

Szczegółowe wymagania edukacyjne na poszczególne oceny z informatyki rozszerzonej w szkole ponadgimnazjalnej dla klasy drugiej, trzeciej.

Podręcznik: Informatyka Europejczyka, Helion – poziom rozszerzony, nr dopuszczenia: 410/2/2012

Obowiązują następujące kryteria dla poszczególnych ocen:

• **stopień celujący** otrzymuje uczeń, który: posiada wiedzę i umiejętności znacznie wykraczające poza program nauczania przedmiotu w danej klasie, samodzielnie i twórczo rozwija własne uzdolnienia, biegłe posługuje się zdobytymi wiadomościami w rozwiązywaniu problemów, proponuje rozwiązania nietypowe, rozwiązuje także zadania wykraczające poza program lub posiada inne porównywalne osiągnięcia.

Stopień celujący otrzymuje uczeń, który spełnia kryteria na ocenę bardzo dobrą, a ponadto: samodzielnie rozwija swoją wiedzę i służy pomocą innym, potrafi w sposób twórczy, odkrywczy rozwiązywać różnorodne problemy, potrafi włączyć do swoich rozwiązań wiadomości z innych dziedzin nauki, bierze udział w konkursach i olimpiadach, na tle grupy uczniów wyróżnia się sprawnością swojej pracy i jakością otrzymanych wyników.

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Rozumie pojęcie: algorytm i podaje samodzielnie przykłady. Samodzielnie formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Samodzielnie podaje algorytmy związane z życiem codziennym, innymi nauczonymi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	Samodzielnie i poprawnie definiuje problem i formułuje jego specyfikację. Stosuje podejście algorytmiczne do rozwiązywanego problemu zaawansowanego. Zna i realizuje etapy rozwiązywania zadań za pomocą komputera (od sformułowania specyfikacji problemu po testowanie rozwiązania). Rozwiązuje problem zgodnie z etapami: zaplanowanie działań, wydzielenie podproblemów i określenie zależności między nimi, skonstruowanie algorytmów rozwiązujących podproblemy, analiza tych algorytmów i określenie ogólnego algorytmu rozwiązującego problem. Dokonuje zaawansowanej analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją. Ocenia szczegółowo zgodność algorytmu ze specyfikacją problemu.
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Stosuje zaawansowane sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji: <ul style="list-style-type: none">• schemat blokowy — graficzna prezentacja algorytmu,• lista kroków — słowna prezentacja algorytmu,• program w wybranym języku programowania	Zapisuje zaawansowane algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków, program w języku programowania wysokiego poziomu, arkusz kalkulacyjny. Stosuje elementy składowe algorytmu z doбором właściwych sposobów rozwiązania i prezentacji algorytmu do konkretnego problemu. Stosuje rozwiązanie zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie notacji.

<p>wysokiego poziomu — praktyczna realizacja algorytmu,</p> <ul style="list-style-type: none"> • arkusz kalkulacyjny. 	<p>Wykorzystuje zaawansowane funkcje arkusza kalkulacyjnego do obrazowania zależności funkcyjnych i zapisywania algorytmów.</p> <p>Stosuje informatyczne rozwiązanie algorytmu, dobiera odpowiednie struktury danych (w tym struktury dynamiczne) i najlepszy algorytm oraz zapisuje go w wybranym języku programowania.</p> <p>Stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu.</p> <p>Wykorzystuje różne metody algorytmiczne i algorytmy klasyczne do rozwiązywania problemu oraz świadomie wybiera właściwy sposób rozwiązania zadania.</p> <p>Stosuje zdobytą wiedzę i umiejętności do rozwiązywania złożonych zadań z różnych dziedzin.</p>
<p>Algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.</p>	<p>Stosuje algorytmy liniowe i algorytmy z warunkami przy rozwiązywaniu problemów.</p> <p>Stosuje algorytmy liniowe i warunkowe: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe itp.</p>
<p>Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.</p>	<p>Stosuje pojęcia: iteracja, wzór iteracyjny, Stosuje metodę iteracji przy realizacji algorytmów.</p> <p>Stosuje wykorzystaną metodę iteracji w przykładowych algorytmach.</p> <p>Stosuje iterację do zapisu algorytmów: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach (łańcuchach) oraz tablicach (na przykład znajdowanie podciągów o określonych własnościach) itp.</p>
<p>Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.</p>	<p>Stosuje pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna w sytuacjach życiowych.</p> <p>Stosuje zamiany metody iteracyjnej wykorzystanej w algorytmie na rekurencyjną.</p> <p>Stosuje wykorzystaną metodę rekurencji w przykładowych algorytmach.</p> <p>Stosuje algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi itp.</p> <p>Analizuje wartości zmiennych w algorytmach rekurencyjnych.</p> <p>Stosuje rekurencję w prostych sytuacjach problemowych.</p>
<p>Metoda „dziel i zwyciężaj”.</p>	<p>Stosuje pojęcie: metoda „dziel i zwyciężaj”.</p> <p>Stosuje metodę „dziel i zwyciężaj” w odpowiednich sytuacjach.</p> <p>Stosuje algorytmy z wykorzystaniem metody „dziel i zwyciężaj”.</p>
<p>Kryptografia i kryptoanaliza. Metody szyfrowania.</p>	<p>Stosuje zaawansowane metody szyfrowania.</p>
<p>ALGORYTMY I ICH ZASTOSOWANIE</p>	
<p>Algorytmy badające własności geometryczne:</p> <ul style="list-style-type: none"> • sprawdzanie warunku trójkąta, • określanie prostokątności i równoległości prostych, • obliczanie odległości punktu od prostej, 	<p>Definiuje prostą w postaci ogólnej i kierunkowej na podstawie wzorów wyznacza odległość punktu od prostej oraz odległość punktów na płaszczyźnie; na podstawie równań kierunkowych określa prostokątność i równoległość prostych.</p> <p>Stosuje algorytm sprawdzający warunek w języku programowania.</p>

<ul style="list-style-type: none"> • obliczanie odległości punktów na płaszczyźnie, • badanie położenia punktu względem prostej, • badanie przynależności punktu do odcinka. 	
<p>Wyznaczanie największego wspólnego dzielnika i najmniejszej wspólnej wielokrotności dwóch liczb naturalnych:</p> <ul style="list-style-type: none"> • wyznaczanie największego wspólnego dzielnika — algorytm Euklidesa, • wyznaczanie najmniejszej wspólnej wielokrotności. 	<p>Wyjaśnia i implementuje w języku programowania największy wspólny dzielnik dwóch liczb naturalnych algorytmem Euklidesa. Stosuje iteracyjną i rekurencyjną realizację algorytmu.</p> <p>Wyjaśnia i implementuje w języku programowania najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający największy wspólny dzielnik.</p>
<p>Wyznaczanie wartości wielomianu, pozycyjne systemy liczbowe i reprezentacja danych liczbowych w komputerze:</p> <ul style="list-style-type: none"> • wyznaczanie wartości wielomianu schematem Hornera, • szybkie podnoszenie do potęgi, • pozycyjne systemy liczbowe stosowane w informatyce — system dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny) — definicja, zamiana liczb zapisanych w podanych systemach, realizacja podstawowych operacji arytmetycznych w podanych systemach, 	<p>Wyjaśnia i implementuje w języku programowania algorytm obliczający wartość wielomianu za pomocą schematu Hornera.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera.</p> <p>Wyjaśnia i implementuje w języku programowania: system liczbowy, pozycyjny system liczbowy: dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny), oraz zamienia liczby zapisane w tych systemach liczbowych na system dziesiętny i odwrotnie.</p> <p>Wyjaśnia i implementuje w języku programowania przykład systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera.</p> <p>Wyjaśnia i implementuje w języku programowania operacje arytmetyczne w dowolnych pozycyjnych systemach liczbowych.</p> <p>Wyjaśnia reprezentacje danych liczbowych w komputerze, źródło błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.</p>
<ul style="list-style-type: none"> • zamiana liczb z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera, • zamiana liczb z systemu dziesiętnego na dowolny pozycyjny system liczbowy, • wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych, • reprezentacja danych liczbowych w komputerze — reprezentacja binarna liczb całkowitych i niecałkowitych, reprezentacja stałopozycyjna i zmiennopozycyjna liczb, błędy w obliczeniach. 	
<p>Liczby pierwsze:</p>	<p>Wyjaśnia pojęcie: liczba pierwsza.</p>

<ul style="list-style-type: none"> • sprawdzanie, czy liczba jest liczbą pierwszą, • rozkładanie liczby na czynniki pierwsze, • sito Eratostenesa. 	<p>Wyjaśnia i implementuje w języku programowania algorytm sprawdzający, czy dana liczba jest pierwsza.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm generujący liczby pierwsze — sito Eratostenesa.</p>
<p>Przeszukiwanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • przeszukiwanie liniowe tablicy jednowymiarowej, • przeszukiwanie liniowe tablicy jednowymiarowej z wartownikiem. 	<p>Wyjaśnia i implementuje w języku programowania metody liniowe przy przeszukiwaniu ciągu liczbowego, tablicy jednowymiarowej.</p>
<p>Znajdowanie największego lub najmniejszego elementu w ciągu liczbowym:</p> <ul style="list-style-type: none"> • znajdowanie największego elementu w ciągu liczbowym, • znajdowanie najmniejszego elementu w ciągu liczbowym. 	<p>Wyjaśnia i implementuje w języku programowania algorytm znajdowania największego elementu w ciągu liczbowym.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm znajdowania najmniejszego elementu w ciągu liczbowym.</p>
<p>Znajdowanie lidera w zbiorze.</p>	<p>Wyjaśnia pojęcie: lider w zbiorze.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm znajdowania lidera w zbiorze.</p>
<p>Sprawdzanie monotoniczności ciągu liczbowego:</p> <ul style="list-style-type: none"> • sprawdzanie, czy ciąg liczbowy jest rosnący lub malejący, • sprawdzanie, czy ciąg liczbowy jest nierosnący lub niemalejący, • sprawdzanie, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny. 	<p>Rozumie pojęcie: monotoniczność ciągu liczbowego.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm sprawdzający, czy ciąg liczbowy jest rosnący lub malejący.</p> <p>Wyjaśnia i implementuje w języku programowania algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.</p>
<p>Liniowe sortowanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • porządkowanie przez wybór, • porządkowanie przez wstawianie, • porządkowanie bąbelkowe. 	<p>Podaje wyjaśnienie algorytmu porządkowania przez wybór.</p> <p>Podaje wyjaśnienie algorytmu porządkowania przez wstawianie.</p> <p>Podaje wyjaśnienie algorytmu porządkowania bąbelkowego.</p>
<p>Zastosowanie metody „dziel i zwyciężaj”:</p> <ul style="list-style-type: none"> • jednoczesne znajdowanie najmniejszego i największego elementu, • przeszukiwanie binarne ciągu uporządkowanego, 	<p>Podaje proste wyjaśnienie algorytmu metody „dziel i zwyciężaj” przy realizacji algorytmów.</p> <p>Podaje proste wyjaśnienie algorytmu jednoczesnego znajdowania najmniejszego i największego elementu.</p> <p>Podaje proste wyjaśnienie algorytmu przeszukiwania binarnego uporządkowanego ciągu liczbowego.</p>
<ul style="list-style-type: none"> • sortowanie przez scalanie, • sortowanie szybkie, • znajdowanie przybliżonej wartości miejsca zerowego 	<p>Podaje wyjaśnienie algorytmu sortowania przez scalanie.</p> <p>Podaje wyjaśnienie algorytmu sortowania szybkiego.</p> <p>Podaje wyjaśnienie algorytmu znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metodą</p>

funkcji ciągłej — metoda połowienia przedziałów.	połowienia przedziałów.
<p>Obliczenia przybliżone. Realizacja wybranych algorytmów numerycznych:</p> <ul style="list-style-type: none"> • obliczanie wartości pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona), • obliczanie pola obszaru ograniczonego wykresem funkcji (całkowanie numeryczne), • przybliżanie wartości liczby metodą Monte Carlo. 	<p>Podaje wyjaśnienie pojęcia: obliczenia przybliżone, metody numeryczne.</p> <p>Podaje wyjaśnienie algorytmu obliczającego wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona).</p> <p>Podaje wyjaśnienie algorytmu obliczającego pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów.</p> <p>Podaje wyjaśnienie algorytmu wykonującego przybliżanie wartości liczby metodą Monte Carlo.</p>
<p>Algorytmy na tekstach (łańcuchach):</p> <ul style="list-style-type: none"> • sprawdzanie, czy tekst jest palindromem, • porządkowanie alfabetyczne tekstu, • sprawdzanie, czy teksty są anagramami, • wyszukiwanie wzorca w tekście, • wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP. 	<p>Podaje wyjaśnienie algorytmu wykonywanego na tekstach, na przykład sprawdzanie, czy tekst jest palindromem, porządkowanie alfabetyczne tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście.</p> <p>Podaje wyjaśnienie notacji wyrażenia podanego w postaci ONP. Zapisuje i stosuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.</p>
<p>Zastosowanie programowania zachłannego:</p> <ul style="list-style-type: none"> • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana. 	<p>Podaje wyjaśnienie algorytmów: algorytm zachłanny dla problemu plecakowego, algorytm wydawania reszty, algorytm Huffmana.</p>
Wybrane algorytmy kryptograficzne.	<p>Podaje wyjaśnienie algorytmów np.: kryptograficzne, w tym algorytmy symetryczne (np. szyfr Cezara, szyfr przestawieniowy, szyfr płotowy, szyfry monoalfabetyczne i wieloalfabetyczne), algorytmy asymetryczne (np. szyfr z kluczem jawnym RSA), podpis cyfrowy itp.</p>
WŁASNOŚCI ALGORYTMÓW	
<p>Złożoność czasowa algorytmów. Analiza złożoności czasowej wybranych algorytmów.</p>	<p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność czasową algorytmów.</p> <p>Analizuje złożoność czasową wybranych algorytmów (oblicza liczbę operacji dominujących wykonywanych przez algorytm).</p> <p>Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera możliwie najszybszy algorytm rozwiązujący postawiony problem.</p>
<p>Złożoność pamięciowa algorytmów. Analiza złożoności pamięciowej wybranych algorytmów.</p>	<p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność pamięciową algorytmów.</p> <p>Analizuje złożoność pamięciową wybranych algorytmów (szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu).</p>

	<p>Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera odpowiednie struktury danych w rozwiązaniu postawionego problemu.</p> <p>Wyjaśnia pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu.</p> <p>Opisuje własności algorytmów na podstawie ich analizy.</p> <p>Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności.</p> <p>Bada efektywność komputerowych rozwiązań problemów.</p> <p>Ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>
--	---

2. Programowanie

Tabela 2. Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
Języki programowania — pojęcie, klasyfikacja, przykłady. Prezentacja przykładowych programów w różnych językach programowania. Podstawowe zasady i metody programowania. Znajomość pojęć: translator, kompilator, interpreter, debugger, linker.	<p>Rozumie pojęcia: język programowania, translator, kompilator, interpreter, debugger, linker.</p> <p>Zna klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego i wysokiego poziomu.</p> <p>Wyjaśnia języki programowania.</p> <p>Tworzy programy napisane w różnych językach programowania.</p>
Elementy zintegrowanego systemu programowania, w tym umiejętność pisania, kompilowania i uruchamiania programów za pomocą wybranego kompilatora.	Dobiera z wybranego środowiska programistycznego (na przykład kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.
<p>Wprowadzenie do programowania:</p> <ul style="list-style-type: none"> • struktura programu, • operacje wejścia-wyjścia, • zmienne i ich deklaracja, • stałe i ich deklaracja, • wyrażenia arytmetyczne, relacje i operatory logiczne, • priorytety relacji i działań, • komentarze. 	<p>Zna podstawową strukturę programu.</p> <p>Korzysta w programach z podstawowych operacji wejścia i wyjścia.</p> <p>Potrafi deklorować zmienne i wykorzystywać je w programach.</p> <p>Zna podstawowe wyrażenia arytmetyczne, relacje i operatory logiczne oraz stosuje je w programach.</p> <p>Zna priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je przy pisaniu programów.</p> <p>Stosuje komentarze przy pisaniu programów.</p>
<p>Podstawowe konstrukcje algorytmiczne:</p> <ul style="list-style-type: none"> • instrukcja przypisania, 	<p>Stosuje podstawowe konstrukcje algorytmiczne, w tym instrukcję przypisania, instrukcję złożoną, instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne.</p> <p>Stosuje w programach powyższe konstrukcje algorytmiczne.</p>

<ul style="list-style-type: none"> • instrukcja złożona, • instrukcje warunkowe, • instrukcja wyboru, • instrukcje iteracyjne. <p>Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.</p>	
<p>Proste typy danych:</p> <ul style="list-style-type: none"> • typy całkowite, • typy rzeczywiste, • typ znakowy, • typ logiczny. <p>Realizacja operacji z wykorzystaniem prostych typów danych oraz ich zastosowanie w programach.</p>	<p>Rozumie pojęcie: proste typy danych.</p> <p>Zna proste typy danych, w tym typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny.</p> <p>Wykonuje operacje na typach danych dla wykorzystywanych zmiennych.</p>
<p>Strukturalizacja programu:</p> <ul style="list-style-type: none"> • definiowanie procedur-funkcji, • zmienne lokalne i globalne, • parametry formalne i aktualne, • przekazywanie parametrów w procedurach-funkcjach, • przeciążanie funkcji. <p>Zastosowanie w implementacjach procedur-funkcji z parametrami. Umiejętność wyboru właściwego w danej sytuacji sposobu przekazywania parametrów.</p>	<p>Rozumie zasady i cel strukturalizacji programu.</p> <p>Rozumie różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach.</p> <p>Rozumie różnice między parametrami formalnymi i aktualnymi.</p> <p>Zna sposoby przekazywania parametrów w procedurach-funkcjach i świadomie stosuje je podczas realizacji programów.</p> <p>Zna podstawowe sposoby przeciążania funkcji i stosuje je w programach.</p>
<p>Strukturalne typy danych:</p> <ul style="list-style-type: none"> • łańcuchy (w tym operacje na tekście), • tablice (w tym jednowymiarowe i wielowymiarowe), • rekordy-struktury (w tym tablice rekordów-struktur), • struktury dynamiczne (w tym listy, stosy, kolejki, drzewa binarne). <p>Realizacja programów z wykorzystaniem strukturalnych typów danych.</p>	<p>Rozumie pojęcia: strukturalne typy danych, abstrakcyjne typy danych.</p> <p>Potrafi definiować typ łańcuchowy.</p> <p>Zna predefiniowane operacje na tekście i z nich korzysta.</p> <p>Zapisuje programy z wykorzystaniem typu łańcuchowego.</p> <p>Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe.</p> <p>Zapisuje programy z wykorzystaniem typu tablicowego.</p> <p>Potrafi definiować typ rekordowy-strukturalny.</p> <p>Potrafi definiować tablice rekordów-struktur.</p> <p>Zapisuje programy z wykorzystaniem typu rekordowego-strukturalnego, w tym tablic rekordów-struktur.</p> <p>Potrafi definiować dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne.</p>

	<p>Zapisuje programy z wykorzystaniem dynamicznych struktur danych.</p> <p>Dobiera struktury danych (w tym dynamiczne struktury danych) odpowiednio do przetwarzanych informacji.</p> <p>Wykorzystuje typy strukturalne, w tym typ łańcuchowy, tablicowy i rekordowy-strukturalny, oraz dynamiczne struktury danych do implementacji algorytmów.</p>
<p>Plikowe operacje wejścia-wyjścia:</p> <ul style="list-style-type: none"> • rodzaje plików, • definicja i etapy przetwarzania plików, • podstawowe operacje na plikach. <p>Realizacja programów z zastosowaniem operacji plikowych.</p>	<p>Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne.</p> <p>Zna i realizuje etapy przetwarzania plików.</p> <p>Zapisuje programy z wykorzystaniem typu plikowego, w tym importuje dane z plików zewnętrznych, eksportuje wyniki do plików zewnętrznych.</p> <p>Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach.</p>
<p>Programowanie zorientowane obiektowo:</p> <ul style="list-style-type: none"> • klasy i obiekty, • dziedziczenie i hierarchia klas, • konstruktory i destruktory, • polimorfizm, • metody wirtualne. <p>Realizacja implementacji z zastosowaniem programowania zorientowanego obiektowo.</p>	<p>Rozumie pojęcia: programowanie obiektowe, klasa, obiekt, enkapsulacja, dziedziczenie, hierarchia klas, polimorfizm, konstruktor, destruktor.</p> <p>Potrafi definiować typ obiektowy.</p> <p>Tworzy proste klasy i statyczne obiekty.</p> <p>Stosuje metody konstruktora i destruktora.</p> <p>Projektuje i realizuje hierarchię klas.</p> <p>Potrafi definiować i stosować w programach dynamiczne obiekty.</p> <p>Stosuje metody wirtualne.</p> <p>Realizuje programy z wykorzystaniem typu obiektowego.</p>
<p>Projekt programistyczny.</p>	<p>Zna pojęcie: inżynieria oprogramowania.</p> <p>Potrafi określić kolejne fazy konstruowania oprogramowania.</p> <p>Znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin.</p> <p>Realizuje indywidualnie lub zespołowo projekt programistyczny, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.</p>

3. Bazy danych

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych

TRZĘCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
BAZODANOWE FUNKCJE ARKUSZA KALKULACYJNEGO	
Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.	<p>Zna i rozumie pojęcie: baza danych.</p> <p>Zna i stosuje reguły, jakim podlega tabela stanowiąca źródło danych.</p>

	<p>Potrafi zgromadzić w pliku tekstowym dane będące bazą danych.</p> <p>Importuje dane umieszczone w pliku tekstowym do tabeli arkusza kalkulacyjnego.</p> <p>Zapisuje zgromadzone w tabeli dane w pliku tekstowym, stosując odpowiednie znaki separacji.</p>
Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.	<p>Potrafi stosować złożone kryteria wyboru w filtrach zaawansowanych.</p> <p>Stosuje odpowiedni rodzaj filtru w celu wybrania informacji potrzebnych w danej sytuacji.</p>
Tworzenie podsumowań danych.	<p>Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich.</p> <p>Rozumie pojęcie: tabela przestawna.</p> <p>Potrafi wyjaśnić do czego służą tabele przestawne.</p> <p>Wskazuje, do jakich danych w celu ich podsumowania można zastosować tabele przestawne.</p> <p>Potrafi wykonać zestawienie podsumowań danych przy użyciu kreatora tabel przestawnych.</p> <p>Potrafi zmienić ustawienia w istniejącej tabeli przestawnej.</p>
Graficzna prezentacja danych na wykresach.	<p>Potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych.</p> <p>Umie tworzyć różnego typu wykresy do danych zawartych w tabelach i tabelach przestawnych.</p> <p>Potrafi ustalić odpowiednie opcje wykresu i formatować poszczególne jego elementy.</p> <p>Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauki szkolnej (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).</p>
RELACYJNA BAZA DANYCH	
Podstawy relacyjnej bazy danych.	<p>Zna różne typy baz danych — proste i złożone.</p> <p>Wyjaśnia pojęcia: system zarządzania bazą danych, relacyjna baza danych.</p> <p>Wyjaśnia zasady projektowania bazy danych. Rozumie potrzebę unikania nadmiarowości danych.</p> <p>Wyjaśnia, czym są pola i rekordy oraz jaką rolę pełni klucz w bazie danych.</p> <p>Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makra).</p> <p>Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego.</p> <p>Potrafi zaprojektować układ tabel, unikając powtarzania danych.</p> <p>Tworzy tabele przy użyciu kreatora tabel oraz w widoku projektu.</p> <p>Potrafi ustalić właściwości pól. Zna i poprawnie stosuje typy danych w tabelach.</p> <p>Ustala podstawowe właściwości pól tabeli, takie jak rozmiar pola, maska wprowadzania czy reguła sprawdzania poprawności.</p> <p>Rozumie, jaką rolę w bazie danych pełni indeks.</p> <p>Potrafi wyjaśnić pojęcia: klucz główny oraz klucz obcy.</p> <p>Poprawnie definiuje związki między tabelami.</p>

	<p>Rozróżnia podstawowe typy relacji między tabelami, potrafi je nazwać.</p> <p>Kontroluje i zmienia warunki wymuszania więzów integralności podczas edytowania właściwości określonej relacji.</p> <p>Potrafi tworzyć i modyfikować tabele, uwzględniając różne typy danych oraz ustalone dla nich właściwości pól (rozmiar, ograniczenia, wymagania, komunikaty dla użytkownika itp.).</p> <p>Planuje współpracę i zespołowe wykonanie projektu relacyjnej bazy danych</p>
<p>Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.</p>	<p>Stosuje różne formaty danych.</p> <p>Wykonuje czynności edycyjne na tabelach bazy danych.</p> <p>Stosuje, jak wprowadzić dane bezpośrednio do tabeli lub poprzez formularz.</p> <p>Stosuje, jak korzystać z formularzy do wprowadzania, wyszukiwania i edycji danych.</p> <p>Potrafi porządkować tabele i przeglądać wybrane rekordy przez sortowanie jedno- i wieloparametrowe.</p> <p>Zmienia sposób prezentowania danych poprzez ich sortowanie lub filtrowanie.</p> <p>Potrafi drukować tabele i formularze.</p> <p>Tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu.</p> <p>Stosuje etykiety, formanty, nagłówek lub stopkę do formularza, zmienia jego szatę graficzną.</p>
<p>Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend.</p>	<p>Stosuje typy kwerend.</p> <p>Samodzielnie projektuje zapytania, korzystając z widoku projektu oraz kreatorów różnego typu kwerend wybierających dane z jednej, kilku tabel lub z wcześniej zdefiniowanych kwerend.</p> <p>Stosuje w zapytaniach znaki wieloznaczne (zastępcze).</p> <p>Konstruuje pola obliczeniowe, wprowadza korekty.</p> <p>Określa złożone kryteria liczbowe i definiuje wyrażenia.</p> <p>Stosuje odpowiednie kryteria i parametry do wyszukiwania danych.</p> <p>Stosuje wiele kryteriów wyboru w jednym polu, stosując odpowiednie operatory.</p> <p>Stosuje kwerendy obliczające podsumowania.</p> <p>Stosuje zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn.</p> <p>Stosuje kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane.</p> <p>Zmienia właściwości sprzężenia między tabelami, na podstawie, których tworzona jest kwerenda.</p> <p>Analizuje kod zapisany w języku zapytań SQL.</p> <p>Modyfikując kryteria zapytania, wprowadza zmiany w kodzie języka zapytań SQL.</p> <p>Stosuje podstawy języka zapytań SQL pozwalające na tworzenie i modyfikowanie kwerend wybierających, parametrycznych, krzyżowych i modyfikujących tabele.</p>
<p>Przygotowanie zestawień wybranych danych w raportach.</p>	<p>Stosuje raporty przy użyciu narzędzia Autoraport.</p> <p>Stosuje, jak korzystać z kreatora raportów do generowania dowolnych rodzajów raportów.</p>

	<p>Potrafi grupować informacje w raporcie.</p> <p>Stosuje podsumowania statystyczne.</p> <p>Przygotowuje raport do druku.</p> <p>Definiuje makropolecenia złożone z kilku sekwencji akcji.</p> <p>Rozumie znaczenie komunikatu dotyczącego ostrzeżenia o zabezpieczeniach związanych z wyłączeniem określonej zawartości bazy danych.</p>
Ochrona bazy danych.	<p>Rozumie konieczność chronienia zgromadzonych danych przed osobami niepowołanymi do korzystania z nich i ich przetwarzania.</p> <p>Stosuje mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją.</p> <p>Stosuje efekt działania szyfrowania (kodowania) bazy danych.</p> <p>Rozumie konieczność wykonywania kopii bezpieczeństwa (na przykład przy wprowadzaniu i testowaniu kwerend funkcjonalnych).</p>

4. Multimedia i grafika komputerowa

Tabela 4. Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimediiów i grafiki komputerowej

TRĘŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
MULTIMEDIA I GRAFIKA KOMPUTEROWA	
Dźwięk w komputerze.	<p>Potrafi wyjaśnić typy plików dźwiękowych oraz ich zastosowanie.</p> <p>Potrafi wyjaśnić, jak powstał cyfrowy zapis dźwięku. Rozumie jego podstawowe parametry: częstotliwość próbkowania, poziomy kwantyzacji.</p> <p>Potrafi wyjaśnić metody kompresji plików dźwiękowych.</p> <p>Potrafi wyjaśnić konwersję między różnymi formatami dźwięku.</p> <p>Potrafi nagrać głos i specjalne efekty dźwiękowe.</p> <p>Potrafi zastosować filtry korygujące i wzbogacające dźwięk.</p> <p>Potrafi przeprowadzić edycję, miksowanie i montaż dźwięku, wykorzystując materiały własne oraz bazy gotowych efektów dźwiękowych.</p> <p>Posługuje się fachową terminologią.</p> <p>Zna i charakteryzuje sposoby reprezentowania dźwięku w komputerze.</p>
Grafika komputerowa.	<p>Potrafi wyjaśnić pojęcia związane z grafiką rastrową i wektorową.</p> <p>Objaśnia podstawowe różnice pomiędzy grafiką rastrową a wektorową.</p> <p>Zna wady i zalety grafiki rastrowej i wektorowej.</p>

	<p>Stosuje kompresję stratną i bezstratną plików graficznych oraz rozumie, jaki ma ona wpływ na wielkość plików i jakość obrazu.</p> <p>Potrafi wyjaśnić algorytmy kompresji: RLE, LZW i kodowanie Huffmana.</p> <p>Potrafi wyjaśnić sposoby reprezentowania obrazu w komputerze.</p> <p>Projektuje grafikę przeznaczoną na stronę WWW.</p> <p>Potrafi wyjaśnić zasadę tworzenia obrazu 3D.</p> <p>Potrafi wyjaśnić zasadę powstawania animacji.</p> <p>Tworzy własną animację.</p> <p>Rozumie, jaki wpływ na wydrukowaną grafikę mają wymiary obrazu, rozdzielczość obrazu i rozdzielczość wydruku.</p> <p>Rozumie pojęcia: wymiary obrazu i rozdzielczość obrazu.</p> <p>Wykonuje czynności edycyjne, uwzględniając warstwy, transformacje, filtry, zmianę rozmiaru obrazu, kadrowanie obrazu.</p> <p>Rozumie, w jaki sposób jest reprezentowany obraz w komputerze.</p> <p>Potrafi wyjaśnić modele barw i ich zastosowanie.</p> <p>Potrafi wyjaśnić własności barwy.</p> <p>Potrafi wyjaśnić jak działa skaner i jak powstaje cyfrowy obraz ze skanera.</p>
Cyfrowy zapis filmu.	<p>Rozumie pojęcie: kontener multimedialny.</p> <p>Wie, jak działają kodeki wideo na przykładzie pojedynczego obrazu.</p> <p>Potrafi zmontować film i opublikować go.</p>

5. Systemy operacyjne i sieci komputerowe

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
SYSTEMY OPERACYJNE	
Przegląd systemów operacyjnych.	<p>Potrafi wyjaśnić budowę komputera oraz rozumie rolę jego poszczególnych podzespołów.</p> <p>Potrafi wyjaśnić rolę BIOS-u i wie, na czym polega rozruch komputera.</p> <p>Wymienia podstawowe funkcje systemu operacyjnego:</p> <ul style="list-style-type: none"> — wykonywanie programów, — operacje wejścia-wyjścia, — zarządzanie systemem plików, — komunikacja, — przekazywanie komunikatów,

	<p>— wykrywanie błędów.</p> <p>Potrafi wyjaśnić budowę systemu operacyjnego.</p> <p>Potrafi wyjaśnić rodzaj systemów operacyjnych.</p> <p>Potrafi wyjaśnić pojęcia: system plików, klaster, sektor.</p> <p>Uruchamia maszynę wirtualną z systemem operacyjnym DR-DOS, Linux i wykonuje podstawowe operacje na plikach.</p> <p>Potrafi wyjaśnić pojęcia: środowisko wielodostępowe, wielozadaniowość, jądro systemu, GUI, maszyna wirtualna.</p>
Podstawowe zadania systemu operacyjnego.	<p>Zna zadania systemu operacyjnego, do których należą:</p> <ul style="list-style-type: none"> — zarządzanie procesami, — zarządzanie pamięcią operacyjną, — zarządzanie plikami, — zarządzanie systemem wejścia-wyjścia, — zarządzanie pamięcią pomocniczą (dyskową), — praca w sieci, — system ochrony. <p>Potrafi wyjaśnić, na czym polega i jak przebiega interpretacja poleceń (wiersz poleceń, powłoka).</p>
SIECI KOMPUTEROWE	
Budowa, działanie i eksploatacja sieci komputerowych.	<p>Potrafi wyjaśnić rodzaje sieci ze względu na ich zakresy: LAN, MAN, WAN.</p> <p>Potrafi wyjaśnić rodzaje sieci ze względu na topologię. Potrafi Rozróżnia podstawowe opisać topologie sieciowe (topologia magistrali, gwiazdy, pierścienia, pełnych połączeń, mieszana).</p> <p>Potrafi wyjaśnić urządzenia i elementy sieci komputerowych (karty sieciowe, okablowanie sieciowe, urządzenia sieciowe).</p> <p>Potrafi wyjaśnić technologie sieciowe (metody dostępu do sieci).</p> <p>Potrafi wyjaśnić rolę urządzeń służących do rozbudowy sieci (wzmacniaki, koncentratory, mosty, przełączniki, routery, bramy).</p> <p>Potrafi wyjaśnić zasady administrowania siecią komputerową.</p> <p>Potrafi wyjaśnić usługi oraz narzędzia sieciowe i korzysta z nich.</p>
Protokoły sieciowe. Model OSI.	<p>Potrafi wyjaśnić warstwowy model sieci komputerowych.</p> <p>Potrafi wyjaśnić, czym jest protokół i zestaw protokołów.</p> <p>Potrafi wyjaśnić funkcje poszczególnych warstw modelu OSI.</p> <p>Potrafi określić, czy dany komputer może się komunikować z innymi komputerami w sieci.</p> <p>Potrafi wyjaśnić protokołów sieciowych i opisać ich własności.</p>
Zestaw protokołów TCP/IP.	Potrafi wyjaśnić podział na warstwy zestawu protokołów TCP/IP i porównuje go z modelem OSI.

	<p>Potrafi wyjaśnić sposób przesyłania danych między protokołami TCP/IP.</p> <p>Potrafi wyjaśnić i stosuje pojęcie: pakiet danych.</p> <p>Szczegółowo analizuje warstwę sieciową, używając pojęć: protokół IP, datagram IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska podsieci, serwer DNS.</p> <p>Potrafi wyjaśnić ustawienia sieciowe danego komputera i jego lokalizację w sieci.</p>
Bezpieczeństwo informacji w sieciach.	<p>Potrafi wyjaśnić podstawowe zasady administrowania siecią komputerową w architekturze klient-serwer.</p> <p>Potrafi wyjaśnić zagrożenia związane z pracą komputera w sieci: niszczące programy i najczęstsze ataki, sniffer, podszywanie i naśladownictwo. Potrafi wyjaśnić i podjąć działania prewencyjne.</p> <p>Potrafi wyjaśnić sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie SSL, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zaporę sieciową, filtrowanie pakietów.</p> <p>Potrafi wyjaśnić konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów.</p> <p>Potrafi wyjaśnić zagrożenia związane z przestępczością komputerową, w tym z piractwem komputerowym i nielegalnymi transakcjami w sieci.</p> <p>Potrafi wyjaśnić kopie bezpieczeństwa danych.</p>
Tworzenie i publikowanie własnych materiałów w sieci.	<p>Potrafi wyjaśnić strukturę oraz podstawowe znaczniki dokumentu HTML.</p> <p>Potrafi stworzyć i modyfikować witrynę WWW z wykorzystaniem tekstu, tabel, odpowiednich formatów grafiki i opracowań multimedialnych.</p> <p>Potrafi wyjaśnić korzyści związane ze stosowaniem arkuszy stylów.</p> <p>Przygotowuje własne materiały (grafiki, galerie zdjęć, wykresy, prezentacje multimedialne, animacje, dźwięk), które zamieszcza w dokumencie HTML, dodając odpowiednie menu i odsyłacze.</p> <p>Publikuje w sieci własne materiały oraz opracowania multimedialne.</p> <p>Potrafi wyjaśnić i stosuje obowiązujące zasady prawne dotyczące prawa autorskiego.</p> <p>Potrafi wyjaśnić podstawy języka PHP i wykorzystuje go w dokumentach HTML.</p> <p>Planuje współpracę i zespołowo wykonuje projekt witryny internetowej.</p>

6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

Tabela 6. Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	
Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	Potrafi wyjaśnić przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie.

	<p>Potrafi wyjaśnić rodzaje licencji programów komputerowych.</p> <p>Przestrzega zasad etyki.</p> <p>Potrafi wyjaśnić bariery w korzystaniu z technologii informacyjnej przez różne grupy społeczne.</p> <p>Potrafi wyjaśnić korzyści i zagrożenia wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.</p>
Kierunki rozwoju informatyki i jej zastosowań.	<p>Potrafi wyjaśnić nowości i zna najnowsze osiągnięcia rozwoju technik informatycznych.</p> <p>Potrafi wyjaśnić możliwości współczesnych komputerów i tendencje w ich rozwoju.</p> <p>Śledzi na bieżąco i gromadzi informacje na temat zmian w rozwoju oprogramowania i systemów operacyjnych, ich nowych dystrybucji.</p> <p>Korzysta z zasobów edukacyjnych udostępnianych w portalach przeznaczonych do kształcenia na odległość.</p>

• **stopień bardzo dobry** otrzymuje uczeń, który: opanował cały zakres wiedzy i umiejętności określonym programem nauczania przedmiotu w danej klasie, sprawnie posługuje się zdobytymi wiadomościami, rozwiązuje samodzielnie problemy informatyczne.

Stopień bardzo dobry otrzymuje uczeń, który: posiada kompletną i spójną wiedzę teoretyczną w zakresie określonym wymaganiami programowymi, potrafi rozwiązać zadania wymagające zastosowania łącznej wiedzy i umiejętności z wielu działów przedmiotu, zna i rozumie działanie poszczególnych składników sprzętowych i zależności występujące między nimi, sprawnie posługuje się wszystkimi narzędziami i technikami programowymi, w swojej pracy jest systematyczny i zdyscyplinowany, wyróżnia się obowiązkowością i dokładnością w wypełnianiu przydzielonych mu zadań z zakresu informatyki lub komputerowego wspomaganie w ochronie środowiska.

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Rozumie pojęcie: algorytm. Formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Podaje algorytmy związane z życiem codziennym, innymi nauczanyimi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	Poprawnie definiuje problem i formułuje jego specyfikację. Stosuje podejście algorytmiczne do rozwiązywanego problemu. Zna i realizuje etapy rozwiązywania zadań za pomocą komputera (od sformułowania specyfikacji problemu po testowanie rozwiązania). Rozwiązuje problem zgodnie z etapami: zaplanowanie działań, wydzielenie podproblemów i określenie zależności między nimi, skonstruowanie algorytmów rozwiązujących podproblemy, analiza algorytmów i określenie ogólnego algorytmu rozwiązującego problem. Dokonuje analizy umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją. Ocenia zgodność algorytmu ze specyfikacją problemu.
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Prezentuje algorytmy, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji: <ul style="list-style-type: none"> • schemat blokowy — graficzna prezentacja algorytmu, • lista kroków — słowna prezentacja algorytmu, • program w wybranym języku programowania wysokiego poziomu — praktyczna realizacja algorytmu, • arkusz kalkulacyjny. 	Stosuje różne sposoby reprezentowania schematu blokowego, listę kroków, realizuje algorytm w języku programowania wysokiego poziomu, arkusza kalkulacyjnym. Specyfikuje składowe algorytmu. Stosuje sposoby rozwiązania i prezentacji algorytmu do konkretnego problemu. Rozwiązanie zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie notacji. Stosuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i zapisywania algorytmów. Stosuje informatyczne rozwiązanie algorytmu, dobiera struktury danych (w tym struktury dynamiczne) i możliwie najlepszy algorytm oraz zapisuje go w wybranym języku programowania. Stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu. Wykorzystuje różne metody algorytmiczne i algorytmy klasyczne do rozwiązywania problemu oraz świadomie

	<p>wybiera właściwy sposób rozwiązania zadania.</p> <p>Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania umiarkowanie złożonych zadań z różnych dziedzin.</p>
<p>Algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.</p>	<p>Stosuje algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami) przy rozwiązywaniu problemów.</p> <p>Stosuje algorytmy liniowe i warunkowe: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe itp.</p>
<p>Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.</p>	<p>Stosuje metodę iteracji przy realizacji algorytmów.</p> <p>Wykorzystuje metodę iteracji w przykładowych algorytmach.</p> <p>Stosuje iterację do zapisu algorytmów: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach (łańcuchach) oraz tablicach (na przykład znajdowanie podciągów o określonych własnościach) itp.</p>
<p>Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.</p>	<p>Rozumie pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna.</p> <p>Podaje przykłady zastosowania rekurencji oraz powiązać sytuacje życiowe z rozwiązaniami rekurencyjnymi.</p> <p>Definiuje zależności rekurencyjne oraz odpowiednie procedury-funkcje rekurencyjne.</p> <p>Dokonuje zamiany metody iteracyjnej wykorzystanej w algorytmie na rekurencyjną.</p> <p>Stosuje algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi itp. Potrafi zastosować iterację do realizacji tych algorytmów.</p> <p>Dokonuje analizy wartości zmiennych w przykładowych algorytmach rekurencyjnych.</p> <p>Stosuje rekurencję w prostych sytuacjach problemowych.</p>

• **stopień dobry** otrzymuje uczeń, który: nie opanował w pełni wiadomości określonych programem w danej klasie, ale opanował je na poziomie przekraczającym wymagania zawarte w podstawach programowych, poprawnie stosuje wiadomości, rozwiązuje (wykonuje) samodzielnie typowe zadania z zakresu informatyki

Stopień dobry otrzymuje uczeń, który: posiada pełną wiedzę teoretyczną w zakresie poruszonym na lekcjach, poza drobnymi wyjątkami potrafi rozwiązać problemy charakterystyczne dla każdej z omawianych dziedzin, sprawnie posługuje się sprzętem i narzędziami programowymi, jest samodzielny w poszukiwaniu rozwiązań, na lekcjach wykazuje twórczą i właściwie ukierunkowaną aktywność, jest zdyscyplinowany i uporządkowany w swojej pracy z zakresu informatyki lub komputerowego wspomaganie w ochronie środowiska.

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczonymi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	<p>Poprawnie definiuje problem i formułuje jego specyfikację.</p> <p>Stosuje podejście algorytmiczne do rozwiązywanego problemu.</p> <p>Realizuje etapy rozwiązywania zadań za pomocą komputera od sformułowania specyfikacji problemu po testowanie rozwiązania.</p> <p>Rozwiązuje problem zgodnie z etapami: zaplanowanie działań, wydzielenie podproblemów i określenie zależności między nimi, skonstruowanie algorytmów rozwiązujących podproblemy, analiza tych algorytmów i określenie ogólnego algorytmu rozwiązującego problem.</p> <p>Dokonuje analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją.</p> <p>Ocenia zgodność algorytmu ze specyfikacją problemu.</p>
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Zna różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
<p>Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji:</p> <ul style="list-style-type: none"> • schemat blokowy — graficzna prezentacja algorytmu, • lista kroków — słowna prezentacja algorytmu, • program w wybranym języku programowania wysokiego poziomu — praktyczna realizacja algorytmu, • arkusz kalkulacyjny. 	<p>Zapisuje algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków, program w języku programowania wysokiego poziomu, arkusz kalkulacyjny.</p> <p>Zna elementy składowe algorytmu.</p> <p>Stosuje właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu.</p> <p>Zapisuje rozwiązanie zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie notacji.</p> <p>Wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i zapisywania algorytmów.</p> <p>Formułuje informatyczne rozwiązanie algorytmu, dobiera odpowiednie struktury danych i możliwie najlepszy algorytm oraz zapisuje go w wybranym języku programowania.</p> <p>Wykorzystuje różne metody algorytmiczne i algorytmy klasyczne do rozwiązywania problemu oraz świadomie</p>

	<p>wybiera właściwy sposób rozwiązania zadania.</p> <p>Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych i umiarkowanie złożonych zadań z różnych dziedzin.</p>
<p>Algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.</p>	<p>Stosuje algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami) przy rozwiązywaniu problemów.</p> <p>Rozumie wykorzystaną metodę programowania liniowego i z warunkami w przykładowych algorytmach.</p> <p>Zapisuje i stosuje algorytmy liniowe i warunkowe: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe itp.</p>
<p>Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.</p>	<p>Zna i rozumie metody iteracji przy realizacji algorytmów.</p> <p>Zna metody iteracji w przykładowych algorytmach np.: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach (łańcuchach) oraz tablicach..</p>
<p>Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.</p>	<p>Rozumie pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna.</p> <p>Potrafi podać przykłady zastosowania rekurencji oraz powiązać sytuacje życiowe z rozwiązaniami rekurencyjnymi.</p> <p>Zna i rozumie zależności rekurencyjne oraz odpowiednie procedury-funkcje rekurencyjne.</p> <p>Zna i rozumie zamiany metody iteracyjnej wykorzystanej w algorytmie na rekurencyjną.</p> <p>Zna i rozumie algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi itp. Potrafi zastosować iterację do realizacji tych algorytmów.</p> <p>Zna i rozumie analizy wartości zmiennych w przykładowych algorytmach rekurencyjnych.</p> <p>Stosuje rekurencję w prostych sytuacjach problemowych.</p>
<p>Metoda „dziel i zwyciężaj”.</p>	<p>Rozumie i podaje pojęcie: metoda „dziel i zwyciężaj”.</p> <p>Rozumie i definiuje metodę „dziel i zwyciężaj” w odpowiednich sytuacjach.</p> <p>Rozumie i podaje algorytmy z wykorzystaniem metody „dziel i zwyciężaj”.</p>
<p>Programowanie zachłanne.</p>	<p>Rozumie i definiuje pojęcia: programowanie zachłanne.</p> <p>Rozumie i definiuje metodę programowania zachłannego w przykładowych algorytmach.</p> <p>Rozumie i definiuje podejście zachłanne w rozwiązywaniu problemów.</p>
<p>Kryptografia i kryptoanaliza. Metody szyfrowania.</p>	<p>Wyjaśnia podstawowe metody szyfrowania.</p>
<p>ALGORYTMY I ICH ZASTOSOWANIE</p>	
<p>Algorytmy badające własności geometryczne:</p> <ul style="list-style-type: none"> • sprawdzanie warunku trójkąta, • określanie prostokątności i równoległości prostych, • obliczanie odległości punktu od prostej, 	<p>Rozumie i definiuje prostą w postaci ogólnej i kierunkowej: na podstawie wzorów wyznacza odległość punktu od prostej oraz odległość punktów na płaszczyźnie; na podstawie równań kierunkowych określa prostokątność i równoległość prostych.</p> <p>Rozumie i definiuje algorytm sprawdzający warunek w języku programowania.</p>

<ul style="list-style-type: none"> • obliczanie odległości punktów na płaszczyźnie, • badanie położenia punktu względem prostej, • badanie przynależności punktu do odcinka. 	
<p>Wyznaczanie największego wspólnego dzielnika i najmniejszej wspólnej wielokrotności dwóch liczb naturalnych:</p> <ul style="list-style-type: none"> • wyznaczanie największego wspólnego dzielnika — algorytm Euklidesa, • wyznaczanie najmniejszej wspólnej wielokrotności. 	<p>Rozumie i definiuje największy wspólny dzielnik dwóch liczb naturalnych algorytmem Euklidesa. Stosuje iteracyjną i rekurencyjną realizację algorytmu.</p> <p>Rozumie i definiuje najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający największy wspólny dzielnik.</p>
<p>Wyznaczanie wartości wielomianu, pozycyjne systemy liczbowe i reprezentacja danych liczbowych w komputerze:</p> <ul style="list-style-type: none"> • wyznaczanie wartości wielomianu schematem Hornera, • szybkie podnoszenie do potęgi, • pozycyjne systemy liczbowe stosowane w informatyce — system dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny) — definicja, zamiana liczb zapisanych w podanych systemach, realizacja podstawowych operacji arytmetycznych w podanych systemach, 	<p>Rozumie i definiuje algorytm obliczający wartość wielomianu za pomocą schematu Hornera.</p> <p>Rozumie i definiuje algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera.</p> <p>Rozumie i definiuje pojęcia: system liczbowy, pozycyjny system liczbowy.</p> <p>Rozumie i definiuje znaczenie reprezentacji liczb w dowolnym systemie pozycyjnym, w tym dwójkowym (binarnym), ósemkowym (oktalnym) i szesnastkowym (heksadecymalnym), oraz zamienia liczby zapisane w tych systemach liczbowych na system dziesiętny i odwrotnie.</p> <p>Rozumie i definiuje liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera.</p> <p>Rozumie i definiuje operacje arytmetyczne w dowolnych pozycyjnych systemach liczbowych.</p> <p>Rozumie i definiuje reprezentacje danych liczbowych w komputerze, źródło błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.</p>
<ul style="list-style-type: none"> • zamiana liczb z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera, • zamiana liczb z systemu dziesiętnego na dowolny pozycyjny system liczbowy, • wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych, • reprezentacja danych liczbowych w komputerze — reprezentacja binarna liczb całkowitych i niecałkowitych, reprezentacja stałopozycyjna i zmiennopozycyjna liczb, błędy w obliczeniach. 	
<p>Liczby pierwsze:</p>	<p>Wyjaśnia pojęcie: liczba pierwsza.</p>

<ul style="list-style-type: none"> • sprawdzanie, czy liczba jest liczbą pierwszą, • rozkładanie liczby na czynniki pierwsze, • sito Eratostenesa. 	<p>Rozumie i definiuje algorytm sprawdzający, czy dana liczba jest pierwsza.</p> <p>Rozumie i definiuje algorytm generujący liczby pierwsze — sito Eratostenesa.</p>
<p>Przeszukiwanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • przeszukiwanie liniowe tablicy jednowymiarowej, • przeszukiwanie liniowe tablicy jednowymiarowej z wartownikiem. 	<p>Rozumie i definiuje metody liniowe przy przeszukiwaniu ciągu liczbowego, tablicy jednowymiarowej.</p>
<p>Znajdowanie największego lub najmniejszego elementu w ciągu liczbowym:</p> <ul style="list-style-type: none"> • znajdowanie największego elementu w ciągu liczbowym, • znajdowanie najmniejszego elementu w ciągu liczbowym. 	<p>Rozumie i definiuje algorytm znajdowania największego elementu w ciągu liczbowym.</p> <p>Rozumie i definiuje algorytm znajdowania najmniejszego elementu w ciągu liczbowym.</p>
<p>Znajdowanie lidera w zbiorze.</p>	<p>Rozumie i definiuje pojęcie: lider w zbiorze.</p> <p>Rozumie i definiuje algorytm znajdowania lidera w zbiorze.</p>
<p>Sprawdzanie monotoniczności ciągu liczbowego:</p> <ul style="list-style-type: none"> • sprawdzanie, czy ciąg liczbowy jest rosnący lub malejący, • sprawdzanie, czy ciąg liczbowy jest nierosnący lub niemalejący, • sprawdzanie, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny. 	<p>Rozumie i definiuje pojęcie: monotoniczność ciągu liczbowego.</p> <p>Rozumie i definiuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący lub malejący.</p> <p>Rozumie i definiuje algorytm sprawdzający, czy ciąg liczbowy jest nierosnący lub niemalejący.</p> <p>Rozumie i definiuje algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.</p>
<p>Liniowe sortowanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • porządkowanie przez wybór, • porządkowanie przez wstawianie, • porządkowanie bąbelkowe. 	<p>Rozumie i definiuje wyjaśnienie algorytmu porządkowania przez wybór.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu porządkowania przez wstawianie.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu porządkowania bąbelkowego.</p>
<p>Zastosowanie metody „dziel i zwyciężaj”:</p> <ul style="list-style-type: none"> • jednoczesne znajdowanie najmniejszego i największego elementu, • przeszukiwanie binarne ciągu uporządkowanego, 	<p>Rozumie i definiuje wyjaśnienie algorytmu metody „dziel i zwyciężaj” przy realizacji algorytmów.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu jednoczesnego znajdowania najmniejszego i największego elementu.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu przeszukiwania binarnego uporządkowanego ciągu liczbowego.</p>
<ul style="list-style-type: none"> • sortowanie przez scalanie, • sortowanie szybkie, • znajdowanie przybliżonej wartości miejsca zerowego 	<p>Rozumie i definiuje wyjaśnienie algorytmu sortowania przez scalanie.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu sortowania szybkiego.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej —</p>

funkcji ciągłej — metoda połowienia przedziałów.	metodą połowienia przedziałów.
<p>Obliczenia przybliżone. Realizacja wybranych algorytmów numerycznych:</p> <ul style="list-style-type: none"> • obliczanie wartości pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona), • obliczanie pola obszaru ograniczonego wykresem funkcji (całkowanie numeryczne), • przybliżanie wartości liczby metodą Monte Carlo. 	<p>Rozumie i definiuje wyjaśnienie pojęcia: obliczenia przybliżone, metody numeryczne.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu obliczającego wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona).</p> <p>Rozumie i definiuje wyjaśnienie algorytmu obliczającego pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów.</p> <p>Rozumie i definiuje wyjaśnienie algorytmu wykonującego przybliżanie wartości liczby metodą Monte Carlo.</p>
<p>Algorytmy na tekstach (łańcuchach):</p> <ul style="list-style-type: none"> • sprawdzanie, czy tekst jest palindromem, • porządkowanie alfabetyczne tekstu, • sprawdzanie, czy teksty są anagramami, • wyszukiwanie wzorca w tekście, • wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP. 	<p>Rozumie i definiuje wyjaśnienie algorytmu wykonywanego na tekstach, na przykład sprawdzanie, czy tekst jest palindromem, porządkowanie alfabetyczne tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście.</p> <p>Rozumie i definiuje wyjaśnienie notacji wyrażenia podanego w postaci ONP. Zapisuje i stosuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.</p>
<p>Zastosowanie programowania zachłannego:</p> <ul style="list-style-type: none"> • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana. 	<p>Rozumie i definiuje wyjaśnienie algorytmów: algorytm zachłanny dla problemu plecakowego, algorytm wydawania reszty, algorytm Huffmana.</p>
Wybrane algorytmy kryptograficzne.	<p>Rozumie i definiuje wyjaśnienie algorytmów np.: kryptograficzne, w tym algorytmy symetryczne (np. szyfr Cezara, szyfr przestawieniowy, szyfr plotowy, szyfry monoalfabetyczne i wieloalfabetyczne), algorytmy asymetryczne (np. szyfr z kluczem jawnym RSA), podpis cyfrowy itp.</p>
WŁASNOŚCI ALGORYTMÓW	
<p>Złożoność czasowa algorytmów. Analiza złożoności czasowej wybranych algorytmów.</p>	<p>Rozumie i definiuje wyjaśnienie pojęcia: złożoność obliczeniowa, złożoność czasowa.</p> <p>Rozumie i definiuje wyjaśnienie złożoność czasową algorytmów.</p> <p>Analizuje złożoność czasową wybranych algorytmów (oblicza liczbę operacji dominujących wykonywanych przez algorytm).</p> <p>Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem.</p> <p>Rozumie i definiuje wyjaśnienie możliwie najszybszy algorytm rozwiązujący postawiony problem.</p>
<p>Złożoność pamięciowa algorytmów. Analiza złożoności pamięciowej wybranych algorytmów.</p>	<p>Rozumie i definiuje wyjaśnienie pojęcia: złożoność pamięciowa.</p> <p>Potrąfi wyznaczać, oceniać oraz uzasadniać złożoność pamięciową algorytmów.</p>

	<p>Analizuje złożoność pamięciową wybranych algorytmów (szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu).</p> <p>Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera odpowiednie struktury danych w rozwiązaniu postawionego problemu.</p> <p>Rozumie i definiuje wyjaśnienie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu.</p> <p>Opisuje własności algorytmów na podstawie ich analizy.</p> <p>Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności.</p> <p>Bada efektywność komputerowych rozwiązań problemów.</p> <p>Ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.</p>
--	---

2. Programowanie

Tabela 2. Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
Języki programowania — pojęcie, klasyfikacja, przykłady. Prezentacja przykładowych programów w różnych językach programowania. Podstawowe zasady i metody programowania. Znajomość pojęć: translator, kompilator, interpreter, debugger, linker.	Rozumie i definiuje wyjaśnienie pojęcia: język programowania, translator, kompilator, interpreter, debugger, linker. Rozumie i definiuje wyjaśnienie: klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego i wysokiego poziomu. i metody programowania.
Elementy zintegrowanego systemu programowania, w tym umiejętność pisania, kompilowania i uruchamiania programów za pomocą wybranego kompilatora.	Zna wybrane środowiska programistycznego (na przykład kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.
Wprowadzenie do programowania: <ul style="list-style-type: none"> • struktura programu, • operacje wejścia-wyjścia, • zmienne i ich deklaracja, • stałe i ich deklaracja, • wyrażenia arytmetyczne, relacje i operatory logiczne, • priorytety relacji i działań, • komentarze. 	Zna strukturę programu, operacje wejścia i wyjścia. Potrafi deklorować stałe i zmienne i wykorzystywać je w programach. Rozumie i definiuje wyjaśnienie wyrażenia arytmetyczne, relacje i operatory logiczne oraz stosuje je w programach. Rozumie i definiuje wyjaśnienie priorytetów relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je przy pisaniu programów. Stosuje komentarze przy pisaniu programów.
Podstawowe konstrukcje algorytmiczne:	Stosuje podstawowe konstrukcje algorytmiczne, w tym instrukcję przypisania, instrukcję złożoną, instrukcje

<ul style="list-style-type: none"> • instrukcja przypisania, • instrukcja złożona, • instrukcje warunkowe, • instrukcja wyboru, • instrukcje iteracyjne. <p>Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.</p> <p>Proste typy danych:</p> <ul style="list-style-type: none"> • typy całkowite, • typy rzeczywiste, • typ znakowy, • typ logiczny. <p>Realizacja operacji z wykorzystaniem prostych typów danych oraz ich zastosowanie w programach.</p>	<p>warunkowe, instrukcję wyboru, instrukcje iteracyjne.</p> <p>Stosuje w programach powyższe konstrukcje algorytmiczne.</p> <p>Rozumie i definiuje wyjaśnienie pojęcia: proste typy danych.</p> <p>Stosuje typy danych, w tym typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny.</p> <p>Stosuje operacje na prostych typach danych.</p> <p>Potrafi w danej sytuacji problemowej wybrać właściwy typ danych dla wykorzystywanych zmiennych.</p>
<p>Strukturalizacja programu:</p> <ul style="list-style-type: none"> • definiowanie procedur-funkcji, • zmienne lokalne i globalne, • parametry formalne i aktualne, • przekazywanie parametrów w procedurach-funkcjach, • przeciążanie funkcji. <p>Zastosowanie w implementacjach procedur-funkcji z parametrami. Umiejętność wyboru właściwego w danej sytuacji sposobu przekazywania parametrów.</p>	<p>Rozumie i definiuje wyjaśnienie zasad i celów strukturalizacji programu.</p> <p>Potrafi wydzielać fragmenty programu i definiować procedury-funkcje.</p> <p>Rozumie i wykorzystuje różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach.</p> <p>Rozumie i wykorzystuje różnice między parametrami formalnymi i aktualnymi.</p> <p>Rozumie i wykorzystuje sposoby przekazywania parametrów w procedurach-funkcjach i świadomie stosuje je podczas realizacji programów.</p> <p>Rozumie i wykorzystuje sposoby przeciążania funkcji i stosuje je w programach.</p>
<p>Strukturalne typy danych:</p> <ul style="list-style-type: none"> • łańcuchy (w tym operacje na tekście), • tablice (w tym jednowymiarowe i wielowymiarowe), • rekordy-struktury (w tym tablice rekordów-struktur), • struktury dynamiczne (w tym listy, stosy, kolejki, drzewa binarne). <p>Realizacja programów z wykorzystaniem strukturalnych typów danych.</p>	<p>Rozumie i wykorzystuje pojęcia: strukturalne typy danych, abstrakcyjne typy danych.</p> <p>Potrafi definiować typ łańcuchowy.</p> <p>Rozumie i wykorzystuje predefiniowane operacje na tekście i z nich korzysta.</p> <p>Zapisuje programy z wykorzystaniem typu łańcuchowego.</p> <p>Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe.</p> <p>Zapisuje programy z wykorzystaniem typu tablicowego.</p> <p>Definiuje typ rekordowy-strukturalny.</p> <p>Zapisuje programy z wykorzystaniem typu rekordowego-strukturalnego, w tym tablic rekordów-struktur.</p> <p>Potrafi definiować dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne.</p>

	<p>Zapisuje programy z wykorzystaniem dynamicznych struktur danych.</p> <p>Dobiera struktury danych (w tym dynamiczne struktury danych) odpowiednio do przetwarzanych informacji.</p> <p>Wykorzystuje typy strukturalne, w tym typ łańcuchowy, tablicowy i rekordowy-strukturalny, oraz dynamiczne struktury danych do implementacji algorytmów.</p>
<p>Plikowe operacje wejścia-wyjścia:</p> <ul style="list-style-type: none"> • rodzaje plików, • definicja i etapy przetwarzania plików, • podstawowe operacje na plikach. <p>Realizacja programów z zastosowaniem operacji plikowych.</p>	<p>Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne.</p> <p>Zna i realizuje etapy przetwarzania plików.</p> <p>Zna operacje na plikach i z nich korzysta.</p> <p>Zapisuje programy z wykorzystaniem typu plikowego, w tym importuje dane z plików zewnętrznych, eksportuje wyniki do plików zewnętrznych.</p> <p>Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach.</p>
<p>Programowanie zorientowane obiektowo:</p> <ul style="list-style-type: none"> • klasy i obiekty, • dziedziczenie i hierarchia klas, • konstruktory i destruktory, • polimorfizm, • metody wirtualne. <p>Realizacja implementacji z zastosowaniem programowania zorientowanego obiektowo.</p>	<p>Rozumie i wykorzystuje pojęcia: programowanie obiektowe, klasa, obiekt, enkapsulacja, dziedziczenie, hierarchia klas, polimorfizm, konstruktor, destruktor.</p> <p>Definiuje typ obiektowy.</p> <p>Tworzy proste klasy i statyczne obiekty.</p> <p>Stosuje metody konstruktora i destruktora.</p> <p>Projektuje i realizuje hierarchię klas.</p> <p>Potrafi definiować i stosować w programach dynamiczne obiekty.</p> <p>Stosuje metody wirtualne.</p> <p>Realizuje programy z wykorzystaniem typu obiektowego.</p>
<p>Projekt programistyczny.</p>	<p>Zna pojęcie: inżynieria oprogramowania.</p> <p>Stosuje fazy konstruowania oprogramowania.</p> <p>Stosuje informacje niezbędne do realizacji projektów z różnych dziedzin.</p> <p>Realizuje indywidualnie lub zespołowo projekt programistyczny, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.</p>

3. Bazy danych

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
BAZODANOWE FUNKCJE ARKUSZA KALKULACYJNEGO	
<p>Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.</p>	<p>Zna i rozumie pojęcie: baza danych.</p> <p>Zna i stosuje reguły, jakim podlega tabela stanowiąca źródło danych.</p> <p>Potrafi zgromadzić w pliku tekstowym dane będące bazą danych.</p>

	<p>Importuje dane umieszczone w pliku tekstowym do tabeli arkusza kalkulacyjnego.</p> <p>Zapisuje zgromadzone w tabeli dane w pliku tekstowym, stosując odpowiednie znaki separacji.</p>
<p>Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.</p>	<p>Korzysta z autofiltru w celu wyselekcjonowania danych.</p> <p>Potrafi stosować złożone kryteria wyboru w filtrach zaawansowanych.</p> <p>Stosuje odpowiedni rodzaj filtru w celu wybrania informacji potrzebnych w danej sytuacji.</p>
<p>Tworzenie podsumowań danych.</p>	<p>Potrafi zastosować funkcje standardowe arkusza do podsumowań danych.</p> <p>Wyjaśnia pojęcie: suma pośrednia.</p> <p>Stosuje sumy pośrednie.</p> <p>Potrafi dokonać podsumowań danych zawartych w tabeli przy użyciu kreatora sum częściowych.</p> <p>Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich.</p> <p>Wyjaśnia pojęcie: tabela przestawna.</p> <p>Stosuje tabele przestawne.</p> <p>Wskazuje, do jakich danych w celu ich podsumowania można zastosować tabele przestawne.</p> <p>Stosuje zestawienie podsumowań danych przy użyciu kreatora tabel przestawnych.</p> <p>Stosuje ustawienia w istniejącej tabeli przestawnej.</p>
<p>Graficzna prezentacja danych na wykresach.</p>	<p>Stosuje typów wykresu do prezentowanych na nim danych.</p> <p>Umie tworzyć różnego typu wykresy do danych zawartych w tabelach i tabelach przestawnych.</p> <p>Stosuje opcje wykresu i formatować poszczególne jego elementy.</p> <p>Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauki szkolnej (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).</p>
<p>RELACYJNA BAZA DANYCH</p>	
<p>Podstawy relacyjnej bazy danych.</p>	<p>Stosuje proste i złożone.</p> <p>Stosuje pojęcia: system zarządzania bazą danych, relacyjna baza danych.</p> <p>Stosuje zasady projektowania bazy danych. Rozumie potrzebę unikania nadmiarowości danych.</p> <p>Stosuje pola i rekordy oraz jaką rolę pełni klucz w bazie danych.</p> <p>Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makra).</p>
<p>Tabele w relacyjnej bazie danych i relacje między nimi.</p>	<p>Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego.</p> <p>Potrafi zaprojektować układ tabel, unikając powtarzania danych.</p> <p>Tworzy tabele przy użyciu kreatora tabel oraz w widoku projektu.</p> <p>Potrafi ustalić właściwości pól. Zna i poprawnie stosuje typy danych w tabelach.</p> <p>Ustala właściwości pól tabeli, takie jak rozmiar pola, maska wprowadzania czy reguła sprawdzania poprawności.</p>

	<p>Rozumie, jaką rolę w bazie danych pełni indeks.</p> <p>Stosuje pojęcia: klucz główny oraz klucz obcy.</p> <p>Stosuje związki między tabelami.</p> <p>Rozróżnia typy relacji między tabelami, potrafi je nazwać.</p> <p>Kontroluje i zmienia warunki wymuszania więzów integralności podczas edytowania właściwości określonej relacji.</p> <p>Potrafi tworzyć i modyfikować tabele, uwzględniając różne typy danych oraz ustalone dla nich właściwości pól (rozmiar, ograniczenia, wymagania, komunikaty dla użytkownika itp.).</p> <p>Planuje współpracę i zespołowe wykonanie projektu relacyjnej bazy danych.</p>
<p>Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.</p>	<p>Stosuje formaty danych.</p> <p>Stosuje tabele baz danych.</p> <p>Stosuje dane bezpośrednio do tabeli lub poprzez formularz.</p> <p>Stosuje formularze do wprowadzania, wyszukiwania i edycji danych.</p> <p>Stosuje tabele i przeglądać wybrane rekordy przez sortowanie jedno- i wieloparametrowe.</p> <p>Zmienia sposób prezentowania danych poprzez ich sortowanie lub filtrowanie.</p> <p>Drukuje tabele i formularze.</p> <p>Tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu.</p> <p>Stosuje etykiety, formanty, nagłówek lub stopkę do formularza, zmienia jego szatę graficzną.</p>
<p>Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend.</p>	<p>Zna podstawowe typy kwerend.</p> <p>Samodzielnie projektuje zapytania, korzystając z widoku projektu oraz kreatorów różnego typu kwerend wybierających dane z jednej, kilku tabel lub z wcześniej zdefiniowanych kwerend.</p> <p>Stosuje w zapytaniach znaki wieloznaczne (zastępcze).</p> <p>Konstruuje pola obliczeniowe, wprowadza korekty.</p> <p>Określa złożone kryteria liczbowe i definiuje wyrażenia.</p> <p>Stosuje odpowiednie kryteria i parametry do wyszukiwania danych.</p> <p>Wprowadza wiele kryteriów wyboru w jednym polu, stosując odpowiednie operatory.</p> <p>Tworzy kwerendy obliczające podsumowania.</p> <p>Tworzy zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn.</p> <p>Tworzy kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane.</p> <p>Zmienia właściwości sprzężenia między tabelami, na podstawie, których tworzona jest kwerenda.</p> <p>Analizuje kod zapisany w języku zapytań SQL.</p> <p>Modyfikując kryteria zapytania, wprowadza zmiany w kodzie języka zapytań SQL.</p> <p>Stosuje język zapytań SQL pozwalające na tworzenie i modyfikowanie kwerend wybierających, parametrycznych,</p>

	krzyżowych i modyfikujących tabele.
Przygotowanie zestawień wybranych danych w raportach.	Stosuje raporty przy użyciu narzędzia Autoraport. Stosuje kreatory raportów do generowania dowolnych rodzajów raportów. Potrafi grupować informacje w raporcie. Stosuje podsumowania statystyczne. Przygotowuje raport do druku. Definiuje makropolecenia złożone z kilku sekwencji akcji.
Ochrona bazy danych.	Rozumie konieczność chronienia zgromadzonych danych przed osobami niepowołanymi do korzystania z nich i ich przetwarzania. Potrafi stosować podstawowe mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją. Wyjaśnia działania szyfrowania (kodowania) bazy danych. Rozumie konieczność wykonywania kopii bezpieczeństwa (na przykład przy wprowadzaniu i testowaniu kwerend funkcjonalnych).

4. Multimedia i grafika komputerowa

Tabela 4. Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimediiów i grafiki komputerowej

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
MULTIMEDIA I GRAFIKA KOMPUTEROWA	
Dźwięk w komputerze.	Stosuje typy plików dźwiękowych oraz ich zastosowanie. Wyjaśnia cyfrowy zapis dźwięku, parametry: częstotliwość próbkowania, poziomy kwantyzacji. Wyjaśnia i stosuje metody kompresji plików dźwiękowych. Wyjaśnia i stosuje konwersję między różnymi formatami dźwięku. Potrafi nagrać głos i specjalne efekty dźwiękowe. Wyjaśnia i stosuje filtry korygujące i wzbogacające dźwięk, Wykorzystuje miksowanie i montaż dźwięku, wykorzystując materiały własne oraz bazy gotowych efektów dźwiękowych. Posługuje się fachową terminologią. Wyjaśnia i stosuje sposoby reprezentowania dźwięku w komputerze.
Grafika komputerowa.	Wyjaśnia i stosuje pojęcia związane z grafiką rastrową i wektorową. Wyjaśnia i stosuje różnice pomiędzy grafiką rastrową a wektorową.

	<p>Wyjaśnia i stosuje wady i zalety grafiki rastrowej i wektorowej.</p> <p>Wyjaśnia i stosuje pliki graficzne w różnych formatach.</p> <p>Stosuje kompresję stratną i bezstratną plików graficznych oraz rozumie, jaki ma ona wpływ na wielkość plików i jakość obrazu.</p> <p>Wyjaśnia i stosuje algorytmy kompresji: RLE, LZW i kodowanie Huffmana.</p> <p>Wyjaśnia i stosuje sposoby reprezentowania obrazu w komputerze.</p> <p>Projektuje grafikę przeznaczoną na stronę WWW.</p> <p>Potrafi wyjaśnić zasadę tworzenia obrazu 3D.</p> <p>Wyjaśnia i stosuje zasadę powstawania animacji.</p> <p>Tworzy własną animację.</p> <p>Rozumie, jaki wpływ na wydrukowaną grafikę mają wymiary obrazu, rozdzielczość obrazu i rozdzielczość wydruku.</p> <p>Rozumie pojęcia: wymiary obrazu i rozdzielczość obrazu.</p> <p>Wykonuje czynności edycyjne, uwzględniając warstwy, transformacje, filtry, zmianę rozmiaru obrazu, kadrowanie obrazu.</p> <p>Wyjaśnia i stosuje sposób jest reprezentowany obraz w komputerze.</p> <p>Wyjaśnia i stosuje modele barw i ich zastosowanie.</p> <p>Wyjaśnia i stosuje własności barwy.</p> <p>Wyjaśnia i stosuje skaner i jak powstaje cyfrowy obraz ze skanera.</p>
Cyfrowy zapis filmu.	<p>Wyjaśnia i stosuje pojęcie: kontener multimedialny.</p> <p>Potrafi zmontować film i opublikować go.</p>

5. Systemy operacyjne i sieci komputerowe

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
SYSTEMY OPERACYJNE	
Przegląd systemów operacyjnych.	<p>Zna budowę komputera oraz rozumie rolę jego poszczególnych podzespołów.</p> <p>Rozumie rolę BIOS-u i wie, na czym polega rozruch komputera.</p> <p>Wyjaśnia funkcje systemu operacyjnego:</p> <ul style="list-style-type: none"> — wykonywanie programów, — operacje wejścia-wyjścia, — zarządzanie systemem plików,

	<ul style="list-style-type: none"> — komunikacja, — przekazywanie komunikatów, — wykrywanie błędów. <p>Wyjaśnia budowę systemu operacyjnego. Rozróżnia podstawowe systemy operacyjne. Wyjaśnia różne systemy operacyjne. Rozumie pojęcia: system plików, klaster, sektor. Uruchamia maszynę wirtualną z systemem operacyjnym DR-DOS, Linux i wykonuje podstawowe operacje na plikach. Wyjaśnia pojęcia: środowisko wielodostępowe, wielozadaniowość, jądro systemu, GUI, maszyna wirtualna.</p>
Podstawowe zadania systemu operacyjnego.	<p>Wyjaśnia zadania systemu operacyjnego, do których należą:</p> <ul style="list-style-type: none"> — zarządzanie procesami, — zarządzanie pamięcią operacyjną, — zarządzanie plikami, — zarządzanie systemem wejścia-wyjścia, — zarządzanie pamięcią pomocniczą (dyskową), — praca w sieci, — system ochrony. <p>Rozumie, na czym polega i jak przebiega interpretacja poleceń (wiersz poleceń, powłoka).</p>
SIECI KOMPUTEROWE	
Budowa, działanie i eksploatacja sieci komputerowych.	<p>Wyjaśnia rodzaje sieci ze względu na ich zakresy: LAN, MAN, WAN.</p> <p>Wyjaśnia rodzaje sieci ze względu na topologię. Potrafi opisać topologie sieciowe (topologia magistrali, gwiazdy, pierścienia, pełnych połączeń, mieszana).</p> <p>Wyjaśnia urządzenia i elementy sieci komputerowych (karty sieciowe, okablowanie sieciowe, urządzenia sieciowe).</p> <p>Potrafi opisać technologie sieciowe (metody dostępu do sieci).</p> <p>Potrafi opisać rolę urządzeń służących do rozbudowy sieci (wzmacniaki, koncentratory, mosty, przełączniki, routery, bramy).</p> <p>Wyjaśnia zasady administrowania siecią komputerową.</p> <p>Wyjaśnia usługi oraz narzędzia sieciowe i korzysta z nich.</p>
Protokoły sieciowe. Model OSI.	<p>Wyjaśnia warstwowy model sieci komputerowych.</p> <p>Wyjaśnia protokół i zestaw protokołów.</p> <p>Wyjaśnia funkcje poszczególnych warstw modelu OSI.</p> <p>Wyjaśnia komunikowanie z innymi komputerami w sieci.</p>

	Wyjaśnia nazwy protokołów sieciowych i opisać ich własności.
Zestaw protokołów TCP/IP.	Wyjaśnia podział na warstwy zestawu protokołów TCP/IP i porównuje go z modelem OSI. Omawia sposób przesyłania danych między protokołami TCP/IP. Zna i stosuje pojęcie: pakiet danych. Szczegółowo analizuje warstwę sieciową, używając pojęć: protokół IP, datagram IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska podsieci, serwer DNS. Wyjaśnia ustawienia sieciowe danego komputera i jego lokalizację w sieci.
Bezpieczeństwo informacji w sieciach.	Wyjaśnia zasady administrowania siecią komputerową w architekturze klient-serwer. Zna zagrożenia związane z pracą komputera w sieci: niszczące programy i najczęstsze ataki, sniffer, podszywanie i naśladownictwo. Potrafi podjąć działania prewencyjne. Zna różne sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie SSL, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zaporę sieciową, filtrowanie pakietów. Wyjaśnia konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów. Zna zagrożenia związane z przestępczością komputerową, w tym z piractwem komputerowym i nielegalnymi transakcjami w sieci. Potrafi wykonać kopie bezpieczeństwa danych. Zna podstawy kodowania danych.
Tworzenie i publikowanie własnych materiałów w sieci.	Wyjaśnia strukturę oraz podstawowe znaczniki dokumentu HTML. Potrafi stworzyć i modyfikować witrynę WWW z wykorzystaniem tekstu, tabel, odpowiednich formatów grafiki i opracowań multimedialnych. Wyjaśnia korzyści związane ze stosowaniem arkuszy stylów. Przygotowuje własne materiały (grafiki, galerie zdjęć, wykresy, prezentacje multimedialne, animacje, dźwięk), które zamieszcza w dokumencie HTML, dodając odpowiednie menu i odsyłacze. Publikuje w sieci własne materiały oraz opracowania multimedialne. Zna i stosuje obowiązujące zasady prawne dotyczące prawa autorskiego. Zna język PHP i wykorzystuje go w dokumentach HTML. Planuje współpracę i zespołowo wykonuje projekt witryny internetowej.

6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

Tabela 6. Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki

TREŚCI	OSIĄGNIĘCIA UCZNIA
--------	--------------------

NAUCZANIA	
ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	
Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	<p>Wyjaśnia przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie.</p> <p>Wyjaśnia rodzaje licencji programów komputerowych.</p> <p>Przestrzega zasad etyki.</p> <p>Wyjaśnia bariery w korzystaniu z technologii informacyjnej przez różne grupy społeczne.</p> <p>Wyjaśnia korzyści i zagrożenia wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.</p>
Kierunki rozwoju informatyki i jej zastosowań.	<p>Śledzi na bieżąco nowości i zna najnowsze osiągnięcia rozwoju technik informatycznych.</p> <p>Potrafi omówić możliwości współczesnych komputerów i tendencje w ich rozwoju.</p> <p>Śledzi na bieżąco i gromadzi informacje na temat zmian w rozwoju oprogramowania i systemów operacyjnych, ich nowych dystrybucji.</p> <p>Korzysta z zasobów edukacyjnych udostępnianych w portalach przeznaczonych do kształcenia na odległość.</p>

- **stopień dostateczny** otrzymuje uczeń, który: opanował wiadomości i umiejętności określone programem nauczania w danej klasie na poziomie nieprzekraczającym wymagana zawartych w podstawach programowych, rozwiązuje (wykonuje) zadania teoretyczne i praktyczne o średnim stopniu trudności.
- Stopień dostateczny otrzymuje uczeń, który: zna i rozumie podstawowe pojęcia oraz zasadnicze zależności między nimi, sprawnie posługuje się systemem operacyjnym komputera oraz poznanymi programami narzędziowymi z zakresu informatyki.

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Rozumie pojęcie: algorytm. Formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczanymi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	Poprawnie definiuje problem i formułuje jego specyfikację. Stosuje podejście algorytmiczne do rozwiązywanego problemu. Zna i realizuje etapy rozwiązywania zadań za pomocą komputera (od sformułowania specyfikacji problemu po testowanie rozwiązania). Rozwiązuje problem zgodnie z etapami: zaplanowanie działań, wydzielenie podproblemów i określenie zależności między nimi, skonstruowanie algorytmów rozwiązujących podproblemy, analiza tych algorytmów i określenie ogólnego algorytmu rozwiązującego problem. Dokonuje analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją. Ocenia zgodność algorytmu ze specyfikacją problemu.
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Zna różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji: <ul style="list-style-type: none"> • schemat blokowy — graficzna prezentacja algorytmu, • lista kroków — słowna prezentacja algorytmu, • program w wybranym języku programowania wysokiego poziomu — praktyczna realizacja algorytmu, • arkusz kalkulacyjny. 	Zna, rozumie, wykorzystuje algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków, program w języku programowania wysokiego poziomu, arkusz kalkulacyjny. Zna, rozumie, wykorzystuje właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu. Zna, rozumie, wykorzystuje rozwiązanie zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie notacji. Zna, rozumie, wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i zapisywania algorytmów. Zna, rozumie, wykorzystuje informatyczne rozwiązanie problemu (realizację algorytmu), dobiera odpowiednie struktury danych i możliwie najlepszy algorytm oraz zapisuje go w wybranym języku programowania. Zna, rozumie, wykorzystuje metody algorytmiczne i algorytmy klasyczne do rozwiązywania problemu oraz świadomie wybiera właściwy sposób rozwiązania zadania. Zna, rozumie, wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych zadań z różnych dziedzin.
Algorytmy liniowe i algorytmy z warunkami	Zna, rozumie, wykorzystuje algorytmy liniowe i algorytmy z warunkami przy rozwiązywaniu problemów.

(rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.	Zna, rozumie, wykorzystuje algorytmy liniowe i warunkowe: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe itp.
Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.	Zna, rozumie, wykorzystuje iterację do zapisu algorytmów: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach oraz tablicach,
Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.	Zna, rozumie, wykorzystuje przykłady zastosowania rekurencji oraz powiązać sytuacje życiowe z rozwiązaniami rekurencyjnymi. Zna, rozumie, definiuje zależności rekurencyjne oraz odpowiednie procedury-funkcje rekurencyjne. Zna, rozumie, wykorzystuje metody iteracyjnej wykorzystanej w algorytmie na rekurencyjną. Zna, wyjaśnia algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi itp. Zna, rozumie, wykorzystuje rekurencję w prostych sytuacjach problemowych.
Metoda „dziel i zwyciężaj”.	Zna, podaje pojęcie: metoda „dziel i zwyciężaj”. Potrafi prosto definiować i stosować metodę „dziel i zwyciężaj” w odpowiednich sytuacjach. Zna, algorytmy z wykorzystaniem metody „dziel i zwyciężaj”.
Programowanie zachłanne.	Zna, podaje podstawowe pojęcia: programowanie zachłanne. Identyfikuje wykorzystaną metodę programowania zachłannego w przykładowych algorytmach. Zna, podaje podejście zachłanne w rozwiązywaniu problemów.
Kryptografia i kryptoanaliza. Metody szyfrowania.	Zna, podaje podstawowe metody szyfrowania.
ALGORYTMY I ICH ZASTOSOWANIE	
Algorytmy badające własności geometryczne: <ul style="list-style-type: none"> • sprawdzanie warunku trójkąta, • określanie prostokątności i równoległości prostych, • obliczanie odległości punktu od prostej, • obliczanie odległości punktów na płaszczyźnie, • badanie położenia punktu względem prostej, • badanie przynależności punktu do odcinka. 	Potrafi definiować prostą w postaci ogólnej i kierunkowej: na podstawie wzorów wyznacza odległość punktu od prostej oraz odległość punktów na płaszczyźnie; na podstawie równań kierunkowych określa prostokątność i równoległość prostych. Podaje algorytm sprawdzający warunek. Zapisuje prosty algorytm w języku programowania.
Wyznaczanie największego wspólnego dzielnika i najmniejszej wspólnej wielokrotności dwóch liczb naturalnych:	Zna, podaje największy wspólny dzielnik dwóch liczb naturalnych algorytmem Euklidesa. Stosuje iteracyjną i rekurencyjną realizację algorytmu. Zna, podaje najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający

<ul style="list-style-type: none"> • wyznaczanie największego wspólnego dzielnika — algorytm Euklidesa, • wyznaczanie najmniejszej wspólnej wielokrotności. 	<p>największy wspólny dzielnik.</p>
<p>Wyznaczanie wartości wielomianu, pozycyjne systemy liczbowe i reprezentacja danych liczbowych w komputerze:</p> <ul style="list-style-type: none"> • wyznaczanie wartości wielomianu schematem Hornera, • szybkie podnoszenie do potęgi, • pozycyjne systemy liczbowe stosowane w informatyce — system dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny) — definicja, zamiana liczb zapisanych w podanych systemach, realizacja podstawowych operacji arytmetycznych w podanych systemach, • zamiana liczb z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera, • zamiana liczb z systemu dziesiętnego na dowolny pozycyjny system liczbowy, • wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych, • reprezentacja danych liczbowych w komputerze — reprezentacja binarna liczb całkowitych i niecałkowitych, reprezentacja stałopozycyjna i zmiennopozycyjna liczb, błędy w obliczeniach. 	<p>Rozumie algorytm obliczający wartość wielomianu za pomocą schematu Hornera. Rozumie algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera. Podaje pojęcia: system liczbowy, pozycyjny system liczbowy. Podaje znaczenie reprezentacji liczb w dowolnym systemie pozycyjnym, w tym dwójkowym (binarnym), ósemkowym (oktalnym) i szesnastkowym (heksadecymalnym), oraz zamienia liczby zapisane w tych systemach liczbowych na system dziesiętny i odwrotnie. Podaje liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera. Potrafi podać podstawowe operacje arytmetyczne w dowolnych pozycyjnych systemach liczbowych. Zna, podaje reprezentacje danych liczbowych w komputerze, źródło błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.</p>
<p>Liczby pierwsze:</p> <ul style="list-style-type: none"> • sprawdzanie, czy liczba jest liczbą pierwszą, • rozkładanie liczby na czynniki pierwsze, • sito Eratostenesa. 	<p>Podaje pojęcie: liczba pierwsza. Identyfikuje algorytm sprawdzający, czy dana liczba jest pierwsza. Zna, podaje algorytm generujący liczby pierwsze — sito Eratostenesa.</p>
<p>Przeszukiwanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • przeszukiwanie liniowe tablicy jednowymiarowej, • przeszukiwanie liniowe tablicy jednowymiarowej 	<p>Zna, podaje metody liniowe przy przeszukiwaniu ciągu liczbowego, tablicy jednowymiarowej.</p>

z wartownikiem.	
<p>Znajdowanie największego lub najmniejszego elementu w ciągu liczbowym:</p> <ul style="list-style-type: none"> • znajdowanie największego elementu w ciągu liczbowym, • znajdowanie najmniejszego elementu w ciągu liczbowym. 	<p>Podaje bardzo prosty algorytm znajdowania największego elementu w ciągu liczbowym.</p> <p>Zna i identyfikuje algorytm znajdowania najmniejszego elementu w ciągu liczbowym.</p>
Znajdowanie lidera w zbiorze.	<p>Zna pojęcie: lider w zbiorze.</p> <p>Identyfikuje algorytm znajdowania lidera w zbiorze.</p>
<p>Sprawdzanie monotoniczności ciągu liczbowego:</p> <ul style="list-style-type: none"> • sprawdzanie, czy ciąg liczbowy jest rosnący lub malejący, • sprawdzanie, czy ciąg liczbowy jest nierosnący lub niemalejący, • sprawdzanie, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny. 	<p>Rozumie pojęcie: monotoniczność ciągu liczbowego.</p> <p>Identyfikuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący lub malejący.</p> <p>Podaje bardzo prosty algorytm sprawdzający, czy ciąg liczbowy jest nierosnący lub niemalejący.</p> <p>Podaje bardzo prosty algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.</p>
<p>Liniowe sortowanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • porządkowanie przez wybór, • porządkowanie przez wstawianie, • porządkowanie bąbelkowe. 	<p>Podaje proste wyjaśnienie algorytmu porządkowania przez wybór.</p> <p>Podaje proste wyjaśnienie algorytmu porządkowania przez wstawianie.</p> <p>Podaje proste wyjaśnienie algorytmu porządkowania bąbelkowego.</p>
<p>Zastosowanie metody „dziel i zwyciężaj”:</p> <ul style="list-style-type: none"> • jednoczesne znajdowanie najmniejszego i największego elementu, • przeszukiwanie binarne ciągu uporządkowanego, 	<p>Podaje proste wyjaśnienie algorytmu metody „dziel i zwyciężaj” przy realizacji algorytmów.</p> <p>Podaje proste wyjaśnienie algorytmu jednoczesnego znajdowania najmniejszego i największego elementu.</p> <p>Podaje proste wyjaśnienie algorytmu przeszukiwania binarnego uporządkowanego ciągu liczbowego.</p>
<ul style="list-style-type: none"> • sortowanie przez scalanie, • sortowanie szybkie, • znajdowanie przybliżonej wartości miejsca zerowego funkcji ciągłej — metoda połowienia przedziałów. 	<p>Podaje proste wyjaśnienie algorytmu sortowania przez scalanie.</p> <p>Podaje proste wyjaśnienie algorytmu sortowania szybkiego.</p> <p>Podaje proste wyjaśnienie algorytmu znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metodą połowienia przedziałów.</p>
<p>Obliczenia przybliżone. Realizacja wybranych algorytmów numerycznych:</p> <ul style="list-style-type: none"> • obliczanie wartości pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona), 	<p>Podaje proste wyjaśnienie pojęcia: obliczenia przybliżone, metody numeryczne.</p> <p>Podaje proste wyjaśnienie algorytmu obliczającego wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona).</p> <p>Podaje proste wyjaśnienie algorytmu obliczającego pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów.</p>

<ul style="list-style-type: none"> • obliczanie pola obszaru ograniczonego wykresem funkcji (całkowanie numeryczne), • przybliżanie wartości liczby metodą Monte Carlo. 	Podaje proste wyjaśnienie algorytmu wykonującego przybliżanie wartości liczby metodą Monte Carlo.
<p>Algorytmy na tekstach (łańcuchach):</p> <ul style="list-style-type: none"> • sprawdzanie, czy tekst jest palindromem, • porządkowanie alfabetyczne tekstu, • sprawdzanie, czy teksty są anagramami, • wyszukiwanie wzorca w tekście, • wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP. 	<p>Podaje proste wyjaśnienie algorytmu wykonywanego na tekstach, na przykład sprawdzanie, czy tekst jest palindromem, porządkowanie alfabetyczne tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście.</p> <p>Podaje proste wyjaśnienie notacji wyrażenia podanego w postaci ONP. Zapisuje i stosuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.</p>
<p>Zastosowanie programowania zachłannego:</p> <ul style="list-style-type: none"> • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana. 	Podaje proste wyjaśnienie algorytmów: algorytm zachłanny dla problemu plecakowego, algorytm wydawania reszty, algorytm Huffmana.
Wybrane algorytmy kryptograficzne.	Podaje proste wyjaśnienie algorytmów np.: kryptograficzne, w tym algorytmy symetryczne (np. szyfr Cezara, szyfr przestawieniowy, szyfr pętlowy, szyfry monoalfabetyczne i wieloalfabetyczne), algorytmy asymetryczne (np. szyfr z kluczem jawnym RSA), podpis cyfrowy itp.
WŁASNOŚCI ALGORYTMÓW	
Złożoność czasowa algorytmów. Analiza złożoności czasowej wybranych algorytmów.	<p>Rozumie pojęcia: złożoność obliczeniowa, złożoność czasowa.</p> <p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność czasową algorytmów.</p> <p>Zna, podaje złożoność czasową wybranych algorytmów (oblicza liczbę operacji dominujących wykonywanych przez algorytm).</p> <p>Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera możliwie najszybszy algorytm rozwiązujący postawiony problem.</p>
Złożoność pamięciowa algorytmów. Analiza złożoności pamięciowej wybranych algorytmów.	<p>Rozumie pojęcia: złożoność pamięciowa.</p> <p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność pamięciową algorytmów.</p> <p>Zna, podaje złożoność pamięciową wybranych algorytmów (szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu).</p> <p>Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera odpowiednie struktury danych w rozwiązaniu postawionego problemu.</p>
Tabela 1. Własności algorytmów: efektywność, poprawność, skończoność, optymalność. Analiza	Rozumie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu.

własności wybranych algorytmów.	Zna, podaje własności algorytmów na podstawie ich analizy. Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności. Bada efektywność komputerowych rozwiązań problemów. Zna, podaje poprawność komputerowego rozwiązania problemu na podstawie jego testowania.
---------------------------------	---

2. Programowanie

Tabela 2. Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
Języki programowania — pojęcie, klasyfikacja, przykłady. Prezentacja przykładowych programów w różnych językach programowania. Podstawowe zasady i metody programowania. Znajomość pojęć: translator, kompilator, interpreter, debugger, linker.	Rozumie pojęcia: język programowania, translator, kompilator, interpreter, debugger, linker. Zna klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego i wysokiego poziomu. Potrafi wymienić i sklasyfikować podstawowe języki programowania. Kompiluje i uruchamia przykładowe programy napisane w różnych językach programowania. Zna, podaje podstawowe zasady i metody programowania.
Elementy zintegrowanego systemu programowania, w tym umiejętność pisania, kompilowania i uruchamiania programów za pomocą wybranego kompilatora.	Korzysta z wybranego środowiska programistycznego (na przykład kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.
Wprowadzenie do programowania: <ul style="list-style-type: none"> • struktura programu, • operacje wejścia-wyjścia, • zmienne i ich deklaracja, • stałe i ich deklaracja, • wyrażenia arytmetyczne, relacje i operatory logiczne, • priorytety relacji i działań, • komentarze. 	Zna podstawową strukturę programu. Korzysta w programach z podstawowych operacji wejścia i wyjścia. Zna, podaje zmienne i wykorzystywać je w programach. Potrafi deklarować stałe i wykorzystywać je w programach. Zna podstawowe wyrażenia arytmetyczne, relacje i operatory logiczne oraz stosuje je w programach. Zna priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je przy pisaniu programów. Zna, podaje komentarze przy pisaniu programów.
Podstawowe konstrukcje algorytmiczne: <ul style="list-style-type: none"> • instrukcja przypisania, • instrukcja złożona, • instrukcje warunkowe, • instrukcja wyboru, • instrukcje iteracyjne. 	Zna, podaje podstawowe konstrukcje algorytmiczne, w tym instrukcję przypisania, instrukcję złożoną, instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne. Rozumie pojęcie: proste typy danych. Zna proste typy danych, w tym typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny. Wykonuje operacje na prostych typach danych. Korzysta w programach z prostych typów danych.

Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.	Potrafi w danej sytuacji problemowej wybrać właściwy typ danych dla wykorzystywanych zmiennych.
<p>Proste typy danych:</p> <ul style="list-style-type: none"> • typy całkowite, • typy rzeczywiste, • typ znakowy, • typ logiczny. <p>Realizacja operacji z wykorzystaniem prostych typów danych oraz ich zastosowanie w programach.</p>	
<p>Strukturalizacja programu:</p> <ul style="list-style-type: none"> • definiowanie procedur-funkcji, • zmienne lokalne i globalne, • parametry formalne i aktualne, • przekazywanie parametrów w procedurach-funkcjach, • przeciążanie funkcji. <p>Zastosowanie w implementacjach procedur-funkcji z parametrami. Umiejętność wyboru właściwego w danej sytuacji sposobu przekazywania parametrów.</p>	<p>Zna, podaje podstawowe zasady i cel strukturalizacji programu. Potrafi wydzielać fragmenty programu i definiować procedury-funkcje. Zna, podaje różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach. Zna, podaje różnice między parametrami formalnymi i aktualnymi. Zna, podaje sposoby przekazywania parametrów w procedurach-funkcjach i świadomie stosuje je podczas realizacji programów. Zna, podaje podstawowe sposoby przeciążania funkcji i stosuje je w programach.</p>
<p>Strukturalne typy danych:</p> <ul style="list-style-type: none"> • łańcuchy (w tym operacje na tekście), • tablice (w tym jednowymiarowe i wielowymiarowe), • rekordy-struktury (w tym tablice rekordów-struktur), • struktury dynamiczne (w tym listy, stosy, kolejki, drzewa binarne). <p>Realizacja programów z wykorzystaniem strukturalnych typów danych.</p>	<p>Zna, podaje pojęcia: strukturalne typy danych, abstrakcyjne typy danych. Potrafi definiować typ łańcuchowy. Zna, podaje predefiniowane operacje na tekście i z nich korzysta. Zapisuje programy z wykorzystaniem typu łańcuchowego. Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe. Zna, podaje proste programy z wykorzystaniem typu tablicowego. Zna, podaje proste typy rekordowo-strukturalny. Potrafi definiować tablice rekordów-struktur. Identyfikuje znaczenie dynamicznych struktur danych, w tym listy, stosy, kolejki, drzewa binarne. Zapisuje programy z wykorzystaniem dynamicznych struktur danych. Dobiera struktury danych (w tym dynamiczne struktury danych) odpowiednio do przetwarzanych informacji. Identyfikuje znaczenie typów strukturalnych, w tym typ łańcuchowy, tablicowy i rekordowy-strukturalny, oraz dynamiczne struktury danych do implementacji algorytmów.</p>
Plikowe operacje wejścia-wyjścia:	Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne.

<ul style="list-style-type: none"> • rodzaje plików, • definicja i etapy przetwarzania plików, • podstawowe operacje na plikach. <p>Realizacja programów z zastosowaniem operacji plikowych.</p>	<p>Zna i realizuje etapy przetwarzania plików. Zna podstawowe operacje na plikach i z nich korzysta. Identyfikuje znaczenie programów z wykorzystaniem typu plikowego, w tym importuje dane z plików zewnętrznych, eksportuje wyniki do plików zewnętrznych. Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach.</p>
<p>Programowanie zorientowane obiektowo:</p> <ul style="list-style-type: none"> • klasy i obiekty, • dziedziczenie i hierarchia klas, • konstruktory i destruktory, • polimorfizm, • metody wirtualne. <p>Realizacja implementacji z zastosowaniem programowania zorientowanego obiektowo.</p>	<p>Identyfikuje znaczenie pojęć: programowanie obiektowe, klasa, obiekt, enkapsulacja, dziedziczenie, hierarchia klas, polimorfizm, konstruktor, destruktor. Potrafi definiować typ obiektowy. Tworzy proste klasy i statyczne obiekty. Identyfikuje znaczenie metod konstruktora i destruktora. Identyfikuje znaczenie dynamicznych obiektów. Identyfikuje znaczenie metod wirtualnych.</p>
<p>Projekt programistyczny.</p>	<p>Zna pojęcie: inżynieria oprogramowania. Potrafi określić kolejne fazy konstruowania oprogramowania. Znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin. Działa indywidualnie lub zespołowo projekt programistyczny, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.</p>

3. Bazy danych

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
BAZODANOWE FUNKCJE ARKUSZA KALKULACYJNEGO	
Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.	<p>Zna i rozumie pojęcie: baza danych. Identyfikuje znaczenie reguł, jakim podlega tabela stanowiąca źródło danych. Potrafi zgromadzić w pliku tekstowym dane będące bazą danych. Identyfikuje działania typu: importuje dane umieszczone w pliku tekstowym do tabeli arkusza kalkulacyjnego. Pracuje w tabeli dane w pliku tekstowym, stosując odpowiednie znaki separacji.</p>
Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.	<p>Korzysta z autofiltru w celu wyselekcjonowania danych. Potrafi stosować kryteria wyboru w filtrach.</p>

Tworzenie podsumowań danych.	<p>Potrafi zastosować funkcje standardowe arkusza do podsumowań danych.</p> <p>Zna pojęcie: suma pośrednia.</p> <p>Wie, jak i do czego używa się sum pośrednich.</p> <p>Potrafi dokonać podsumowań danych zawartych w tabeli przy użyciu kreatora sum częściowych.</p> <p>Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich.</p> <p>Rozumie pojęcie: tabela przestawna.</p> <p>Wie, do czego służą tabele przestawne.</p> <p>Wskazuje, do jakich danych w celu ich podsumowania można zastosować tabele przestawne.</p> <p>Potrafi wykonać zestawienie podsumowań danych przy użyciu kreatora tabel przestawnych.</p> <p>Potrafi zmienić ustawienia w istniejącej tabeli przestawnej.</p>
Graficzna prezentacja danych na wykresach.	<p>Potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych.</p> <p>Umie tworzyć różnego typu wykresy do danych zawartych w tabelach i tabelach przestawnych.</p> <p>Potrafi ustalić odpowiednie opcje wykresu i formatować poszczególne jego elementy.</p> <p>Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauki szkolnej (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).</p>
RELACYJNA BAZA DANYCH	
Podstawy relacyjnej bazy danych.	<p>Identyfikuje typy baz danych — proste i złożone.</p> <p>Zna i rozumie pojęcia: system zarządzania bazą danych, relacyjna baza danych.</p> <p>Zna zasady projektowania bazy danych. Rozumie potrzebę unikania nadmiarowości danych.</p> <p>Rozumie, czym są pola i rekordy oraz jaką rolę pełni klucz w bazie danych.</p> <p>Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makra).</p>

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
RELACYJNA BAZA DANYCH	
Tabele w relacyjnej bazie danych i relacje między nimi.	<p>Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego.</p> <p>Potrafi zaprojektować układ tabel, unikając powtarzania danych.</p> <p>Tworzy tabele przy użyciu kreatora tabel oraz w widoku projektu.</p> <p>Potrafi ustalić właściwości pól. Zna i poprawnie stosuje typy danych w tabelach.</p> <p>Identyfikuje właściwości pól tabeli, takie jak rozmiar pola, maska wprowadzania czy reguła sprawdzania</p>

	<p>poprawności.</p> <p>Identyfikuje rolę w bazie danych pełni indeks.</p> <p>Zna pojęcia: klucz główny oraz klucz obcy.</p> <p>Poprawnie definiuje związki między tabelami.</p> <p>Rozróżnia podstawowe typy relacji między tabelami, potrafi je nazwać.</p> <p>Kontroluje i zmienia warunki wymuszania więzów integralności podczas edytowania właściwości określonej relacji.</p> <p>Identyfikuje tabele, uwzględniając różne typy danych oraz ustalone dla nich właściwości pól (rozmiar, ograniczenia, wymagania, komunikaty dla użytkownika itp.).</p>
<p>Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.</p>	<p>Zna różne formaty danych.</p> <p>Wykonuje czynności edycyjne na tabelach bazy danych.</p> <p>Wie, jak wprowadzić dane bezpośrednio do tabeli lub poprzez formularz.</p> <p>Wie, jak korzystać z formularzy do wprowadzania, wyszukiwania i edycji danych.</p> <p>Potrafi porządkować tabele i przeglądać wybrane rekordy przez sortowanie jedno- i wieloparametrowe.</p> <p>Zmienia sposób prezentowania danych poprzez ich sortowanie lub filtrowanie.</p> <p>Potrafi drukować tabele i formularze.</p> <p>Tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu.</p> <p>Identyfikuje nagłówek lub stopkę do formularza, zmienia jego szatę graficzną.</p>
<p>Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend.</p>	<p>Zna podstawowe typy kwerend.</p> <p>Identyfikuje zapytania, korzystając z widoku projektu oraz kreatorów różnego typu kwerend wybierających dane z jednej, kilku tabel lub z wcześniej zdefiniowanych kwerend.</p> <p>Konstruuje pola obliczeniowe, wprowadza korekty.</p> <p>Określa złożone kryteria liczbowe i definiuje wyrażenia.</p> <p>Stosuje odpowiednie kryteria i parametry do wyszukiwania danych.</p> <p>Wprowadza wiele kryteriów wyboru w jednym polu, stosując odpowiednie operatory.</p> <p>Tworzy kwerendy obliczające podsumowania.</p> <p>Tworzy kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane.</p> <p>Zmienia właściwości sprzężenia między tabelami, na podstawie, których tworzona jest kwerenda.</p> <p>Identyfikuje podstawy języka zapytań SQL pozwalające na tworzenie i modyfikowanie kwerend wybierających, parametrycznych, krzyżowych i modyfikujących tabele.</p>
<p>Przygotowanie zestawień wybranych danych w raportach.</p>	<p>Wie, jak korzystać z kreatora raportów do generowania dowolnych rodzajów raportów.</p> <p>Wie, kiedy korzystać z podsumowania statystycznego. Przygotowuje raport do druku.</p>

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
RELACYJNA BAZA DANYCH	
Makropolecenia.	Definiuje makropolecenia złożone z kilku sekwencji akcji. Rozumie znaczenie komunikatu dotyczącego ostrzeżenia o zabezpieczeniach związanych z wyłączeniem określonej zawartości bazy danych.
Ochrona bazy danych.	Rozumie konieczność chronienia zgromadzonych danych przed osobami niepowołanymi do korzystania z nich i ich przetwarzania. Potrafi stosować podstawowe mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją. Zna efekt działania szyfrowania (kodowania) bazy danych. Rozumie konieczność wykonywania kopii bezpieczeństwa (na przykład przy wprowadzaniu i testowaniu kwerend funkcjonalnych).

4. Multimedia i grafika komputerowa

Tabela 4. Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimedków i grafiki komputerowej

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
MULTIMEDIA I GRAFIKA KOMPUTEROWA	
Dźwięk w komputerze.	Potrafi wymienić kilka typów plików dźwiękowych oraz ich zastosowanie. Wie, jak powstał cyfrowy zapis dźwięku. Rozumie jego podstawowe parametry: częstotliwość próbkowania, poziomy kwantyzacji. Zna i charakteryzuje metody kompresji plików dźwiękowych. Zna konwersję między różnymi formatami dźwięku. Potrafi nagrać głos i specjalne efekty dźwiękowe. Potrafi przeprowadzić edycję, miksowanie i montaż dźwięku, wykorzystując materiały własne oraz bazy gotowych efektów dźwiękowych.
Grafika komputerowa.	Zna podstawowe pojęcia związane z grafiką rastrową i wektorową. Objaśnia podstawowe różnice pomiędzy grafiką rastrową a wektorową. Zna wady i zalety grafiki rastrowej i wektorowej. Zapisuje pliki graficzne w różnych formatach.

	<p>Zna kompresję stratną i bezstratną plików graficznych oraz rozumie, jaki ma ona wpływ na wielkość plików i jakość obrazu.</p> <p>Zna podstawowe algorytmy kompresji</p> <p>Zna sposoby reprezentowania obrazu w komputerze.</p> <p>Podaje znaczenie grafiki przeznaczonej na stronę WWW.</p> <p>Potrafi wyjaśnić zasadę powstawania animacji.</p> <p>Tworzy własną animację.</p> <p>Rozumie pojęcia: wymiary obrazu i rozdzielczość obrazu.</p> <p>Wykonuje czynności edycyjne obrazu.</p> <p>Opisuje podstawowe modele barw.</p> <p>Zna własności barwy.</p> <p>Wie, jak działa skaner i jak powstaje cyfrowy obraz ze skanera.</p>
Cyfrowy zapis filmu.	<p>Rozumie pojęcie: kontener multimedialny.</p> <p>Wie, jak działają kodeki wideo na przykładzie pojedynczego obrazu.</p> <p>Rozumie zmontowanie filmu i opublikowanie go.</p>

5. Systemy operacyjne i sieci komputerowe

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
SYSTEMY OPERACYJNE	
Przegląd systemów operacyjnych.	<p>Wymienia elementy budowy komputera oraz rozumie rolę jego poszczególnych podzespołów.</p> <p>Wymienia rolę BIOS-u i wie, na czym polega rozruch komputera.</p> <p>Wymienia podstawowe funkcje systemu operacyjnego:</p> <ul style="list-style-type: none"> — wykonywanie programów, — operacje wejścia-wyjścia, — zarządzanie systemem plików, — komunikacja, — przekazywanie komunikatów, — wykrywanie błędów. <p>Podaje znaczenie systemu operacyjnego.</p> <p>Rozróżnia podstawowe systemy operacyjne.</p>

	<p>Podaje pojęcia: system plików, klaster, sektor.</p> <p>Podaje znaczenie pracy maszyn wirtualnych.</p> <p>Zna pojęcia: środowisko wielodostępowe, wielozadaniowość, jądro systemu, GUI, maszyna wirtualna.</p>
Podstawowe zadania systemu operacyjnego.	<p>Wylicza i podaje podstawowe zadania systemu operacyjnego, do których należą:</p> <ul style="list-style-type: none"> — zarządzanie procesami, — zarządzanie pamięcią operacyjną, — zarządzanie plikami, — zarządzanie systemem wejścia-wyjścia, — zarządzanie pamięcią pomocniczą (dyskową), — praca w sieci, — system ochrony. <p>Podaje na czym polega i jak przebiega interpretacja poleceń (wiersz poleceń, powłoka).</p>
SIECI KOMPUTEROWE	
Budowa, działanie i eksploatacja sieci komputerowych.	<p>Rozróżnia rodzaje sieci ze względu na ich zakresy: LAN, MAN, WAN.</p> <p>Rozróżnia podstawowe rodzaje sieci ze względu na topologię.</p> <p>Zna i podaje bardzo podstawowe topologie sieciowe (topologia magistrali, gwiazdy, pierścienia, pełnych połączeń, mieszana).</p> <p>Zna i podaje bardzo podstawowe urządzenia i elementy sieci komputerowych (karty sieciowe, okablowanie sieciowe, urządzenia sieciowe).</p> <p>Zna i podaje bardzo podstawowe technologie sieciowe (metody dostępu do sieci).</p> <p>Zna i podaje bardzo podstawowe rolę urządzeń służących do rozbudowy sieci (wzmacniaki, koncentratory, mosty, przełączniki, routery, bramy).</p> <p>Zna i podaje bardzo podstawowe zasady administrowania siecią komputerową.</p> <p>Zna i podaje bardzo podstawowe usługi oraz narzędzia sieciowe i korzysta z nich.</p>
Protokoły sieciowe. Model OSI.	<p>Zna i wylicza bardzo podstawowe warstwowy model sieci komputerowych.</p> <p>Potrafi wyjaśnić, czym jest protokół i zestaw protokołów.</p> <p>Zna i wylicza bardzo podstawowe funkcje poszczególnych warstw modelu OSI.</p> <p>Potrafi określić, czy dany komputer może się komunikować z innymi komputerami w sieci.</p> <p>Zna i podaje bardzo podstawowe nazwy podstawowych protokołów sieciowych i opisać ich własności.</p>

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
SIECI KOMPUTEROWE	
Zestaw protokołów TCP/IP.	<p>Zna i wylicza podział na warstwy zestawu protokołów TCP/IP i porównuje go z modelem OSI.</p> <p>Wymienia sposób przesyłania danych między protokołami TCP/IP.</p> <p>Zna pojęcie: pakiet danych.</p> <p>Identyfikuje warstwy sieciowe, używając pojęć: protokół IP, datagram IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska podsieci, serwer DNS.</p> <p>Identyfikuje możliwość ustawień sieciowych danego komputera i jego lokalizację w sieci.</p>
Bezpieczeństwo informacji w sieciach.	<p>Wylicza podstawowe zasady administrowania siecią komputerową w architekturze klient-serwer.</p> <p>Wylicza zagrożenia związane z pracą komputera w sieci: niszczące programy i najczęstsze ataki, sniffer, podszywanie i naśladownictwo. Potrafi podjąć działania prewencyjne.</p> <p>Wylicza sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie SSL, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zaporę sieciową, filtrowanie pakietów.</p> <p>Rozumie konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów.</p> <p>Wylicza zagrożenia związane z przestępczością komputerową, w tym z piractwem komputerowym i nielegalnymi transakcjami w sieci.</p> <p>Identyfikuje kopie bezpieczeństwa danych.</p> <p>Identyfikuje podstawy kodowania danych.</p>
Tworzenie i publikowanie własnych materiałów w sieci.	<p>Zna bardzo podstawową strukturę oraz podstawowe znaczniki dokumentu HTML.</p> <p>Potrafi edytować witrynę WWW z wykorzystaniem tekstu, tabel, odpowiednich formatów grafiki i opracowań multimedialnych.</p> <p>Dostrzega korzyści związane ze stosowaniem arkuszy stylów.</p> <p>Edytuje materiały (grafiki, galerie zdjęć, wykresy, prezentacje multimedialne, animacje, dźwięk), które zamieszcza w dokumencie HTML, dodając odpowiednie menu i odsyłacze.</p> <p>Zna i stosuje obowiązujące zasady prawne dotyczące prawa autorskiego.</p> <p>Identyfikuje podstawowe polecenia języka PHP i wykorzystuje go w dokumentach HTML.</p>

6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

Tabela 6. Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	
Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	Identyfikuje przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie. Wylicza rodzaje licencji programów komputerowych. Przestrzega zasad etyki. Wylicza bariery w korzystaniu z technologii informacyjnej przez różne grupy społeczne. Zna korzyści i zagrożenia wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.
Kierunki rozwoju informatyki i jej zastosowań.	Śledzi na bieżąco nowości i zna najnowsze osiągnięcia rozwoju technik informatycznych. Potrafi omówić możliwości współczesnych komputerów i tendencje w ich rozwoju. Śledzi na bieżąco i gromadzi informacje na temat zmian w rozwoju oprogramowania i systemów operacyjnych, ich nowych dystrybucji. Korzysta z zasobów edukacyjnych udostępnianych w portalach przeznaczonych do kształcenia na odległość.

● **stopień dopuszczający** otrzymuje uczeń, który: ma braki w opanowaniu materiału programowego określonego w podstawach programowych, ale braki te nie przekraczają możliwości uzyskania przez ucznia podstawowej wiedzy z danego przedmiotu w ciągu dalszej nauki, rozwiązuje (wykonuje) zadania teoretyczne i praktyczne o niewielkim stopniu trudności.

Stopień dopuszczający otrzymuje uczeń, który: w stopniu podstawowym posługuje się systemem operacyjnym komputera, potrafi samodzielnie zastosować podstawowe elementy środowiska, potrafi samodzielnie wykonać zadania o niewielkim stopniu trudności, dysponuje możliwościami pozwalającymi uzupełnić występujące braki, brak mu systematyczności i pilności w wykonywaniu powierzonych mu zadań z zakresu informatyki.

Treści nauczania i szczegółowe umiejętności ucznia podzielono na działy i zamieszczono w tabelach.

1. Algorytmika

Tabela 1. Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Rozumie pojęcie: algorytm. Wylicza przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczonymi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	Poprawnie definiuje problem i formułuje jego specyfikację. Stosuje podejście algorytmiczne do rozwiązywanego problemu. Identyfikuje i wylicza etapy rozwiązywania zadań za pomocą komputera od sformułowania specyfikacji problemu po testowanie rozwiązania. Identyfikuje problem zgodnie z etapami: zaplanowanie działań, wydzielenie podproblemów i określenie zależności między nimi, skonstruowanie algorytmów rozwiązujących podproblemy, analiza tych algorytmów i określenie ogólnego algorytmu rozwiązującego problem. Potrafi wskazać zgodność algorytmu ze specyfikacją problemu.
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Potrafi wskazać różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji: <ul style="list-style-type: none"> • schemat blokowy — graficzna prezentacja algorytmu, • lista kroków — słowna prezentacja algorytmu, • program w wybranym języku programowania wysokiego poziomu — praktyczna realizacja 	Potrafi wskazać algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków, program w języku programowania wysokiego poziomu, arkusz kalkulacyjny. Wyodrębnia elementy składowe algorytmu. Potrafi wskazać właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu. Formułuje informatyczne rozwiązanie bardzo algorytmu, wylicza elementy struktury oraz zapisuje go w wybranym języku programowania.

algorytmu, • arkusz kalkulacyjny.	Identyfikuje metody algorytmiczne i algorytmy klasyczne. Podaje jak zdobyta wiedza i umiejętności może być w różnych dziedzinach.
Algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.	Identyfikuje pojęcia: algorytm liniowy, algorytm z warunkami (rozgałęzieniami). Wycisza algorytmy liniowe i algorytmy z warunkami przy rozwiązywaniu problemów. Identyfikuje wykorzystaną metodę programowania liniowego i z warunkami w przykładowych algorytmach. Zapisuje i stosuje bardzo proste algorytmy liniowe i warunkowe
Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.	Rozumie pojęcia: iteracja, wzór iteracyjny. Potrafi zdefiniować iterację. Identyfikuje wykorzystaną metodę iteracji w przykładowych algorytmach. Identyfikuje iterację do zapisu algorytmów: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych.
Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.	Identyfikuje pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna. Identyfikuje wykorzystaną metodę rekurencji w przykładowych algorytmach. Podaje bardzo prostą analizę wartości zmiennych w przykładowych algorytmach rekurencyjnych.
Metoda „dziel i zwyciężaj”.	Identyfikuje pojęcie: metoda „dziel i zwyciężaj”. Potrafi prosto definiować i stosować metodę „dziel i zwyciężaj” w odpowiednich sytuacjach. Identyfikuje algorytmy z wykorzystaniem metody „dziel i zwyciężaj”.
Programowanie zachłanne.	Identyfikuje podstawowe pojęcia: programowanie zachłanne. Identyfikuje wykorzystaną metodę programowania zachłannego w przykładowych algorytmach. Identyfikuje podejście zachłanne w rozwiązywaniu problemów.
Kryptografia i kryptoanaliza. Metody szyfrowania.	Wycisza podstawowe metody szyfrowania.
ALGORYTMY I ICH ZASTOSOWANIE	
Algorytmy badające własności geometryczne: • sprawdzanie warunku trójkąta, • określanie prostokątności i równoległości prostych, • obliczanie odległości punktu od prostej, • obliczanie odległości punktów na płaszczyźnie, • badanie położenia punktu względem prostej,	Potrafi definiować prostą w postaci ogólnej i kierunkowej: na podstawie wzorów wyznacza odległość punktu od prostej oraz odległość punktów na płaszczyźnie, na podstawie równań kierunkowych określa prostokątność i równoległość prostych. Podaje prosty algorytm sprawdzający warunek. Zapisuje prosty algorytm w języku programowania.

<ul style="list-style-type: none"> • badanie przynależności punktu do odcinka. 	
<p>Wyznaczanie największego wspólnego dzielnika i najmniejszej wspólnej wielokrotności dwóch liczb naturalnych:</p> <ul style="list-style-type: none"> • wyznaczanie największego wspólnego dzielnika — algorytm Euklidesa, • wyznaczanie najmniejszej wspólnej wielokrotności. 	<p>Identyfikuje największy wspólny dzielnik dwóch liczb naturalnych algorytmem Euklidesa. Stosuje iteracyjną i rekurencyjną realizację algorytmu.</p> <p>Wyznacza najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający największy wspólny dzielnik.</p>
<p>Wyznaczanie wartości wielomianu, pozycyjne systemy liczbowe i reprezentacja danych liczbowych w komputerze:</p> <ul style="list-style-type: none"> • wyznaczanie wartości wielomianu schematem Hornera, • szybkie podnoszenie do potęgi, • pozycyjne systemy liczbowe stosowane w informatyce — system dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny) — definicja, zamiana liczb zapisanych w podanych systemach, realizacja podstawowych operacji arytmetycznych w podanych systemach, zamiana liczb z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera, • zamiana liczb z systemu dziesiętnego na dowolny pozycyjny system liczbowy, • wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych, • reprezentacja danych liczbowych w komputerze — reprezentacja binarna liczb całkowitych • i niecałkowitych, reprezentacja • stałopozycyjna i zmiennopozycyjna liczb, błędy w obliczeniach. 	<p>Rozumie algorytm obliczający wartość wielomianu za pomocą schematu Hornera.</p> <p>Rozumie algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera.</p> <p>Podaje pojęcia: system liczbowy, pozycyjny system liczbowy.</p> <p>Podaje znaczenie reprezentacji liczb w dowolnym systemie pozycyjnym, w tym dwójkowym (binarnym), ósemkowym (oktalnym) i szesnastkowym (heksadecymalnym), oraz zamienia liczby zapisane w tych systemach liczbowych na system dziesiętny i odwrotnie.</p> <p>Określa liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera.</p> <p>Potrafi podać podstawowe operacje arytmetyczne w dowolnych pozycyjnych systemach liczbowych.</p> <p>Identyfikuje reprezentacje danych liczbowych w komputerze, źródło błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.</p>
<p>Liczby pierwsze:</p> <ul style="list-style-type: none"> • sprawdzanie, czy liczba jest liczbą pierwszą, • rozkładanie liczby na czynniki pierwsze, 	<p>Podaje pojęcie: liczba pierwsza.</p> <p>Identyfikuje algorytm sprawdzający, czy dana liczba jest pierwsza.</p> <p>Rozumie algorytm generujący liczby pierwsze — sito Eratostenesa.</p>

<ul style="list-style-type: none"> • sito Eratostenesa. 	
<p>Przeszukiwanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • przeszukiwanie liniowe tablicy jednowymiarowej, • przeszukiwanie liniowe tablicy jednowymiarowej z wartownikiem. 	Wylicza metody liniowe przy przeszukiwaniu ciągu liczbowego, tablicy jednowymiarowej.
<p>Znajdowanie największego lub najmniejszego elementu w ciągu liczbowym:</p> <ul style="list-style-type: none"> • znajdowanie największego elementu w ciągu liczbowym, • znajdowanie najmniejszego elementu w ciągu liczbowym. 	<p>Podaje bardzo prosty algorytm znajdowania największego elementu w ciągu liczbowym.</p> <p>Identyfikuje algorytm znajdowania najmniejszego elementu w ciągu liczbowym.</p>
Znajdowanie lidera w zbiorze.	<p>Identyfikuje pojęcie: lider w zbiorze.</p> <p>Identyfikuje algorytm znajdowania lidera w zbiorze.</p>
<p>Sprawdzanie monotoniczności ciągu liczbowego:</p> <ul style="list-style-type: none"> • sprawdzanie, czy ciąg liczbowy jest rosnący lub malejący, • sprawdzanie, czy ciąg liczbowy jest nierosnący lub niemalejący, • sprawdzanie, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny. 	<p>Rozumie pojęcie: monotoniczność ciągu liczbowego.</p> <p>Identyfikuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący lub malejący.</p> <p>Podaje bardzo prosty algorytm sprawdzający, czy ciąg liczbowy jest nierosnący lub niemalejący.</p> <p>Podaje bardzo prosty algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.</p>
<p>Liniowe sortowanie ciągu liczbowego:</p> <ul style="list-style-type: none"> • porządkowanie przez wybór, • porządkowanie przez wstawianie, • porządkowanie bąbelkowe. 	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmu porządkowania przez wybór.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu porządkowania przez wstawianie.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu porządkowania bąbelkowego.</p>
<p>Zastosowanie metody „dziel i zwyciężaj”:</p> <ul style="list-style-type: none"> • jednoczesne znajdowanie najmniejszego i największego elementu, • przeszukiwanie binarne ciągu uporządkowanego, 	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmu metody „dziel i zwyciężaj” przy realizacji algorytmów.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu jednoczesnego znajdowania najmniejszego i największego elementu.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu przeszukiwania binarnego uporządkowanego ciągu liczbowego.</p>
<ul style="list-style-type: none"> • sortowanie przez scalanie, • sortowanie szybkie, • znajdowanie przybliżonej wartości miejsca zerowego funkcji ciągłej — metoda połowienia przedziałów. 	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmu sortowania przez scalanie.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu sortowania szybkiego.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metodą połowienia przedziałów.</p>

<p>Obliczenia przybliżone. Realizacja wybranych algorytmów numerycznych:</p> <ul style="list-style-type: none"> • obliczanie wartości pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona), • obliczanie pola obszaru ograniczonego wykresem funkcji (całkowanie numeryczne), • przybliżanie wartości liczby metodą Monte Carlo. 	<p>Identyfikuje lub podaje proste wyjaśnienie pojęcia: obliczenia przybliżone, metody numeryczne.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu obliczającego wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona).</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu obliczającego pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów.</p> <p>Identyfikuje lub podaje proste wyjaśnienie algorytmu wykonującego przybliżanie wartości liczby metodą Monte Carlo.</p>
<p>Algorytmy na tekstach (łańcuchach):</p> <ul style="list-style-type: none"> • sprawdzanie, czy tekst jest palindromem, • porządkowanie alfabetyczne tekstu, • sprawdzanie, czy teksty są anagramami, • wyszukiwanie wzorca w tekście, • wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP. 	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmu wykonywanego na tekstach, na przykład sprawdzanie, czy tekst jest palindromem, porządkowanie alfabetyczne tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście.</p> <p>Identyfikuje lub podaje proste wyjaśnienie notacji wyrażenia podanego w postaci ONP. Zapisuje i stosuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.</p>
<p>Zastosowanie programowania zachłannego:</p> <ul style="list-style-type: none"> • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana. 	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmów: algorytm zachłanny dla problemu plecakowego, algorytm wydawania reszty, algorytm Huffmana.</p>
<p>Wybrane algorytmy kryptograficzne.</p>	<p>Identyfikuje lub podaje proste wyjaśnienie algorytmów np.: kryptograficzne, w tym algorytmy symetryczne (np. szyfr Cezara, szyfr przestawieniowy, szyfr płotowy, szyfry monoalfabetyczne i wieloalfabetyczne), algorytmy asymetryczne (np. szyfr z kluczem jawnym RSA), podpis cyfrowy itp.</p>
<p>WŁASNOŚCI ALGORYTMÓW</p>	
<p>Złożoność czasowa algorytmów. Analiza złożoności czasowej wybranych algorytmów.</p>	<p>Rozumie pojęcia: złożoność obliczeniowa, złożoność czasowa.</p> <p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność czasową algorytmów.</p> <p>Identyfikuje lub podaje złożoność czasową wybranych algorytmów (oblicza liczbę operacji dominujących wykonywanych przez algorytm).</p> <p>Identyfikuje lub podaje złożoność czasową algorytmów rozwiązujących ten sam problem.</p> <p>Dobiera możliwie najszybszy algorytm rozwiązujący postawiony problem.</p>
<p>Złożoność pamięciowa algorytmów. Analiza złożoności pamięciowej wybranych algorytmów.</p>	<p>Rozumie pojęcie: złożoność pamięciowa.</p> <p>Potrafi wyznaczać, oceniać oraz uzasadniać złożoność pamięciową algorytmów.</p> <p>Identyfikuje lub podaje złożoność pamięciową wybranych algorytmów (szacuje wielkość pamięci potrzebnej do</p>

	komputerowej realizacji algorytmu). Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem. Dobiera odpowiednie struktury danych w rozwiązaniu postawionego problemu.
Tabela 1. Własności algorytmów: efektywność, poprawność, skończoność, optymalność. Analiza własności wybranych algorytmów.	Rozumie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu. Identyfikuje lub podaje własności algorytmów na podstawie ich analizy. Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności. Identyfikuje lub podaje efektywność komputerowych rozwiązań problemów. Identyfikuje lub podaje poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

2. Programowanie

Tabela 2. Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
Języki programowania — pojęcie, klasyfikacja, przykłady. Prezentacja przykładowych programów w różnych językach programowania. Podstawowe zasady i metody programowania. Znajomość pojęć: translator, kompilator, interpreter, debugger, linker.	Rozumie pojęcia: język programowania, translator, kompilator, interpreter, debugger, linker. Identyfikuje lub podaje klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego i wysokiego poziomu. Potrafi wymienić i sklasyfikować podstawowe języki programowania. Kompiluje i uruchamia przykładowe programy napisane w różnych językach programowania. Zna i rozumie podstawowe zasady i metody programowania.
Elementy zintegrowanego systemu programowania, w tym umiejętność pisania, kompilowania i uruchamiania programów za pomocą wybranego kompilatora.	Korzysta z wybranego środowiska programistycznego (na przykład kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.
Wprowadzenie do programowania: <ul style="list-style-type: none"> • struktura programu, • operacje wejścia-wyjścia, • zmienne i ich deklaracja, • stałe i ich deklaracja, • wyrażenia arytmetyczne, relacje i operatory logiczne, • priorytety relacji i działań, • komentarze. 	Zna podstawową strukturę programu. Korzysta w programach z podstawowych operacji wejścia i wyjścia. Identyfikuje lub podaje zmienne i wykorzystywać je w programach. Potrafi deklarować stałe i wykorzystywać je w programach. Identyfikuje lub podaje podstawowe wyrażenia arytmetyczne, relacje i operatory logiczne oraz stosuje je w programach. Identyfikuje lub podaje priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je przy pisaniu programów. Identyfikuje lub podaje komentarze przy pisaniu programów.

<p>Podstawowe konstrukcje algorytmiczne:</p> <ul style="list-style-type: none"> • instrukcja przypisania, • instrukcja złożona, • instrukcje warunkowe, • instrukcja wyboru, • instrukcje iteracyjne. <p>Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.</p>	<p>Identyfikuje lub podaje podstawowe konstrukcje algorytmiczne, w tym instrukcję przypisania, instrukcję złożoną, instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne.</p> <p>Identyfikuje lub podaje konstrukcje algorytmiczne.</p>
<p>Proste typy danych:</p> <ul style="list-style-type: none"> • typy całkowite, • typy rzeczywiste, • typ znakowy, • typ logiczny. <p>Realizacja operacji z wykorzystaniem prostych typów danych oraz ich zastosowanie w programach.</p>	<p>Rozumie pojęcie: proste typy danych.</p> <p>Zna proste typy danych, w tym typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny.</p> <p>Wykonuje operacje na prostych typach danych.</p> <p>Korzysta w programach z prostych typów danych.</p> <p>Potrafi w danej sytuacji problemowej wybrać właściwy typ danych dla wykorzystywanych zmiennych.</p>
<p>Strukturalizacja programu:</p> <ul style="list-style-type: none"> • definiowanie procedur-funkcji, • zmienne lokalne i globalne, • parametry formalne i aktualne, • przekazywanie parametrów w procedurach-funkcjach, • przeciążanie funkcji. <p>Zastosowanie w implementacjach procedur-funkcji z parametrami. Umiejętność wyboru właściwego w danej sytuacji sposobu przekazywania parametrów.</p>	<p>Rozumie podstawowe zasady i cel strukturalizacji programu.</p> <p>Potrafi wydzielać fragmenty programu i definiować procedury-funkcje.</p> <p>Rozumie różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach.</p> <p>Rozumie różnice między parametrami formalnymi i aktualnymi.</p> <p>Identyfikuje lub podaje sposoby przekazywania parametrów w procedurach-funkcjach i świadomie stosuje je podczas realizacji programów.</p> <p>Zna podstawowe sposoby przeciążania funkcji i stosuje je w programach.</p>
<p>Strukturalne typy danych:</p> <ul style="list-style-type: none"> • łańcuchy (w tym operacje na tekście), • tablice (w tym jednowymiarowe i wielowymiarowe), • rekordy-struktury (w tym tablice rekordów-struktur), • struktury dynamiczne (w tym listy, stosy, kolejki, drzewa binarne). <p>Realizacja programów z wykorzystaniem strukturalnych typów danych.</p>	<p>Rozumie pojęcia: strukturalne typy danych, abstrakcyjne typy danych.</p> <p>Potrafi definiować typ łańcuchowy.</p> <p>Identyfikuje lub podaje predefiniowane operacje na tekście i z nich korzysta.</p> <p>Identyfikuje lub podaje programy z wykorzystaniem typu łańcuchowego.</p> <p>Identyfikuje lub podaje definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe.</p> <p>Identyfikuje lub podaje programy z wykorzystaniem typu tablicowego.</p> <p>Identyfikuje lub podaje typ rekordowy-strukturalny.</p> <p>Identyfikuje lub podaje tablice rekordów-struktur.</p>

	<p>Identyfikuje lub podaje programy z wykorzystaniem typu rekordowego-strukturalnego, w tym tablic rekordów-struktur.</p> <p>Identyfikuje lub podaje dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne.</p> <p>Identyfikuje lub podaje programy z wykorzystaniem dynamicznych struktur danych.</p> <p>Dobiera struktury danych (w tym dynamiczne struktury danych) odpowiednio do przetwarzanych informacji.</p> <p>Identyfikuje lub podaje typy strukturalne, w tym typ łańcuchowy, tablicowy i rekordowy-strukturalny, oraz dynamiczne struktury danych do implementacji algorytmów.</p>
<p>Plikowe operacje wejścia-wyjścia:</p> <ul style="list-style-type: none"> • rodzaje plików, • definicja i etapy przetwarzania plików, • podstawowe operacje na plikach. <p>Realizacja programów z zastosowaniem operacji plikowych.</p>	<p>Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne.</p> <p>Identyfikuje lub podaje etapy przetwarzania plików.</p> <p>Identyfikuje lub podaje podstawowe operacje na plikach i z nich korzysta.</p> <p>Identyfikuje lub podaje programy z wykorzystaniem typu plikowego, w tym importuje dane z plików zewnętrznych, eksportuje wyniki do plików zewnętrznych.</p> <p>Identyfikuje lub podaje prostą bazę danych z zastosowaniem operacji na plikach.</p>
<p>Programowanie zorientowane obiektowo:</p> <ul style="list-style-type: none"> • klasy i obiekty, • dziedziczenie i hierarchia klas, • konstruktory i destruktory, • polimorfizm, • metody wirtualne. <p>Realizacja implementacji z zastosowaniem programowania zorientowanego obiektowo.</p>	<p>Rozumie pojęcia: programowanie obiektowe, klasa, obiekt, enkapsulacja, dziedziczenie, hierarchia klas, polimorfizm, konstruktor, destruktor.</p> <p>Identyfikuje lub podaje typ obiektowy.</p> <p>Identyfikuje lub podaje proste klasy i statyczne obiekty.</p> <p>Identyfikuje lub podaje metody konstruktora i destruktora.</p> <p>Projektuje i realizuje hierarchię klas.</p> <p>Identyfikuje lub podaje metody wirtualne.</p> <p>Realizuje programy z wykorzystaniem typu obiektowego.</p>
<p>Projekt programistyczny.</p>	<p>Zna pojęcie: inżynieria oprogramowania.</p> <p>Identyfikuje lub podaje kolejne fazy konstruowania oprogramowania.</p> <p>Identyfikuje lub podaje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin.</p> <p>Uczestniczy indywidualnie lub zespołowo w projekcie programistycznym, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.</p>

3. Bazy danych

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
BAZODANOWE FUNKCJE ARKUSZA KALKULACYJNEGO	

Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.	Zna i rozumie pojęcie: baza danych. Identyfikuje lub podaje reguły, jakim podlega tabela stanowiąca źródło danych. Potrafi zgromadzić w pliku tekstowym dane będące bazą danych. Identyfikuje lub zna Importowanie danych umieszczonych w pliku tekstowym do tabeli arkusza kalkulacyjnego. Zna jak zgromadzić w tabeli dane w pliku tekstowym.
Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.	Korzysta z autofiltru w celu wyselekcjonowania danych. Identyfikuje lub podaje złożone kryteria wyboru w filtrach zaawansowanych. Identyfikuje lub podaje rodzaj filtru w celu wybrania informacji potrzebnych w danej sytuacji.
Tworzenie podsumowań danych.	Identyfikuje lub podaje funkcje standardowe arkusza do podsumowań danych. Zna pojęcie: suma pośrednia. Wie, jak i do czego używa się sum pośrednich. Potrafi dokonać podsumowań danych zawartych w tabeli przy użyciu kreatora sum częściowych. Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich. Zna pojęcie: tabela przestawna. Wie, do czego służą tabele przestawne. Potrafi edytować ustawienia w istniejącej tabeli przestawnej.
Graficzna prezentacja danych na wykresach.	Potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych. Identyfikuje lub tworzy różnego typu wykresy do danych zawartych w tabelach i tabelach przestawnych. Potrafi ustalić odpowiednie opcje wykresu i formatować poszczególne jego elementy. Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauki szkolnej (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).
RELACYJNA BAZA DANYCH	
Podstawy relacyjnej bazy danych.	Zna różne typy baz danych — proste i złożone. Zna i rozumie pojęcia: system zarządzania bazą danych, relacyjna baza danych. Wylicza zasady projektowania bazy danych. Identyfikuje lub podaje potrzebę unikania nadmiarowości danych. Identyfikuje lub podaje pola i rekordy oraz jaką rolę pełni klucz w bazie danych. Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makra).

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
------------------	-------------------

RELACYJNA BAZA DANYCH

Tabele w relacyjnej bazie danych i relacje między nimi.

Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego.
Potrafi wybrać układ tabel, unikając powtarzania danych.
Edytuje tabele przy użyciu kreatora tabel oraz w widoku projektu.
Potrafi ustalić właściwości pól. Zna i poprawnie stosuje typy danych w tabelach.
Identyfikuje lub podaje podstawowe właściwości pól tabeli, takie jak rozmiar pola, maska wprowadzania czy reguła sprawdzania poprawności.
Rozumie, jaką rolę w bazie danych pełni indeks.
Zna pojęcia: klucz główny oraz klucz obcy.
Identyfikuje lub podaje związki między tabelami.
Rozróżnia podstawowe typy relacji między tabelami, potrafi je nazwać.
Edytuje warunki wymuszania więzów integralności podczas edytowania właściwości określonej relacji.
Uczestniczy w zespole przy relacyjnej bazy danych.

Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.

Zna różne formaty danych.
Wykonuje czynności edycyjne na tabelach bazy danych.
Wie, jak wprowadzić dane bezpośrednio do tabeli lub poprzez formularz.
Wie, jak korzystać z formularzy do wprowadzania, wyszukiwania i edycji danych.
Edytuje tabele, przegląda wybrane rekordy przez sortowanie jedno- i wieloparametrowe.
Zmienia sposób prezentowania danych poprzez ich sortowanie lub filtrowanie.
Potrafi drukować tabele i formularze.
Tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu.
Identyfikuje lub podaje etykiety, formanty, nagłówek lub stopkę do formularza, zmienia jego szatę graficzną.

Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend.

Zna podstawowe typy kwerend.
Identyfikuje lub podaje zapytania, korzystając z widoku projektu oraz kreatorów różnego typu kwerend wybierających dane z jednej, kilku tabel lub z wcześniej zdefiniowanych kwerend.
Identyfikuje lub podaje pola obliczeniowe, wprowadza korekty.
Identyfikuje lub podaje kryteria liczbowe i definiuje wyrażenia.
Identyfikuje lub podaje kryteria i parametry do wyszukiwania danych.
Wprowadza wiele kryteriów wyboru w jednym polu.
Identyfikuje lub podaje kwerendy obliczające podsumowania.
Identyfikuje lub podaje zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn.
Tworzy proste kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane.

	<p>Identyfikuje lub podaje między tabelami, na podstawie, których tworzona jest kwerenda.</p> <p>Identyfikuje lub podaje kod zapisany w języku zapytań SQL.</p> <p>Identyfikuje lub podaje podstawy języka zapytań SQL pozwalające na tworzenie i modyfikowanie kwerend wybierających, parametrycznych, krzyżowych i modyfikujących tabele.</p>
Przygotowanie zestawień wybranych danych w raportach.	<p>Identyfikuje lub podaje proste raporty przy użyciu narzędzia Autoraport.</p> <p>Wie, jak korzystać z kreatora raportów do generowania dowolnych rodzajów raportów.</p> <p>Identyfikuje lub podaje, kiedy korzystać z podsumowania statystycznego.</p> <p>Przygotowuje raport do druku.</p>

Tabela 3. Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
RELACYJNA BAZA DANYCH	
Makropolecenia.	<p>Identyfikuje lub podaje makropolecenia złożone z kilku sekwencji akcji.</p> <p>Rozumie znaczenie komunikatu dotyczącego ostrzeżenia o zabezpieczeniach związanych z wyłączeniem określonej zawartości bazy danych.</p>
Ochrona bazy danych.	<p>Identyfikuje lub podaje konieczność chronienia zgromadzonych danych przed osobami niepowołanymi do korzystania z nich i ich przetwarzania.</p> <p>Identyfikuje lub podaje podstawowe mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją.</p> <p>Identyfikuje lub podaje działania szyfrowania (kodowania) bazy danych.</p> <p>Identyfikuje lub podaje konieczność wykonywania kopii bezpieczeństwa (na przykład przy wprowadzaniu i testowaniu kwerend funkcjonalnych).</p>

4. Multimedia i grafika komputerowa

Tabela 4. Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimediiów i grafiki komputerowej

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
MULTIMEDIA I GRAFIKA KOMPUTEROWA	
Dźwięk w komputerze.	<p>Identyfikuje lub podaje typy plików dźwiękowych.</p> <p>Identyfikuje lub podaje, jak powstał cyfrowy zapis dźwięku - rozumie jego podstawowe parametry: częstotliwość próbkowania, poziomy kwantyzacji.</p> <p>Identyfikuje lub podaje metody kompresji plików dźwiękowych.</p>

	<p>Identyfikuje lub podaje konwersję między różnymi formatami dźwięku.</p> <p>Potrafi nagrać głos i specjalne efekty dźwiękowe.</p> <p>Potrafi zastosować filtry korygujące i wzbogacające dźwięk.</p> <p>Potrafi przeprowadzić edycję, miksowanie i montaż dźwięku, wykorzystując materiały własne oraz bazy gotowych efektów dźwiękowych.</p>
Grafika komputerowa.	<p>Zna podstawowe pojęcia związane z grafiką rastrową i wektorową.</p> <p>Objaśnia podstawowe różnice pomiędzy grafiką rastrową a wektorową.</p> <p>Wylicza wady i zalety grafiki rastrowej i wektorowej.</p> <p>Zapisuje pliki graficzne w różnych formatach.</p> <p>Identyfikuje lub podaje kompresję stratną i bezstratną plików graficznych oraz rozumie, jaki ma ona wpływ na wielkość plików i jakość obrazu.</p> <p>Zna podstawowe algorytmy kompresji: RLE, LZW i kodowanie Huffmana.</p> <p>Identyfikuje lub podaje sposoby reprezentowania obrazu w komputerze.</p> <p>Edytuje grafikę przeznaczoną na stronę WWW.</p> <p>Identyfikuje lub podaje zasady tworzenia obrazu 3D.</p> <p>Potrafi wyjaśnić zasadę powstawania animacji.</p> <p>Wie jak stworzyć prostą własną animację.</p> <p>Rozumie pojęcia: wymiary obrazu i rozdzielczość obrazu.</p> <p>Wykonuje czynności edycyjne, uwzględniając warstwy, transformacje, filtry, zmianę rozmiaru obrazu, kadrowanie obrazu.</p> <p>Opisuje podstawowe modele barw i ich zastosowanie.</p> <p>Wylicza własności barwy.</p> <p>Wie, jak działa skaner i jak powstaje cyfrowy obraz ze skanera.</p>
Cyfrowy zapis filmu.	<p>Rozumie pojęcie: kontener multimedialny.</p> <p>Wylicza kodeki wideo na przykładzie pojedynczego obrazu.</p> <p>Rozumie jak zmontować film i opublikować go.</p>

5. Systemy operacyjne i sieci komputerowe

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym

TRZĘCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
SYSTEMY OPERACYJNE	
Przegląd systemów operacyjnych.	Wylicza elementy komputera i podzespołów.

	<p>Podaje podstawy znaczenia BIOS-u.</p> <p>Wymienia podstawowe funkcje systemu operacyjnego:</p> <ul style="list-style-type: none"> — wykonywanie programów, — operacje wejścia-wyjścia, — zarządzanie systemem plików, — komunikacja, — przekazywanie komunikatów, — wykrywanie błędów. <p>Identyfikuje budowę systemu operacyjnego.</p> <p>Rozróżnia podstawowe systemy operacyjne.</p> <p>Identyfikuje pojęcia: system plików, klaster, sektor.</p> <p>Identyfikuje i podaje podział systemów operacyjnych.</p> <p>Identyfikuje pojęcie: maszyna wirtualna z systemem operacyjnym DR-DOS, Linux i wykonuje podstawowe operacje na plikach.</p> <p>Identyfikuje pojęcia: środowisko wielodostępowe, wielozadaniowość, jądro systemu, GUI, maszyna wirtualna.</p>
<p>Podstawowe zadania systemu operacyjnego.</p>	<p>Wylicza podstawowe zadania systemu operacyjnego, do których należą:</p> <ul style="list-style-type: none"> — zarządzanie procesami, — zarządzanie pamięcią operacyjną, — zarządzanie plikami, — zarządzanie systemem wejścia-wyjścia, — zarządzanie pamięcią pomocniczą (dyskową), — praca w sieci, — system ochrony. <p>Identyfikuje znaczenie pojęć: wiersz poleceń, powłoka.</p>
<p>SIECI KOMPUTEROWE</p>	
<p>Budowa, działanie i eksploatacja sieci komputerowych.</p>	<p>Rozróżnia rodzaje sieci ze względu na ich zakresy: LAN, MAN, WAN.</p> <p>Rozróżnia podstawowe rodzaje sieci ze względu na topologię. Potrafi opisać topologie sieciowe (topologia magistrali, gwiazdy, pierścienia, pełnych połączeń, mieszana).</p> <p>Identyfikuje i podaje urządzenia i elementy sieci komputerowych (karty sieciowe, okablowanie sieciowe, urządzenia sieciowe).</p> <p>Identyfikuje i podaje technologie sieciowe (metody dostępu do sieci).</p> <p>Identyfikuje i podaje rolę urządzeń służących do rozbudowy sieci (wzmacniaki, koncentratory, mosty, przełączniki, routery, bramy).</p>

	<p>Identyfikuje i podaje zasady administrowania siecią komputerową.</p> <p>Identyfikuje i podaje usługi oraz narzędzia sieciowe i korzysta z nich.</p>
Protokoły sieciowe. Model OSI.	<p>Identyfikuje i podaje warstwowy model sieci komputerowych.</p> <p>Potrafi wyjaśnić, czym jest protokół i zestaw protokołów.</p> <p>Identyfikuje i podaje funkcje poszczególnych warstw modelu OSI.</p> <p>Potrafi określić, czy dany komputer może się komunikować z innymi komputerami w sieci.</p> <p>Potrafi wymienić nazwy podstawowych protokołów sieciowych i opisać ich własności.</p>

Tabela 5. Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym (cd.)

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNI
SIECI KOMPUTEROWE	
Zestaw protokołów TCP/IP.	<p>Identyfikuje i podaje podział na warstwy zestawu protokołów TCP/IP i porównuje go z modelem OSI.</p> <p>Identyfikuje i podaje sposób przesyłania danych między protokołami TCP/IP.</p> <p>Identyfikuje i podaje pojęcie: pakiet danych.</p> <p>Identyfikuje i podaje warstwę sieciową, używając pojęć: protokół IP, datagram IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska podsieci, serwer DNS.</p> <p>Potrafi określić ustawienia sieciowe danego komputera i jego lokalizację w sieci.</p>
Bezpieczeństwo informacji w sieciach.	<p>Zna podstawowe zasady administrowania siecią komputerową w architekturze klient-serwer.</p> <p>Identyfikuje i podaje zagrożenia związane z pracą komputera w sieci: niszczące programy i najczęstsze ataki, sniffer, podszywanie i naśladownictwo.</p> <p>Potrafi podjąć działania prewencyjne.</p> <p>Identyfikuje i podaje sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie SSL, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zaporę sieciową, filtrowanie pakietów.</p> <p>Rozumie konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów.</p> <p>Wylicza zagrożenia związane z przestępczością komputerową, w tym z piractwem komputerowym i nielegalnymi transakcjami w sieci.</p> <p>Potrafi wykonać kopie bezpieczeństwa danych.</p> <p>Zna podstawy kodowania danych.</p>
Tworzenie i publikowanie własnych materiałów w sieci.	Identyfikuje i podaje strukturę oraz podstawowe znaczniki dokumentu HTML.

	<p>Edytuje witrynę WWW z wykorzystaniem tekstu, tabel, odpowiednich formatów grafiki i opracowań multimedialnych.</p> <p>Dostrzega korzyści związane ze stosowaniem arkuszy stylów.</p> <p>Edytuje materiały (grafiki, galerie zdjęć, wykresy, prezentacje multimedialne, animacje, dźwięk), które zamieszcza w dokumencie HTML, dodając odpowiednie menu i odsyłacze.</p> <p>Identyfikuje i podaje obowiązujące zasady prawne dotyczące prawa autorskiego.</p> <p>Identyfikuje i podaje podstawy języka PHP i wykorzystuje go w dokumentach HTML.</p> <p>Rozumie współpracę przy projektach witryn internetowych.</p>
--	--

6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

Tabela 6. Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIĄ
ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	
Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	<p>Identyfikuje przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie.</p> <p>Wylicza rodzaje licencji programów komputerowych.</p> <p>Przestrzega zasad etyki.</p> <p>Wylicza bariery w korzystaniu z technologii informacyjnej przez różne grupy społeczne.</p> <p>Identyfikuje i podaje korzyści i zagrożenia wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.</p>
Kierunki rozwoju informatyki i jej zastosowań.	<p>Śledzi na bieżąco nowości i zna najnowsze osiągnięcia rozwoju technik informatycznych.</p> <p>Potrafi omówić możliwości współczesnych komputerów i tendencje w ich rozwoju.</p> <p>Śledzi na bieżąco i gromadzi informacje na temat zmian w rozwoju oprogramowania i systemów operacyjnych, ich nowych dystrybucji.</p> <p>Identyfikuje i podaje zasoby edukacyjne udostępniane w portalach przeznaczone do kształcenia na odległość.</p>