenie portali. Specyfikacja protokołu HTTP.

## IV. Uwagi o realizacji programu

Poniżej przedstawiono propozycje rozplanowania zajęć z technologii informacyjnej i informatyki w trzyletnim cyklu nauczania:

#### Propozycja I (6 godzin w cyklu nauczania)

Klasa I – 2 godz. technologii informacyjnej tygodniowo

Klasa II – 2 godz. informatyki tygodniowo

Klasa III – 2 godz. informatyki tygodniowo

#### Propozycja II (8 godzin w cyklu nauczania)

Klasa I – 2 godz. technologii informacyjnej tygodniowo

Klasa II – 3 godz. informatyki tygodniowo

Klasa III – 3 godz. informatyki tygodniowo

#### Propozycja III (7 godzin w cyklu nauczania)

Klasa I – 2 godz. technologii informacyjnej i 1 godz. informatyki tygodniowo

Klasa II – 3 godz. informatyki tygodniowo

Klasa III – 1 godz. informatyki tygodniowo (można zrealizować 2 godziny tygodniowo w pierwszym semestrze)



Pierwsza propozycja to rozwiązanie minimalne, natomiast propozycja druga stanowi rozwiązanie optymalne, aby zrealizować treści wynikające z podstaw programowych.

Propozycja III uwzględnia wprowadzenie już w pierwszej klasie informatyki w zakresie rozszerzonym. W tym przypadku należy jednak zadbać o to, aby zrealizować wcześniej zagadnienia podstawowe z zakresu technologii informacyjnej. Na przykład po zapoznaniu uczniów na lekcjach technologii informacyjnej z podstawowymi możliwościami wyszukiwania informacji w gotowych bazach danych, na informatyce można już omawiać zasady projektowania własnej bazy. Ta uwaga nie dotyczy algorytmiki i programowania, ponieważ zagadnień tych nie ma w podstawie programowej technologii informacyjnej.

## V. Przykładowy rozkład materiału

Przykładowy rozkład materiału do realizacji w pierwszym roku nauczania informatyki (propozycja I). Rozkład przygotowany jest w postaci 67 szczegółowych tematów. Jeden z tematów przeznaczony został do realizacji na dwóch godzinach lekcyjnych. Godziny, które ewentualnie pozostaną, należy przeznaczyć na utrwalanie wiadomości.

W drugim roku nauczania należy zaplanować zagadnienia związane z przetwarzaniem informacji w bazach danych, tworzeniem stron internetowych oraz zaawansowane zagadnienia z algorytmiki i programowania.

### Algorytmika i programowanie

- Temat 1. Związek między problemem, algorytmem a programem komputerowym
- Temat 2. Lista kroków algorytmu
- Temat 3. Graficzne prezentacje algorytmów
- Temat 4. Wykorzystanie oprogramowania edukacyjnego do graficznej prezentacji i analizy algorytmów
- Temat 5. Algorytm liniowy a algorytm z warunkami
- Temat 6. Algorytmy z warunkami zagnieżdżonymi
- Temat 7. Zapisywanie algorytmów w pseudojęzyku programowania
- Temat 8. Realizacja algorytmów z warunkami w postaci programu
- Temat 9. Iteracja – podstawowa technika algorytmiczna
- Temat 10. Iteracja w pseudojęzyku programowania
- Temat 11. Prezentacja algorytmów w arkuszu kalkulacyjnym
- Temat 12. Realizacja warunków zagnieżdżonych i iteracji w arkuszu kalkulacyjnym
- Temat 13. Klasyfikacja języków programowania
- Temat 14. Algorytmy z warunkami w Pascalu
- Temat 15. Realizacja algorytmów iteracyjnych w Pascalu
- Temat 16. Na czym polega programowanie strukturalne?
- Temat 17. Sposoby przekazywania parametrów
- Temat 18. Zastosowanie procedur i funkcji w zadaniach
- Temat 19. Grafika w języku programowania
- Temat 20. Animacje w języku programowania
- Temat 21. Dobór struktur danych do algorytmu
- Temat 22. Stosowanie struktur złożonych (tablic) w zadaniach
- Temat 23. Wprowadzanie danych do pliku i wyprowadzanie z pliku
- Temat 24. Stosowanie techniki iteracji w zadaniach – algorytm Euklidesa
- Temat 25. Metoda „dziel i zwyciężaj”
- Temat 26. Generowanie liczb Fibonacciego
- Temat 27. Praca z danymi tekstowymi
- Temat 28. Algorytmy sortowania
- Temat 29. Na czym polega rekurencja?



- Temat 30. Rekurencyjna realizacja silni
- Temat 31. Rekurencyjna realizacja algorytmu Euklidesa

### **Sieci komputerowe**

- Temat 32. Reprezentacja danych w komputerze
- Temat 33. Operacje arytmetyczne na liczbach binarnych i szesnastkowych
- Temat 34. Konwersje liczb z jednego systemu pozycyjnego na inny
- Temat 35. Kompresja i szyfrowanie danych
- Temat 36. Logiczny model komputera
- Temat 37. Najważniejsze funkcje systemu operacyjnego
- Temat 38. Przykłady systemów operacyjnych
- Temat 39. Podstawowe czynności wykonywane na plikach
- Temat 40. Wyszukiwanie plików i odzyskiwanie utraconych plików
- Temat 41. Ochrona plików
- Temat 42. Topologie i technologie sieciowe
- Temat 43. Urządzenia i elementy sieciowe
- Temat 44. Przegląd sieciowych systemów operacyjnych
- Temat 45. Realizacja prostych sieci komputerowych
- Temat 46. Konfiguracja komputerów sieciowych
- Temat 47. Bezpieczeństwo w sieci
- Temat 48. Ochrona i zabezpieczenia
- Temat 49. System Linux
- Temat 50. Zasady instalacji systemu Linux
- Temat 51. Serwery sieciowe Linuksa
- Temat 52. Internet – poszerzenie wiadomości
- Temat 53. Połączenia z Internetem
- Temat 54. Rozwój urządzeń komputerowych i sieci
- Temat 55. Historia komputerów
- Temat 56. Sieci przyszłości

### **Multimedia**

- Temat 57. Przegląd urządzeń multimedialnych
- Temat 58. Reprezentacja obrazu i dźwięku w komputerze
- Temat 59. Jak przygotować obraz animowany?
- Temat 60. Przykład montowania filmu w programach komputerowych
- Temat 61. Edycja obrazu
- Temat 62. Wybór fragmentów obrazu – praca z maskami
- Temat 63. Przekształcenia obrazu
- Temat 64. Fotomontaż obrazu
- Temat 65. Zasady tworzenia prezentacji
- Temat 66. Prezentacja jako wspomaganie wystąpienia prelegenta
- Temat 67. Prezentacja typu kiosk

## VI. Procedury osiągnięcia szczegółowych celów edukacyjnych – czynności nauczyciela i uczniów

### 1. Metody rozwiązywania problemów algorytmicznych

#### 1.1. Sposoby prezentacji algorytmów

Nauczyciel	Uczniowie
Omawia regulamin pracowni komputerowej oraz zasady bezpieczeństwa i higieny pracy przy komputerze.	Zapoznają się z regulaminem, zasadami pracy i bezpieczeństwa w pracowni komputerowej, a także stosują się do nich w czasie zajęć.
Porządkuje wiedzę dotyczącą algorytmiki, nabytą przez uczniów w gimnazjum. Wskazuje zależności między problemem, algorytmem a programem komputerowym (przedstawia program jako realizację algorytmu). Opisuje sposoby przedstawiania algorytmów i wykonuje z uczniami ćwiczenia w tym zakresie (na konkretnych przykładach). Zwraca uwagę na szczegółowe zasady zapisu algorytmów w różnych notacjach. Uzasadnia potrzebę wykonywania (nawet odręcznego) schematu postępowania podczas rozwiązywania problemów. Prezentuje oprogramowanie edukacyjne do konstrukcji schematów blokowych. Przygotowuje przykładowe zadania. Dyskutuje z uczniami o poprawności działania algorytmów wykonanych w postaci schematów. Na przykładzie pseudojęzyka omawia, na czym polega zapis algorytmu w postaci programu. Podaje przykłady instrukcji, zasady składni oraz omawia ogólną strukturę programu. Poleca powtórzenie zasad pracy w arkuszu kalkulacyjnym (tworzenie, kopiowanie formuł, stosowanie funkcji standardowych).	Przypominają wiadomości z gimnazjum dotyczące algorytmiki. Podają przykłady poznanych algorytmów, w tym z innych przedmiotów szkolnych (matematyki, fizyki) oraz znane im sposoby ich przedstawiania. Zastanawiają się, kiedy dowolny przepis postępowania może być algorytmem Określają specyfikację zadań. Zapisują algorytmy: <ul style="list-style-type: none"> <li>• w postaci listy kroków,</li> <li>• w postaci schematu blokowego,</li> <li>• w symbolicznym języku programowania.</li> </ul> Analizują poprawność budowy schematu blokowego. Realizują przykładowe algorytmy, korzystając z oprogramowania edukacyjnego – analizują przebieg algorytmu symulując działanie schematu. Tworzą proste programy w języku symbolicznym (pseudojęzyku). Realizują wybrane algorytmy w arkuszu kalkulacyjnym. Zainteresowani uczniowie zapoznają się z prezentacją algorytmu w postaci drzewa i z drzewem wyrażenia.

#### 1.2. Przegląd technik algorytmicznych i algorytmów klasycznych

Nauczyciel	Uczniowie
Określa różnicę między algorytmem liniowym a algorytmem z warunkami. Omawia na przykładach cechy tych algorytmów. Przygotowuje przykłady zadań, w których wystąpią sytuacje warunkowe (algorytmy z rozgałęzieniami). Wyjaśnia, jak w poszczególnych notacjach prezentować sytuacje warunkowe, w tym warunki zagnieżdżone. Opisuje i demonstrowa wykorzystanie techniki iteracji. Definiuje pojęcia: iteracja, pętla. Poleca zbudowanie schematu blokowego realizującego iterację, np. sumowanie $n$ liczb. Zwraca uwagę na stosowanie poprawnych sposobów zakończenia iteracji. Podaje je. Wyjaśnia zastosowanie tzw. licznika wykonanych kroków iteracji. Definiuje pojęcie zmiennej sterującej. Omawia realizację wybranych algorytmów iteracyjnych w arkuszu kalkulacyjnym. Na konkretnych przykładach wyjaśnia algorytmy znajdowania elementu najmniejszego (lub największego) w zbiorze. Omawia wybrane algorytmy sortowania, np. przez wybór i bąbelkowy. Wskazuje ich zastosowanie w porządkowaniu różnych zbiorów danych.	Podają przykłady algorytmów liniowych. Określają sytuacje warunkowe w znanych im algorytmach. Wskazują cechy tych algorytmów. Zapisują przykładowe algorytmy w wybranej notacji. Podają przykłady wykorzystania technik iteracyjnych w algorytmach. Budują schematy blokowe realizujące działania w pętli, stosują również pętle zagnieżdżone. Na podanych przykładach ćwiczą różne sposoby zakończenia iteracji. Określają liczbę kroków iteracji. Analizują przykładowe algorytmy iteracyjne w arkuszu kalkulacyjnym. Rozwiązują w arkuszu kalkulacyjnym wskazane zagadnienia. Zapisują algorytmy iteracyjne w pseudojęzyku. Poznają działanie przykładowej instrukcji iteracji. Poznają i wykorzystują w różnym zastosowaniu wybrane techniki porządkowania (sortowania) elementów. Korzystając z oprogramowania edukacyjnego, symulują i analizują działanie gotowych algorytmów sortowania. Konstruują samodzielnie schematy blokowe wybranych algorytmów sortowania.

Przygotowuje zadania, w których konieczne jest wykorzystanie poznanych technik algorytmicznych i algorytmów, np. techniki iteracji do wyboru elementu minimalnego w ciągu  $n$ -liczb czy algorytmu wyboru minimum do porządkowania (sortowania) elementów.

Zapoznaje uczniów z pojęciem rekurencji i podaje przykłady „zjawisk rekurencyjnych”; przygotowuje zadania do realizacji tą metodą. Wyjaśnia różnicę między iteracją i rekurencją. Podaje przykłady tego samego algorytmu zrealizowanego iteracyjnie i rekurencyjnie, np. silnia, algorytm Euklidesa, generowanie liczb Fibonacciego.

Zapoznają się z pojęciem rekurencji i odnajdują przykłady „zjawisk rekurencyjnych” w życiu codziennym oraz zadaniach szkolnych. Realizują wybrany algorytm rekurencyjnie. Porównują jego realizację iteracyjną z postacią rekurencyjną.

Zainteresowani uczniowie przedstawiają inne algorytmy rekurencyjne, np. wieże Hanoi, algorytm trwałych małżeństw oraz inne algorytmy sortowania (np. sortowanie szybkie).

### 1.3. Elementy analizy algorytmów

Nauczyciel	Uczniowie
<p>Omawia, na czym polega analiza problemu algorytmicznego. Przedstawia podział rozwiązania zadania (problemu) na logiczne etapy. Wyjaśnia, kiedy algorytm jest poprawny i uniwersalny.</p> <p>Wskazuje, jak wykorzystać oprogramowanie edukacyjne do analizy algorytmów.</p> <p>Definiuje pojęcie złożoności (pracochłonności) algorytmu. Omawia proste sposoby określania jego złożoności.</p> <p>Poleca uczniom, by porównali złożoność różnych rodzajów algorytmów, które realizują to samo zadanie (dla tych samych danych).</p> <p>Zainteresowanym uczniom proponuje dodatkową literaturę z dziedziny algorytmiki, m.in. na temat analizy algorytmów.</p>	<p>Analizują problem algorytmiczny, omawiają etapy jego rozwiązania.</p> <p>Sprawdzają, czy dany algorytm działa zgodnie ze specyfikacją zadania – oceniają poprawność działania algorytmu.</p> <p>Analizują (śledzą) działanie algorytmu zapisanego w wybranej notacji dla przykładowych danych, np. „krok po kroku” śledzą iterację i wypisują kolejne wartości zmiennych po zrealizowaniu każdego kroku iteracji.</p> <p>Określają liczbę operacji wykonywanych na danych, np. liczbę wykonanych porównań elementów w algorytmie bąbelkowym (złożoność obliczeniową algorytmu).</p> <p>Porównują złożoność różnych algorytmów tego samego zadania (dla tych samych danych), np. dwóch algorytmów sortowania.</p>

## 2. Realizacja algorytmów w wybranym języku programowania

### 2.1. Zasady programowania

Nauczyciel	Uczniowie
<p>Wyjaśnia, czym jest język programowania, podaje jego cechy, porównując je z cechami języka naturalnego. Klasyfikuje języki programowania według różnych kryteriów.</p> <p>Na przykładzie pseudojęzyka programowania oraz wybranego języka wysokiego poziomu wyjaśnia, na czym polega programowanie oraz struktura programu. Omawia zasady działania, składnię podstawowych instrukcji, jak też zasady poprawnego programowania.</p> <p>Wyjaśnia pojęcia programu źródłowego, programu wynikowego, kompilacji, translacji, interpretacji.</p> <p>Na przykładowym programie przedstawia etapy programowania – implementację, kompilację, uruchomienie, testowanie.</p> <p>Opisuje i analizuje procesy zachodzące w czasie uruchamiania programów. Określa rolę procesora i pamięci operacyjnej w realizowaniu instrukcji programowych.</p>	<p>Dzieli się swoją wiedzą na temat języków programowania. Uzupełniają klasyfikację języków programowania, wymieniając znane sobie języki.</p> <p>Kompilują i uruchamiają gotowy (np. przygotowany przez nauczyciela) program zapisany w języku wysokiego poziomu. Zapoznają się z podstawową strukturą programu.</p> <p>Uczniowie, którzy tworzyli już własne programy, prezentują je na forum klasy.</p> <p>Zapoznają się ze sposobem zapisu programu źródłowego w języku wysokiego poziomu (np. z edytorem języka Turbo Pascal).</p>
<p>Poleca zrealizować w postaci programu prosty algorytm liniowy. Zwraca uwagę na łędy kompilacji i błędy wykonania.</p> <p>Wyjaśnia pojęcie zmiennej i sposób deklaracji zmiennych w programie, a także metody wprowadzania danych do programów i wyprowadzania wyników (procedury wejścia/wyjścia, instrukcja przypisania).</p>	<p>Zapisują algorytm liniowy w wybranym języku programowania, stosując poznane zasady składni tego języka.</p> <p>Uczą się poprawiać błędy kompilacji. Odróżniają błędy kompilacji od błędów wykonania. Wykonują program i testują go, podstawiąjąc różne dane.</p>

<p>Omawia zasady zapisu algorytmów z warunkami i algorytmów iteracyjnych w postaci programów, w tym warunki i pętle zagnieżdżone. Poleca, by uczniowie przypomnieli sobie techniki iteracji oraz sposoby jej graficznej prezentacji w postaci schematu blokowego. Zwraca uwagę na poprawne określanie zakończenia iteracji w programie warunków, aby uniknąć jego „zapętlenia”.</p>	<p>Na konkretnych przykładach poznają działanie instrukcji warunkowych i iteracyjnych w wybranym języku programowania wysokiego poziomu. Ćwiczą zapis (w postaci programów) algorytmów zawierających sytuacje warunkowe. Zapisują algorytmy iteracyjne w języku programowania.</p>
<p>Wspólnie z uczniami formułuje zasady zapisu strukturalnego (z wykorzystaniem procedur i funkcji) oraz omawia możliwości jego zastosowania. Określa metodę programowania zstępującego. Przygotowuje zadania, które łatwiej rozwiązać techniką programowania strukturalnego. Podkreśla możliwość wielokrotnego wykorzystania tej samej procedury w różnych segmentach (częściach) programu oraz w innych programach.</p>	<p>Projektują rozwiązanie zadań techniką programowania strukturalnego. Rozważają podział algorytmu na niezależne fragmenty, realizujące określone operacje. Analizują możliwość wielokrotnego wykorzystania zadeklarowanych procedur w tym samym programie, jak również w innych. Zapisują wybrane algorytmy klasyczne w języku programowania wysokiego poziomu.</p>
<p>Wyjaśnia na konkretnym algorytmie (np. obliczania silni), na czym polega zapis algorytmu rekurencyjnego w postaci programu. Wskazuje różnicę między zastosowaniem instrukcji iteracyjnych a zapisem procedur rekurencyjnych.</p> <p>Omawia wykorzystanie podstawowych procedur graficznych do programowania grafiki. Na konkretnych przykładach prezentuje elementy grafiki animowanej.</p> <p>Posługując się przykładami, wyjaśnia zasady zapisu algorytmu sortowania (np. sortowanie bąbelkowe). Poleca zdefiniowanie potrzebnych procedur.</p>	<p>Projektują i realizują w języku programowania wysokiego poziomu wybrane algorytmy rekurencyjne. Zapisują wybrane algorytmy, które zrealizowali iteracyjnie w sposób rekurencyjny, np. algorytm generowania liczb Fibonacciego.</p> <p>Projektują rysunki, wykorzystując procedury graficzne języka programowania. Programują proste gry zawierające animacje. Zainteresowani uczniowie wykorzystują rekurencję do tworzenia grafiki komputerowej.</p> <p>Zapisują wybrany algorytm sortowania, np. algorytm bąbelkowy, w postaci programu.</p>
<p>Omawia istotę programowania zorientowanego obiektowo. Wskazuje różnice w stosunku do programowania strukturalnego. Wyjaśnia pojęcie obiektu. Na przykładzie gotowego programu określa, na czym polega programowanie zdarzeniowe. Zainteresowanym uczniom poleca dodatkową literaturę dotyczącą zaawansowanych metod programowania i proponuje trudniejsze zadania.</p>	<p>Piszą (fakultatywnie) proste programy w wybranym, obiektowym lub zdarzeniowym języku programowania. Porównują zapisy algorytmu iteracyjnego z programem zapisanym w języku strukturalnym. Dyskutują na temat podobieństw i różnic w strukturze programu zapisanego w różnych językach, na temat deklaracji zmiennych, procedur (funkcji), składni i zasad działania podstawowych instrukcji.</p> <p>Wykonują wspólny projekt programistyczny, stosując zasady pracy zespołowej.</p>

## 2.2 Dobór struktur danych do rozwiązywanego problemu

Nauczyciel	Uczniowie
<p>Wyjaśnia pojęcie struktury danych. Przedstawia podział struktur danych. Wyjaśnia różnicę między typem prostym a strukturalnym. Wyjaśnia, jakie są konsekwencje deklaracji zmiennych określonego typu ze względu na zajętość pamięci komputera.</p> <p>Na kilku przykładach pokazuje, jak dobrać odpowiednie struktury danych do rozwiązania konkretnych zadań, m.in. poprzez zastosowanie typu łańcuchowego, rekordowego i tablicowego.</p>	<p>Zapoznają się z klasyfikacją struktur danych. Podczas programowania zwracają uwagę na stosowanie właściwych struktur danych. Szacują wielkość pamięci zajmowanej przez dane określonego typu.</p> <p>Stosują typy proste w zadaniach i uczą się za ich pomocą określać złożone struktury danych, np. deklarują tablicę składającą się z liczb całkowitych. Dobierają typy danych do zadań, np. stosują typ tablicowy w zapisie algorytmów sortowania.</p> <p>Deklarują typ łańcuchowy (lub znakowy) w zadaniach na danych tekstowych. W celu tworzenia baz danych definiują typ rekordowy. Na prostych przykładach zapoznają się z definiowaniem typu obiektowego.</p>
<p>Zainteresowanym uczniom poleca zadania, w których wymagane jest zastosowanie dynamicznych struktur danych – typów wskaźnikowych.</p>	<p>Zainteresowani uczniowie, korzystając z dodatkowej literatury, tworzą programy z zastosowaniem typów wskaźnikowych.</p>

### 3. Zasady działania komputera i sieci komputerowych

#### 3.1. System komputerowy

Nauczyciel	Uczniowie
<p>Przedstawia sposób prezentacji liczb w komputerze. Odwołując się do systemu dziesiętnego, omawia istotne dla informatyki systemy pozycyjne (system dwójkowy i szesnastkowy). Przedstawia algorytm konwersji liczb z systemu binarnego na dziesiętny i odwrotnie. Poleca, by uczniowie przeanalizowali gotowy schemat blokowy tego algorytmu. Omawia sposób zapisu liczb ujemnych oraz rzeczywistych (zmiennoprzecinkowych). Wyjaśnia, czym jest kod ASCII.</p>	<p>Wykonują zamianę liczb dziesiętnych z systemu dziesiętnego na binarny i odwrotnie oraz podstawowe działania arytmetyczne na liczbach binarnych (dodawanie i odejmowanie). Opcjonalnie – wykonują operacje logiczne i przesunięcia bitowe. Poznają system szesnastkowy i wykonują konwersję liczb binarnych na liczby w systemie szesnastkowym. Zapisują algorytm zamiany liczb z systemu dwójkowego na dziesiętny i odwrotnie w postaci programu komputerowego. Zainteresowani zapisują algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny.</p> <p>Dzieli się własną wiedzą na temat kompresji i szyfrowania danych, sięgając do historii kryptografii. Wykonują ćwiczenia, w których szyfrują i odszyfrowują podane słowa, jak też całe teksty. Zainteresowani uczniowie zapisują algorytm szyfrowania w postaci programu.</p>
<p>Dyskutuje z uczniami na temat kompresji i szyfrowania danych. Omawia rodzaje kompresji i ich zastosowanie. Definiuje pojęcie współczynnika kompresji. Prezentuje wybrane algorytmy szyfrowania. Omawia praktyczne zastosowanie szyfrowania – wskazuje gotowe programy. Zadaje ćwiczenia do wykonania.</p> <p>Poleca przypomnienie (z lekcji informatyki w gimnazjum i lekcji technologii informacyjnej) podstawowych informacji o systemie komputerowym. Prezentuje na schemacie logiczny model komputera. Wyjaśnia działanie procesora.</p> <p>Zwraca uwagę na charakterystyczne cechy i funkcje różnych systemów operacyjnych. Wskazuje na ich różnorodne zastosowanie.</p> <p>Poleca powtórzenie wiadomości o zarządzaniu folderami i plikami oraz o zasadach ich ochrony. Prezentuje podobieństwa i różnice w tym zakresie między różnymi systemami operacyjnymi.</p>	<p>Prezentują ogólną klasyfikację środków (urządzeń) i narzędzi (oprogramowania) technologii informacyjnej. Omawiają części składowe komputera – ich przeznaczenie i parametry – oraz organizację pamięci komputera. W Internecie i innych źródłach wyszukują informacje na temat różnych typów aktualnie używanych komputerów, w tym wiadomości o superkomputerach.</p> <p>Przypominają – z lekcji informatyki w gimnazjum i lekcji technologii informacyjnej – poznane funkcje systemu operacyjnego. Poznają inne systemy i ich najważniejsze funkcje.</p> <p>Wykonują ćwiczenia, które porządkują ich wiedzę i umiejętności z zakresu podstawowych operacji wykonywanych na plikach. Poznają nowe operacje, m.in. nadawanie plikom atrybutów, wyszukiwanie plików, odzyskiwanie utraconych plików. Ćwiczenia wykonują na własnych plikach. Dyskutują nad potrzebą ochrony plików.</p>

#### 3.2. Sieci komputerowe, w tym Internet

Nauczyciel	Uczniowie
<p>Poleca przypomnienie zagadnień dotyczących sieci komputerowych z lekcji technologii informacyjnej. Prezentuje na schemacie działanie sieci komputerowej. Definiuje pojęcie topologii i przedstawia różne rodzaje topologii. Omawia urządzenia i elementy sieciowe oraz zasady pracy w sieci. Zwraca uwagę na te elementy i ich parametry, które dotyczą pracowni.</p> <p>Omawia zasady konfiguracji prostych sieci komputerowych. Przygotowuje (jeśli są ku temu możliwości) niezbędne składniki, aby można było w warunkach pracowni szkolnej zbudować sieć domową, składającą się np. z dwóch-trzech komputerów. Dokonuje przeglądu sieciowych systemów operacyjnych. Bardziej szczegółowo omawia system Linux.</p> <p>Zwraca uwagę na zagrożenia płynące z sieci, omawia przykładowe oprogramowanie do ochrony własnych zasobów, m.in. zapory sieciowe.</p> <p>Poleca uczniom, by z lekcji technologii informacyjnej przypomnieli sobie zasady przepływu pakietów danych w Internecie. Rozszerza wiadomości na temat protokołu TCP/IP. Zwraca uwagę na nowe możliwości połączeń z Internetem.</p>	<p>Dyskutują nad korzyściami płynącymi z łączenia komputerów w sieć. Klasyfikują sieci ze względu na wielkość, omawiają rodzaj i topologię sieci działającej w szkolnej pracowni komputerowej. Zastanawiają się nad zaletami i wadami różnych topologii. Wykonują ćwiczenia związane z udostępnianiem zasobów itp., poznając w ten sposób zasady wymiany informacji w sieci komputerowej.</p> <p>Dzieli się swoimi doświadczeniami w zakresie budowania małych sieci domowych. Realizują w praktyce małą sieć komputerową – konfigurują składniki sieci, udostępniają pliki, dysk, drukarki, dodają użytkowników.</p> <p>Poznają możliwości różnych systemów operacyjnych, w tym systemu Linux.</p> <p>Zapoznają się z zagrożeniami płynącymi z sieci i sposobami konfigurowania przykładowego oprogramowania do ochrony zasobów.</p> <p>Dyskutują nad różnymi sposobami połączeń z Internetem.</p>

### 3.3. Tendencje w rozwoju informatyki i jej zastosowań

Nauczyciel	Uczniowie
<p>Zwraca uwagę na zalety i zagrożenia, wynikające z wykorzystywania nowoczesnych technologii.</p> <p>Proponuje, by uczniowie opracowali samodzielnie zagadnienia dotyczące historii i najnowszych osiągnięć w dziedzinie informatyki, ze szczególnym uwzględnieniem wpływu rozwoju informatyki na zachowania społeczne i gospodarkę kraju.</p> <p>Uświadamia wagę prawnych i społecznych aspektów różnorodnego zastosowania informatyki. Zwraca uwagę na zgodne z zasadami etyki zachowanie się w sieci.</p>	<p>Wyszukują w Internecie i innych źródłach informacje na temat nowych możliwości usług internetowych i nowoczesnych zastosowań urządzeń komputerowych.</p> <p>Przygotowują referaty (np. w formie prezentacji multimedialnej) na wybrane tematy: historia komputerów, pierwsze języki programowania, najnowsze pomysły na komputery, sztuczna inteligencja, sieci przyszłości, rozwój multimediiów.</p> <p>Dyskutują nad tendencjami rozwoju informatyki i jej zastosowań, dostrzegając przeobrażenia w tej dziedzinie w kraju i na świecie.</p> <p>Systematyzują wiedzę na temat prawnych i społecznych aspektów stosowania informatyki oraz stosują ją w praktyce. Przypominają zasady netykiety.</p>

## 4. Przetwarzanie danych w bazach danych

### 4.1. Projektowanie relacyjnej bazy danych

Nauczyciel	Uczniowie
<p>Poleca uczniom, by przypomnieli sobie z lekcji technologii informacyjnej metody przetwarzania danych.</p> <p>Na przykładzie gotowej bazy danych wyjaśnia dokładnie, na czym polega przetwarzanie danych.</p> <p>Poleca wykonanie podstawowych operacji na danych (wprowadzanie, redagowanie, sortowanie, indeksowanie, wyszukiwanie, prezentacja).</p> <p>Rozróżnia mechanizmy porządkowania informacji w bazie danych: sortowanie i indeksowanie. Wyjaśnia pojęcie indeksu.</p> <p>Tłumaczy pojęcie relacji w bazach danych (porównuje je z pojęciem relacji w matematyce).</p> <p>Wyjaśnia rolę klucza podstawowego.</p> <p>Omawia główne rodzaje formularzy i raportów.</p> <p>Poleca przypomnienie sposobów wykorzystywania informacji z baz danych w innych dokumentach, np. w tworzeniu korespondencji seryjnej czy etykiet adresowych.</p>	<p>Wymieniają podstawowe obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, zakładów pracy, instytucji naukowych i gospodarczych. Podają przykłady programów do tworzenia baz danych.</p> <p>Wykonując ćwiczenia na gotowych tabelach, przypominają zasady organizacji informacji w bazach danych oraz podstawowe pojęcia: tabela, rekord, pole, typ pola, formularz, kwerenda, raport. Sprawdzają na konkretnym przykładzie, na czym polega indeksowanie bazy i czym różni się od sortowania.</p> <p>Wybierają własny temat bazy lub polecony przez nauczyciela. Projektują samodzielnie relacyjną bazę danych, składającą się z dwóch (ewentualnie trzech) tabel połączonych relacją. Definiują klucz podstawowy. Określają relacje. Przygotowują odpowiednie kwerendy wybierające. Projektują formularze i raporty. Drukują elementy bazy danych (raporty, formularze, tabele).</p> <p>Przypominają z lekcji technologii informacyjnej w gimnazjum zasady tworzenia korespondencji seryjnej i etykiet adresowych. Ćwiczą łączenie informacji z bazy danych z dokumentami tekstowymi.</p>

### 4.2. Wyszukiwanie informacji z użyciem języka zapytań

Nauczyciel	Uczniowie
<p>Systematyzuje (wspólnie z uczniami) poznane zagadnienia dotyczące wyszukiwania informacji w różnych bazach danych i z wykorzystaniem rozmaitych narzędzi.</p> <p>Omawia podstawy języka SQL – zasady składni, podstawowe instrukcje.</p> <p>Poleca przejrzanie kodu SQL we wcześniej tworzonych kwerendach, np. za pomocą programu MS Access.</p> <p>Przygotowuje ćwiczenia polegające na wyszukiwaniu różnych informacji z bazy.</p> <p>Poleca dodatkową literaturę oraz nadzoruje pracę uczniów uzdolnionych i zainteresowanych zaawansowanym przetwarzaniem danych z pomocą innych narzędzi.</p>	<p>Wykonują ćwiczenia powtórzeniowe, wyszukując informacje na różne sposoby. Porównują wszystkie te sposoby.</p> <p>Tworzą kwerendy z wykorzystaniem języka SQL.</p> <p>Zapoznają się ze składnią i działaniem podstawowych instrukcji. Porównują je do innych języków programowania.</p> <p>Stosują instrukcję SELECT i jej główne klauzule, by wybrać kolumny z tabel bazy danych. Wykorzystują klauzulę JOIN do łączenia informacji z wielu tabel i kwerend oraz przedstawiania wyników jako jednego logicznego połączenia rekordów.</p> <p>Zainteresowani uczniowie projektują bazy zawierające więcej niż trzy tabele. Zapoznają się samodzielnie z zaawansowanym tworzeniem kwerend z wykorzystaniem języka SQL. Poznają inne narzędzia do projektowania i tworzenia baz danych.</p>



### 4.3. Realizacja projektu programistycznego, w tym przygotowanie dokumentów i raportów

Nauczyciel	Uczniowie
<p>Omawia, na czym polega praca informatyków (projektantów systemów informatycznych). Wyjaśnia zasady projektowania systemów informatycznych. Szczegółowo prezentuje etapy projektowania.</p> <p>Zwraca uwagę, by każdy z uczniów dokładnie wykonywał swoją część pracy, zgodnie z wcześniejszymi ustaleniami. Nadzoruje przebieg projektu.</p> <p>Podsumowuje (wspólnie z uczniami) wykonanie projektu. Zwraca uwagę na takie umiejętności uczniów jak: współdziałanie w zespole, wypowiedzianie własnego zdania, rozumienie i uwzględnianie poglądów innych, dochodzenie do kompromisu.</p>	<p>Przygotowują się do realizacji projektu grupowego. Planują temat projektu. Realizują go zgodnie z zamierzoną organizacją pracy zespołowej i wytyczonymi wcześniej etapami projektowania.</p> <p>Wybierają koordynatora grupy, który wspólnie z nauczycielem wyznacza zespoły do realizacji poszczególnych zadań. Przeprowadzają analizę systemu informacyjnego, opracowują założenia, wykonują projekt techniczny oraz informatyczny, testują i wdrażają system. Przygotowując założenia w postaci dokumentów edytora tekstu, korzystają z możliwości szablonów.</p>

## 5. Wśród multimediiów

### 5.1. Przetwarzanie informacji w różnych postaciach, m.in. graficznej i dźwiękowej

Nauczyciel	Uczniowie
<p>Porządkuje wiedzę uczniów na temat multimediiów, nabytą w gimnazjum na lekcjach informatyki i technologii informacyjnej. Na przykładach przedstawia ogólne możliwości wybranych programów przeznaczonych do edycji obrazu i dźwięku, do tworzenia animacji i obróbki filmów.</p> <p>Poleca, by uczniowie przypomnieli sobie sposób prezentacji danych liczbowych w komputerze. Omawia, jak w komputerze reprezentowany jest obraz. Wyjaśnia różnice między grafiką rastrową a wektorową.</p> <p>Omawia podstawy pracy z użyciem wybranego programu graficznego.</p>	<p>Podają znane z technologii informacyjnej przykłady urządzeń multimedialnych. Klasyfikują je pod względem ich przeznaczenia. Określają pojęcie komputer multimedialny. Dyskutują nad możliwościami komputera w zakresie edycji obrazu, dźwięku, animacji i wideo.</p> <p>Przypominają odpowiednie treści nauczania z lekcji technologii informacyjnej.</p> <p>Wykonując konkretne ćwiczenia poznają zaawansowaną obróbkę grafiki rastrowej. Stosują różne narzędzia malarskie i korekcyjne oraz wybierają tryb pracy narzędzi.</p>
<p>Wyjaśnia, na czym polega selekcja fragmentów obrazu i praca z maskami. Objaśnia pojęcie fotorealizmu. Pokazuje na przykładzie, w jaki sposób definiować i przekształcać barwy oraz jak wykonywać przekształcenia geometryczne i stosować filtry.</p> <p>Omawia tworzenie fotomontaży i sposoby wykonywania retuszu.</p> <p>Omawia zasady przygotowania grafiki w celu wykorzystania jej w projektowaniu stron internetowych lub prezentacji multimedialnych.</p> <p>Zwraca uwagę na stosowanie właściwych formatów do zapisu plików graficznych, ze szczególnym uwzględnieniem Internetu.</p>	<p>Ćwiczą pracę z maskami (wybierają fragmenty obrazu). Definiują barwy i wykonują różne operacje na barwach. Ćwiczą przekształcenia obrazu (geometryczne i filtry).</p> <p>Tworzą przykładowe fotomontaże. Pracują z warstwami obrazu. Retuszują obraz.</p> <p>Przygotowują grafikę do własnej strony internetowej lub prezentacji multimedialnej. Optymalizują pliki dla konkretnych potrzeb.</p> <p>Przypominają z technologii informacyjnej formaty plików graficznych (zastosowania, zalety i wady).</p>
<p>Pokazuje, jak w komputerze reprezentowany jest dźwięk. Wyjaśnia zasady działania animacji. Poleca utworzenie prostej animacji w dostępnej aplikacji. Wyjaśnia zasady łączenia poszczególnych elementów multimedialnych, np. obrazu z dźwiękiem.</p> <p>Wskazuje możliwości programów komputerowych w zakresie montażu wideo oraz konwersji pomiędzy formatami wideo.</p>	<p>Próbują przygotować własną animację. Łączą wideo, dźwięk, animację i obraz statyczny, np. w programie do animacji (Flash) lub programie do obróbki wideo.</p> <p>Poznają metody przechwytywania danych wideo. Poznają zasady montażu filmu. Poznają sposoby udostępnienia filmu innym osobom oraz metody konwersji między różnymi formatami. Opcjonalnie (zainteresowani): wykonują montaż filmu.</p>

## 5.2. Tworzenie prezentacji multimedialnych

Nauczyciel	Uczniowie
<p>Przedstawia na przykładzie gotowej, poprawnie wykonanej prezentacji, zasady tworzenia prezentacji multimedialnych. Wyjaśnia, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Poleca przypomnienie wiadomości o zasadach tworzenia prezentacji z lekcji technologii informacyjnej.</p> <p>Wyjaśnia różnicę w przygotowaniu prezentacji wspomagającej wystąpienie prelegenta od prezentacji do samodzielnego przeglądania przez odbiorcę w dogodnym dla niego czasie lub samouruchamiającej się bez ingerencji odbiorcy. Zwraca uwagę na różnice w posługiwaniu się Kreatorem zawartości, szablonem projektu a prezentacją pustą.</p> <p>Poleca przygotować konkretne tematy do wykonania prezentacji wspomagającej wystąpienie prelegenta. Zaleca przygotowanie scenariusza prezentacji. Omawia zasady prezentowania pokazu slajdów.</p> <p>Proponuje przygotowanie w formie projektu grupowego prezentacji do samodzielnego przeglądania przez odbiorcę. Proponuje przygotowanie w formie projektu grupowego prezentacji samouruchamiającej się.</p> <p>Omawia sposoby umieszczania prezentacji w Internecie.</p>	<p>Przeglądają przykładowe prezentacje, korzystając z plików znajdujących się na CD dla ucznia. Opcjonalnie wyświetlają własne prezentacje, tworzone podczas zajęć technologii informacyjnej.</p> <p>Projektują własne prezentacje, które mają wspomagać ich wystąpienia. Dobierają temat. Przygotowują scenariusz, wyszukują i gromadzą materiały (teksty, obrazy, dźwięki i animacje). Planują układ slajdów i ich animację.</p> <p>Przygotowują pokaz slajdów. Odtwarzają prezentację.</p> <p>Poznają metody przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.</p> <p>Wybierają koordynatora grupy, który przydzieli zadania szczegółowe poszczególnym uczniom. Nagrywają narrację, testują chronometraż i przygotowują hiperłącza.</p> <p>Publikują prezentację w Internecie.</p>

## 5.3. Tworzenie stron internetowych

Nauczyciel	Uczniowie
<p>Poleca powtórzenie materiału o tworzeniu stron internetowych z technologii informacyjnej, ze szczególnym uwzględnieniem języka HTML.</p> <p>Wspólnie z uczniami klasyfikuje narzędzia (oprogramowanie) do tworzenia stron.</p> <p>Omawia zaawansowane możliwości języka HTML: tabele, ramki, style.</p> <p>Omawia podstawy języka JavaScript. Wskazuje na podobieństwa między językami JavaScript i Pascal. Pokazuje, jak korzystać ze skryptów.</p>	<p>Przypominają podstawy języka znaczników HTML, wykonując nieskomplikowaną stronę na zadany przez nauczyciela temat.</p> <p>Uczniowie, którzy posiadają już własne strony internetowe, prezentują je pozostałym kolegom z grupy. Przedstawiają także narzędzia i metody, jakimi posługiwali się przy ich wykonaniu.</p> <p>Ćwiczą wykorzystywanie nowo poznanych funkcji języka HTML. Wykorzystują je do udoskonalenia własnych, istniejących już stron.</p> <p>Używają języka JavaScript, by osiągnąć takie efekty jak: dynamicznie tworzona treść, podświetlanie przycisków, zarządzanie oknami przeglądarki.</p>
<p>Podaje przykładowe tematy stron internetowych lub poleca wskazanie własnego tematu.</p> <p>Omawia reguły poprawnego projektowania układu strony, m.in. zachowanie jej czytelności i przejrzystości, jak też dbałość o poprawność redakcyjną, merytoryczną oraz prawą umieszczanych na niej tekstów i materiałów. Wskazuje jeden przykładowy program do projektowania i tworzenia stron. Omawia jego możliwości.</p> <p>Przybliża metody publikowania stron w Internecie. Określa zalety i wady poszczególnych metod.</p> <p>Omawia możliwości serwerów WWW wspomagających tworzenie dużych witryn internetowych – SSI (Server-Side Includes). Podaje przykłady języków skryptowych (serwerowych).</p>	<p>Wybierają i przygotowują własne tematy stron internetowych. Planują zawartość strony (teksty, rysunki, dźwięki, animacje). Umieszczają na niej ww. elementy.</p> <p>Pracując w grupie, wykonują własną stronę. Zapoznają się z przykładowym programem do tworzenia stron internetowych. Posługują się tabelami, ramkami i stylami dla efektownego rozmieszczenia elementów strony.</p> <p>Włączają na stronie licznik odwiedzin. Dodają inne typowe elementy, takie jak forum i księga gości.</p> <p>Dyskutują nad możliwościami publikowania stron w Internecie. Przypominają te zagadnienia z lekcji technologii informacyjnej.</p> <p>Uczniowie zainteresowani poznają wykorzystanie baz danych – MySQL w połączeniu z językami skryptowymi.</p>



## VII. Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne

**Uwaga:** zakładamy, że osiągnięcia ucznia z informatyki uzupełniają jego wiedzę i umiejętności z lekcji technologii informacyjnej.

### 1. Metody rozwiązywania problemów algorytmicznych

#### 1.1. Sposoby prezentacji algorytmów

2	3	4	5	6
<p>Wie, co to jest algorytm. Określa dane do zadania oraz wyniki.</p> <p>Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym.</p> <p>Potrafi narysować (od ręcznie) schemat blokowy algorytmu liniowego.</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm w postaci listy kroków.</p> <p>Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu.</p> <p>Korzysta (w stopniu podstawowym) z programu edukacyjnego do symulacji działania algorytmu skonstruowanego w postaci schematu blokowego.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym.</p> <p>Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów. Przedstawia dokładną specyfikację dowolnego zadania.</p> <p>Zna znaczenie i działanie instrukcji symbolicznego języka programowania (pseudojęzyka).</p> <p>Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci.</p> <p>Potrafi skonstruować algorytm z warunkami zagnieżdżonymi i pętlą za pomocą programu edukacyjnego.</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji), m.in. w pseudojęzyku.</p> <p>Zapisuje algorytmy z pętlą zagnieżdżoną.</p> <p>Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego.</p> <p>Analizuje działanie algorytmu dla przykładowych danych.</p> <p>Stosuje swobodnie oprogramowanie edukacyjne do graficznej prezentacji i analizy algorytmów.</p>	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji).</p> <p>Potrafi trafnie dobrać do algorytmu sposób prezentacji.</p> <p>Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia.</p> <p>Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych.</p> <p>Potrafi zaproponować własny pseudojęzyk (postać instrukcji i zasady składni).</p>

#### 1.2. Przegląd technik algorytmicznych i algorytmów klasycznych

2	3	4	5	6
<p>Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe.</p> <p>Wie, na czym polega powtarzanie tych samych operacji.</p> <p>Potrafi omówić, na przykładzie, algorytm znajdowania najmniejszego z trzech elementów.</p>	<p>Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami).</p> <p>Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń.</p>	<p>Analizuje algorytmy, w których występują powtórzenia (iteracje). Zna sposoby zakończenia iteracji.</p> <p>Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania <math>n</math> liczb, algorytm obliczania silni, znajdowania minimum w ciągu <math>n</math> liczb, algorytm rozwiązywania równania liniowego.</p>	<p>Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji.</p> <p>Zna inne algorytmy sortowania, np. kubełkowe, przez wstawianie.</p> <p>Zna przynajmniej jeden algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego.</p>	<p>Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję.</p> <p>Zna trudniejsze algorytmy, np. algorytm trwałego małżeństwa, wieże Hanoi, problem ośmiu hetmanów.</p>

	<p>Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy.</p> <p>Omawia i analizuje wybrane techniki sortowania w postaci gotowych schematów blokowych, skonstruowanych w programie edukacyjnym.</p>	<p>Zna iteracyjną postać algorytmu Euklidesa. Zna przynajmniej dwie techniki sortowania, np. bąbelkowe i przez wybór.</p> <p>Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Zna rekurencyjną realizację wybranego algorytmu, np. silni.</p>	<p>Wskazuje różnicę między rekurencją a iteracją.</p> <p>Zna rekurencyjną realizację wybranych algorytmów, np. silnię i algorytm Euklidesa.</p> <p>Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną.</p>	<p>Zna inne techniki sortowania, np. sortowanie przez scalanie ciągów i metodę szybką.</p> <p>Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu).</p> <p>Zna inne algorytmy numeryczne, np. wyznaczanie miejsca zerowego funkcji.</p> <p>Korzysta samodzielnie z dodatkowej literatury.</p>
--	---	---	---	--

### 1.3. Elementy analizy algorytmów

2	3	4	5	6
<p>Potrafi zanalizować przebieg algorytmu dla przykładowych danych i ocenić w ten sposób jego poprawność.</p>	<p>Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie.</p>	<p>Rozumie, co to jest złożoność algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania.</p>	<p>Potrafi porównać złożoność różnych algorytmów tego samego zadania dla tych samych danych.</p> <p>Wie, kiedy algorytm jest uniwersalny.</p>	<p>Ocenia złożoność czasową i pamięciową algorytmu. Zna odpowiednie wzory.</p>

## 2. Realizacja algorytmów w wybranym języku programowania

### 2.1. Zasady programowania

2	3	4	5	6
<p>Zna klasyfikację języków programowania.</p> <p>Zna ogólną budowę programu i najważniejsze elementy języka – słowa kluczowe, instrukcje, wyrażenia, zasady składni.</p> <p>Potrafi zrealizować prosty algorytm liniowy w języku wysokiego poziomu; potrafi skompilować i uruchomić gotowy program.</p>	<p>Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze.</p> <p>Rozumie pojęcia: implementacja, kompilacja, uruchomienie, testowanie.</p> <p>Rozumie znaczenie i działanie podstawowych instrukcji wybranego języka programowania wysokiego poziomu.</p> <p>Rozróżnia i poprawia błędy kompilacji i błędy wykonania.</p> <p>Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu.</p>	<p>Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna rekurencyjne realizacje prostych algorytmów.</p> <p>Rozumie i stosuje zasady programowania strukturalnego.</p> <p>Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach.</p> <p>Wie, jakie znaczenie ma zasięg działania zmiennej.</p> <p>Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą.</p>	<p>Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów.</p> <p>Potrafi realizować nawet bardzo złożone algorytmy, stosować procedury graficzne w realizacji skomplikowanych zadań – np. tworzyć własne animacje. Potrafi prezentować algorytmy rekurencyjne w postaci programu; potrafi zamienić rozwiązanie iteracyjne algorytmu na rekurencyjne. Zapisuje w postaci programu wybrane algorytmy sortowania.</p>	<p>Ocenia efektywność działania programu.</p> <p>Wie, na czym polega programowanie obiektowe i zdarzeniowe.</p> <p>Potrafi stosować techniki programowania dynamicznego lub programowania obiektowego.</p> <p>Zna i rozumie podobieństwa i różnice w strukturze programu zapisanego w różnych językach programowania – w deklaracji zmiennych i procedur, w składni i zasadach działania poszczególnych procedur.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p>

	Zna podstawowe zasady poprawnego programowania; testuje tworzone programy; wie, jak uniknąć problemów, takich jak np. zapętlenie się programu.	Zna zasady działania wybranych algorytmów sortowania.  Zna podstawowe procedury graficzne, potrafi narysować na ekranie wykres funkcji i podstawowe figury geometryczne.	Opracowuje złożony program w kilkuosobowej grupie – umie podzielić zadania, ustalić sposoby przekazywania danych pomiędzy procedurami. Zabezpiecza tworzone programy przed wprowadzeniem przez użytkownika błędnych danych.	
--	--	--	--	--

## 2.2. Dobór struktur danych do rozwiązywanego problemu

2	3	4	5	6
Wymienia przykłady prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.	Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady. Deklaruje typy złożone.	Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach.	Rozumie, na czym polega dobór struktur danych do algorytmu. Potrafi zastosować rekordowy typ danych.	Zna dynamiczne struktury danych. Potrafi zastosować zmienne typu wskaźnikowego w zadaniach. Zna struktury listowe, np. stos, kolejkę, listę. Rozumie i potrafi zastosować typ obiektowy.

## 3. Zasady działania komputera i sieci komputerowych

### 3.1. System komputerowy

2	3	4	5	6
<p>Określa następujące pojęcia: bit, bajt. Zna pojęcie systemu pozycyjnego.</p> <p>Wymienia części składowe zestawu komputerowego, podaje ich parametry i przeznaczenie.</p> <p>Rozróżnia rodzaje pamięci komputera, określa ich własności i przeznaczenie.</p> <p>Wie, co to jest system operacyjny, i korzysta z jego podstawowych funkcji.</p> <p>Wykonuje podstawowe operacje na plikach i folderach.</p>	<p>Wie, co to jest system binarny, i potrafi dokonać zamiany liczby z systemu dziesiętnego na binarny i odwrotnie.</p> <p>Potrafi sklasyfikować środki (urządzenia) i narzędzia (oprogramowanie) technologii informacyjnej.</p> <p>Wie, jak działa komputer. Wyjaśnia rolę procesora. Rozumie organizację pamięci komputerowych.</p> <p>Potrafi omówić funkcje systemu operacyjnego.</p> <p>Zna zasady ochrony plików. Potrafi nadać podstawowe atrybuty plikom, jak też wyszukać poszczególne pliki.</p>	<p>Potrafi wykonać działania arytmetyczne na liczbach binarnych (dodawanie i odejmowanie).</p> <p>Zna system szesnastkowy i potrafi wykonać konwersję liczb binarnych na liczby w systemie szesnastkowym i odwrotnie.</p> <p>Analizuje model logiczny komputera. Wie, co to jest kod ASCII.</p> <p>Potrafi wymienić rodzaje aktualnie używanych komputerów.</p> <p>Zna metody wyszukiwania plików.</p>	<p>Potrafi omówić dokładnie działanie procesora. Potrafi wykonać dowolną konwersję pomiędzy systemem dziesiętnym, dwójkowym i szesnastkowym.</p> <p>Zna sposób zapisu liczby całkowitej i rzeczywistej (zmiennoprzecinkowej).</p> <p>Umie wymienić przynajmniej dwa systemy operacyjne i podać ich najważniejsze funkcje.</p> <p>Zna zaawansowane metody wyszukiwania i odzyskiwania plików.</p> <p>Zna przynajmniej jeden algorytm szyfrowania danych. Potrafi zaszyfrować i odszyfrować prosty tekst.</p>	<p>Zna operacje logiczne na liczbach binarnych i przesunięcia bitowe. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Wykonuje sprawnie operacje na liczbach zapisanych w różnych systemach pozycyjnych.</p> <p>Potrafi odzyskać utracony plik, stosując zaawansowane metody.</p> <p>Potrafi omówić różne systemy operacyjne, wskazać ich najważniejsze funkcje.</p> <p>Samodzielnie wyszukuje informacje na temat kompresji i szyfrowania danych. Zna kilka sposobów szyfrowania informacji. Potrafi zapisać algorytm szyfrowania w postaci programu. Zna działanie algorytmu kompresji.</p>

### 3.2. Sieci komputerowe, w tym Internet

2	3	4	5	6
<p>Zna pojęcie sieci komputerowej, potrafi wymienić jej rodzaje.</p> <p>Zna pojęcie logowania.</p> <p>Potrafi wymienić kilka cech pracy w sieci, odróżniających ją od pracy na autonomicznym komputerze.</p> <p>Zna kilka sposobów połączenia z Internetem.</p>	<p>Wymienia korzyści płynące z korzystania z sieci.</p> <p>Zna podstawowe klasy i topologie sieciowe.</p> <p>Potrafi wymienić urządzenia i elementy sieciowe oraz omówić ich ogólne przeznaczenie.</p> <p>Zna cechy systemu działającego w szkolnej pracowni.</p> <p>Orientuje się – w zakresie podstawowym – w działaniu Internetu.</p>	<p>Zna znaczenie protokołu w sieciach (w tym TCP/IP). Wie, na czym polega wymiana informacji w sieci.</p> <p>Zna zasady pracy w sieci, m.in. zasady udostępniania zasobów.</p> <p>Potrafi omówić zagrożenia płynące z sieci.</p> <p>Charakteryzuje różne połączenia z Internetem; potrafi omówić przesyłanie pakietów danych w Internecie.</p>	<p>Zna schemat działania sieci komputerowych.</p> <p>Potrafi wymienić zalety i wady różnych topologii sieci. Charakteryzuje topologie gwiazdy, magistrali i pierścienia.</p> <p>Zna podstawowe cechy systemu Linux.</p> <p>Umie z pomocą nauczyciela zrealizować małą sieć komputerową – skonfigurować jej składniki, udostępnić pliki, dyski, drukarki, dodać użytkowników.</p>	<p>Omawia szczegółowo model warstwowi sieci.</p> <p>Omawia różne systemy sieciowe. Dokonuje ich analizy porównawczej.</p> <p>Charakteryzuje system Linux.</p> <p>Potrafi samodzielnie zbudować małą sieć domową.</p>

### 3.3. Tendencje w rozwoju informatyki i jej zastosowań

2	3	4	5	6
<p>Potrafi omówić historię komputerów.</p> <p>Umie wskazać ogólny kierunek zmian w technologiach komputerowych.</p> <p>Zna podstawowe zasady netykiety.</p>	<p>Potrafi określić nowoczesne trendy w zastosowaniu urządzeń komputerowych.</p> <p>Jest w stanie omówić prawne i społeczne aspekty zastosowania informatyki.</p>	<p>Potrafi wskazać nowości w zakresie usług internetowych oraz odszukać informacje na temat najnowszych pomysłów na komputery.</p>	<p>Przygotowuje analizę porównawczą, pokazującą na przestrzeni wielu lat rozwój informatyki, w tym sieci komputerowych, oraz multimedialnych.</p>	<p>Wskazuje tendencje w rozwoju informatyki i jej zastosowania, dostrzegając przeobrażenia w tej dziedzinie w kraju i na świecie.</p>

## 4. Przetwarzanie danych w bazach danych

### 4.1. Projektowanie relacyjnej bazy danych

2	3	4	5	6
<p>Podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych.</p> <p>Podaje przykłady programów do tworzenia baz danych.</p> <p>Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja).</p> <p>Potrafi uporządkować bazę rosnąco lub malejąco według jednego lub kilku pól.</p>	<p>Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych. Określa podstawowe pojęcia (rekord, pole, typ pola).</p> <p>Samodzielnie tworzy w jednej tabeli bazę danych, składającą się z kilku pól różnych typów. Projektuje przykładowy formularz i raport.</p> <p>Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów.</p> <p>Zna zasady przygotowania korespondencji seryjnej.</p>	<p>Projektuje relacyjną bazę danych składającą się z dwóch tabel połączonych relacją (na zadany temat). Projektuje formularz i raport według wskazówek nauczyciela.</p> <p>Zna zasady definiowania kluczy podstawowych.</p> <p>Drukuje wybrane rekordy, formularze i raporty.</p> <p>Łączy informacje z bazy danych z dokumentami innych programów, np. edytora tekstu czy arkusza kalkulacyjnego.</p>	<p>Potrafi wytłumaczyć pojęcie relacji.</p> <p>Projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji).</p> <p>Zna kilka rodzajów formularzy i raportów, w tym raporty w postaci wykresów.</p> <p>Umie zaprojektować samodzielnie wygląd formularza i raportu.</p> <p>Zna pojęcie indeksu. Odróżnia sortowanie od indeksowania.</p> <p>Potrafi w tworzonej bazie ustalić klucze indeksu.</p>	<p>Zna dokładnie wybrany program do projektowania baz danych.</p> <p>Potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykonać ją w praktyce, np. w szkole czy w domu.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p>

## 4.2. Wyszukiwanie informacji z użyciem języka zapytań

2	3	4	5	6
Wyszukuje informacje w bazie, korzystając wyłącznie z gotowych kwerend i narzędzi wbudowanych do programu.	Tworzy samodzielnie kwerendy (proste i złożone), korzystając z wbudowanych do programu narzędzi.	Zna podstawowe konstrukcje języka zapytań. Wie, co to jest język SQL. Potrafi przeanalizować przykład zapytania utworzonego w języku SQL. Z pomocą nauczyciela potrafi zapisać prostą kwerendę, korzystając z języka zapytań.	Zna zasady wyszukiwania informacji w bazie z wykorzystaniem języka zapytań. Potrafi zapisać złożone kwerendy, korzystając z wybranej instrukcji, np. SELECT; stosuje jej główne klauzule.	Opierając się na profesjonalnej literaturze, potrafi samodzielnie zapisywać złożone kwerendy z wykorzystaniem języka zapytań.

## 4.3. Realizacja projektu programistycznego, w tym przygotowanie dokumentów i raportów

2	3	4	5	6
Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje. Bierze udział w testowaniu projektu.	Zna wszystkie etapy projektowania systemów informatycznych. Uczestniczy czynnie w poszczególnych etapach projektu, wykonując zlecone zadania szczegółowe.	Wie, co to jest system informatyczny. Potrafi omówić zakres prac na każdym etapie. Uczestniczy czynnie w analizie systemu informacyjnego, przygotowuje dokumentację. Pracuje przy projektowaniu tabel, formularzy i raportów.	Wykonuje trudniejsze prace związane z projektowaniem bazy. Projektuje złożone kwerendy, formularze, raporty. Uczestniczy we wdrażaniu systemu informatycznego.	Potrafi wystąpić w roli koordynatora projektu. Przydziela zadania szczegółowe, dba o ich prawidłowe wykonanie, nadzoruje pracę innych, dba o dobrą atmosferę w grupie.

# 5. Wśród multimedialnych

## 5.1. Przetwarzanie informacji w różnych postaciach, m.in. graficznej i dźwiękowej

2	3	4	5	6
Wymienia programy do tworzenia i obróbki grafiki. Posługuje się jednym z nich w celu tworzenia własnych rysunków. Potrafi na kilka sposobów umieścić grafikę w dokumencie tekstowym, np. oblać rysunek tekstem. Wykonuje podstawowe operacje na rysunku, np. skalowanie, kadrowanie. Zna urządzenia multimedialne, wymienia przykładowe nazwy, określa ich ogólne przeznaczenie.	Zna sposoby reprezentacji obrazu i dźwięku w komputerze. Zna możliwości kilku wybranych programów do edycji obrazu i do tworzenia animacji. Zapisuje plik graficzny w różnych formatach (zna zastosowanie poszczególnych formatów, ich zalety i wady). Potrafi wybrać proste fragmenty obrazu i wykonać na nich różne operacje. Korzysta z różnych urządzeń multimedialnych, zna ich działanie, podaje ich przeznaczenie. Określa pojęcie: komputer multimedialny.	Rozróżnia grafikę wektorową i rastrową. Zna różne możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo. Zna pojęcia: RGB i CMYK. Potrafi stosować różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy. Orientuje się, co to jest rozdzielczość. Ustala rozdzielczość dla skanowanych i edytowanych obrazów. Posługuje się sprawnie wybranymi urządzeniami multimedialnymi.	Rozumie twórczy charakter tworzenia grafiki. Zna i stosuje w praktyce zaawansowaną obróbkę grafiki rastrowej. Przy użyciu odpowiednich narzędzi potrafi zaznaczyć fragmenty obrazu nawet o skomplikowanym kształcie. Potrafi zapisywać pliki multimedialne w różnych formatach, ze szczególnym uwzględnieniem formatów internetowych. Zna pojęcia: filtr, histogram, krzywa barw. Potrafi zdefiniować barwy i wykonać na nich operacje. Przekształca obraz – geometrycznie i z zastosowaniem filtrów.	Korzysta z profesjonalnej literatury dotyczącej przetwarzania multimedialnych. Potrafi samodzielnie odkrywać możliwości programów komputerowych w zakresie montażu wideo. Zna metody przechwytywania danych wideo. Zna zasady montażu filmu. Potrafi wykonać taki montaż. Zna sposoby udostępniania filmu innym osobom oraz metody konwersji między różnymi formatami.

<p>W stopniu podstawowym posługuje się drukarką i skanerem. Po zeskanowaniu zapisuje obraz w pliku w domyślnym formacie.</p>	<p>Rozumie zasady łączenia poszczególnych elementów multimedialnych, np. obrazu z dźwiękiem.</p>	<p>Potrafi zastosować zasady tworzenia multimedialnych stron internetowych i prezentacji multimedialnych.</p> <p>Zna zasady działania animacji. Tworzy własne animacje.</p>	<p>Zna pojęcia: fotomontaż, warstwa obrazu, maska. Potrafi tworzyć przykładowe fotomontaże. Potrafi pracować z warstwami obrazu i retuszować obraz.</p> <p>Przygotowuje grafikę na własną stronę internetową lub do prezentacji multimedialnej. Optymalizuje pliki dla konkretnych potrzeb.</p> <p>Tworzy własną animację. Łączy wideo, dźwięk, animację i obraz statyczny, np. w programie do animacji (Flash) lub programie do obróbki wideo.</p>	
--	--	---	---	--

## 5.2. Tworzenie prezentacji multimedialnych

2	3	4	5	6
<p>Przy użyciu szablonu projektu tworzy prezentację składającą się z kilku slajdów.</p> <p>Wstawia teksty i obrazy, stosuje proste animacje.</p> <p>Zna ogólne zasady tworzenia prezentacji. Potrafi dobrać tło, atrybuty czcionek, odpowiednio rozmieścić tekst i grafikę na slajdzie.</p>	<p>Rozróżnia sposoby przygotowania prezentacji wspomagającej wystąpienie prelegenta oraz prezentacji typu kiosk.</p> <p>Wie, na czym polega dostosowanie treści i formy do rodzaju prezentacji.</p> <p>Zna i stosuje poprawne zasady tworzenia prezentacji wspomagającej wystąpienie prelegenta.</p> <p>Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Posługuje się szablonem projektu. Wyszukuje i gromadzi gotowe materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animację.</p> <p>Stosuje zasady prezentowania pokazu slajdów.</p> <p>Zna zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.</p>	<p>Zna i stosuje metody projektowania różnych rodzajów prezentacji.</p> <p>Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Wybiera temat, przygotowuje scenariusz, wyszukuje oraz tworzy własne materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animację.</p> <p>Posługuje się widokiem sortowania slajdów.</p> <p>Stosuje zasady referowania konkretnego tematu wspomaganego prezentacją.</p> <p>Zna i stosuje zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.</p> <p>Dodaje efekty multimedialne: animacje, grafiki, dźwięki, podkład muzyczny.</p> <p>Ustawia i testuje chronometraż. Stosuje hiperłącza.</p>	<p>Na gotowym, poprawnie wykonanym przykładzie przedstawia zasady tworzenia prezentacji multimedialnych. Wyjaśnia, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Wyjaśnia różnice w zasadach projektowania prezentacji wspomagającej wystąpienie prelegenta, prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.</p> <p>Potrafi organizować prezentację w widoku konspektu.</p> <p>Wykorzystuje możliwości tworzenia schematu organizacyjnego oraz możliwości tworzenia wykresów.</p> <p>Nagrywa narrację. Poprawnie ustawia i testuje chronometraż, stosuje hiperłącza.</p>	<p>Dodaje do prezentacji materiały ze skanera, aparatu cyfrowego i kamery cyfrowej.</p> <p>Publikuje prezentację w Internecie. Dopasowuje parametry konwersji do formatu HTML.</p> <p>Przygotowuje materiały ułatwiające opracowanie prezentacji, np. wydruk miniaturki slajdów wraz z notatkami.</p> <p>Nagrywa narrację i dodaje ją do prezentacji.</p>



		Aktywnie współpracuje z grupą przy projektowaniu prezentacji.	Zwraca uwagę na różnice między opracowaniem prezentacji za pomocą Kreatora zawartości i szablonu projektu a opracowaniem tzw. prezentacji pustej.  Uczestniczy w przygotowaniu w formie projektów grupowych: prezentacji do samodzielnego przeglądania przez odbiorcę oraz samouruchamiającej się.  Zna sposoby umieszczania prezentacji w Internecie.	
--	--	---	--	--

### 5.3. Tworzenie stron internetowych

2	3	4	5	6
<p>Wymienia przykładowe programy do projektowania i tworzenia stron internetowych.</p> <p>Potrafi wymienić podstawowe elementy, z których składa się strona WWW.</p> <p>W stopniu podstawowym posługuje się wybranym programem do tworzenia stron.</p> <p>Tworzy nieskomplikowaną stronę, np. z informacjami o sobie samym. Wstawia tytuł, formatuje tekst, umieszcza obraz.</p>	<p>Wie, co to jest język znaczników HTML, i potrafi omówić strukturę pliku w tym języku.</p> <p>Zna podstawy języka znaczników HTML i potrafi wykonać prostą stronę na zadany przez nauczyciela temat.</p> <p>Z pomocą nauczyciela projektuje wygląd strony. Planuje jej zawartość (teksty, rysunki, dźwięki, animacje) i umieszcza na niej ww. elementy.</p>	<p>Potrafi samodzielnie zaprojektować wygląd strony.</p> <p>Zna reguły poprawnego projektowania układu strony, m.in. dba o jej czytelność i przejrzystość, o poprawność redakcyjną i merytoryczną oraz prawną umieszczanych na niej tekstów i materiałów.</p> <p>Zna zaawansowane możliwości języka HTML: tabele, ramki, style.</p> <p>Zna sposoby publikowania stron w Internecie oraz wady i zalety tych sposobów.</p>	<p>Samodzielnie korzysta z wybranego programu do tworzenia stron.</p> <p>Potrafi wykorzystać nowo poznane funkcje języka HTML. Wykorzystuje je do udoskonalenia istniejących już, swoich własnych stron.</p> <p>Włącza licznik odwiedzin na stronie. Dodaje inne typowe elementy: forum, księgę gości.</p> <p>Zna podstawy języka JavaScript. Używa go dla osiągnięcia nieskomplikowanych efektów wizualnych na stronie.</p> <p>Potrafi opublikować stronę w Internecie.</p>	<p>Potrafi samodzielnie zapoznać się z nowym programem do tworzenia stron internetowych.</p> <p>Potrafi posługiwać się językiem JavaScript w tworzeniu tzw. stron dynamicznych.</p> <p>Potrafi wykorzystać gotowe lub własne skrypty serwerowe: PHP, CGI, PERL, SSI, ASP.</p> <p>Umie kreować bazy danych – np. MySQL – w połączeniu z językami skryptowymi.</p>

## VIII. Propozycje sposobów oceny osiągnięć ucznia

W rozdziale VII przedstawiono szczegółowe wymagania na poszczególne oceny szkolne. Powinny one stanowić podstawę oceny wiedzy i umiejętności uczniów. Zaleca się, aby uczniowie znali te wymagania.

Uczniowie na zajęciach z technologii informacyjnej, a zwłaszcza informatyki realizowanej w zakresie rozszerzonym, powinni być oceniani nie tylko za wykonywanie ćwiczeń praktycznych, ale również za znajomość materiału nauczania. W ocenie powinniśmy uwzględniać umiejętność łączenia wiedzy teoretycznej z ćwiczeniami praktycznymi, zdolność wnioskowania i przeprowadzania analiz. Uczniowie powinni rozumieć, że poznawanie metod i pojęć informatyki ułatwia im rozwiązywanie zadań praktycznych.

Jeśli badanie kompetencji ucznia odbywa się w postaci sprawdzianu praktycznego przy komputerach, to forma proponowanych zadań nie powinna odbiegać od ćwiczeń, które uczniowie wykonywali na zajęciach. W tym przypadku ćwiczenia powinny sprawdzać, czy dany problem został należycie zrozumiany przez uczniów oraz na ile samodzielnie doszli do rozwiązania. Przy wykonywaniu niektórych ćwiczeń wskazane jest ponadto, aby uczeń mógł korzystać z materiałów pomocniczych, np. własnych notatek z zeszytu przedmiotowego oraz z podręcznika. Ważną umiejętnością ucznia jest bowiem korzystanie z różnych materiałów w celu rozwiązania problemu. Kiedy przygotowujemy sprawdzian, ważne jest prawidłowe sformułowanie ćwiczenia – tak aby uczeń nie odnajdywał szczegółowej odpowiedzi w podręczniku. Należy również bardzo precyzyjnie określić czas trwania sprawdzianu – nie może być zbyt długi. Jest to podejście przemyślane metodycznie i sprawdzone przez autorkę w praktyce. Wówczas tylko ci uczniowie, którzy mają dobre notatki i dobrze znają podręcznik (wiedzą, gdzie znaleźć odpowiednie treści), potrafią je szybko odszukać i z nich skorzystać.

Poniżej przedstawiono przykłady ćwiczeń i pytań (z działu *Algorytmika i programowanie*), które mogą być realizowane, kiedy uczeń korzysta z własnego zeszytu lub podręcznika. Wymagają one bowiem samodzielnej analizy problemu.

### Przykłady ćwiczeń

**Ćwiczenie 1.** W ramce przedstawiono fragment programu.

- Jaki jest wynik działania zapisanej instrukcji dla następujących wartości  $x$ : -6, 100, 0, 4, -4?
- Zapisz, dla jakich wartości  $x$  zostanie wypisany napis 'SMS', dla jakich 'MMS', a dla jakich wartość zmiennej  $x$ ?

**JEŚLI  $x < 4$  TO JEŚLI  $x < 0$  TO WYPROWADŹ ( $x$ )  
W PRZECIWNYM WYPADKU WYPROWADŹ ('SMS')  
W PRZECIWNYM WYPADKU WYPROWADŹ ('MMS')**

- Podaj dodatkowo, co oznaczają skróty SMS i MMS.

**Ćwiczenie 2.** Dokonaj analizy algorytmu przedstawionego na rysunku w postaci schematu blokowego.

Dla jakich wartości  $k$  nastąpi zapętlenie? Popraw schemat tak, aby nie było możliwe zapętlenie, a wyprowadzana została podwojona wartość liczby dodatniej.

### Przykład pytań (problemów)

- Ile razy będzie wykonana pętla:  
**DLA  $k = 15$  DO 1 WYKONAJ WYPROWADŹ** ('To chyba coś nie tak')?
- Dlaczego w instrukcji pętli **DLA** nie trzeba stosować dodatkowego licznika typu:  $i := i + 1$ ?

### Przykład zadania dla zainteresowanych

- Zaproponuj własny pseudojęzyk, określając w nim podstawowe słowa kluczowe, zestaw instrukcji oraz zasady składni. Napisz w nim dowolny program, pokazując na jego przykładzie, na czym polega pisanie programów.

