

PRZEDMIOTOWY SYSTEM OCENIANIA

Przedmiot JĘZYKI I TECHNIKI PROGRAMOWANIA

II

§ 1

Ocenianiu podlegają osiągnięcia edukacyjne ucznia. Ocenianie osiągnięć edukacyjnych ucznia polega na rozpoznawaniu przez nauczycieli poziomu i postępów w opanowaniu przez ucznia wiadomości i umiejętności w stosunku do wymagań edukacyjnych wynikających z podstawy programowej i realizowanych w szkole programów nauczania.

§ 2

1. Ocenianie osiągnięć edukacyjnych ma na celu:

- informowanie ucznia o poziomie jego osiągnięć edukacyjnych oraz postępach w tym zakresie;
- pomoc uczniowi w samodzielnym planowaniu swojego rozwoju;
- motywowanie ucznia do dalszej pracy;
- dostarczenie rodzicom (prawnym opiekunom) i nauczycielom informacji o postępach, trudnościach i specjalnych uzdolnieniach ucznia;
- umożliwienie nauczycielom doskonalenia organizacji i metod pracy dydaktyczno – wychowawczej.

2. Ocenianie przedmiotowe obejmuje:

- formułowanie przez nauczycieli wymagań edukacyjnych niezbędnych do uzyskania poszczególnych śródrocznych i rocznych (semestralnych) ocen klasyfikacyjnych z Informatyki oraz informowanie o nich uczniów;
- bieżące ocenianie (minimum 3 oceny na semestr) i śródroczne/roczne klasyfikowanie, według skali

i w formach przyjętych w szkole;

- przeprowadzanie egzaminów klasyfikacyjnych;
- ustalanie rocznych (semestralnych) ocen klasyfikacyjnych z informatyki oraz warunków ich poprawiania.

§ 3

1. Nauczyciel Języków i technik programowania na pierwszych lekcjach każdego roku szkolnego, informuje uczniów o wymaganiach edukacyjnych z tego przedmiotu oraz o sposobach sprawdzania osiągnięć edukacyjnych.

2. Szczegółowe wymagania edukacyjne są do wglądu u nauczyciel Języki i techniki programowania oraz stanowią załączniki do PSO z Języków i technik programowania.

§ 4

1. Nauczyciel na początku każdego roku szkolnego informuje uczniów i ich rodziców (prawnych opiekunów)

o wymaganiach edukacyjnych wynikających z realizowanego przez niego programu nauczania oraz o sposobach sprawdzania osiągnięć edukacyjnych uczniów.

2. Oceny są jawne zarówno dla ucznia jak i jego rodziców (prawnych opiekunów). Sprawdzone i ocenione prace

kontrolne uczeń i jego rodzice (prawni opiekunowie) mogą otrzymać do wglądu

3. Na prośbę ucznia lub jego rodziców (prawnych opiekunów) nauczyciel jest zobowiązany do uzasadnienia

wystawionej oceny.

4. Przed końcoworocznym (semestralnym) klasyfikacyjnym posiedzeniem rady pedagogicznej nauczyciel informuje rodziców (prawnych opiekunów) o przewidywanej ocenie klasyfikacyjnej.

§ 5

1. Klasyfikowanie śródroczne (raz w ciągu roku szkolnego) polega na okresowym podsumowaniu osiągnięć

edukacyjnych ucznia z informatyki i ustaleniu ocen klasyfikacyjnych według skali oceniania.

2. Klasyfikowanie końcoworoczne polega na podsumowaniu osiągnięć edukacyjnych ucznia w danym roku szkolnym z informatyki i ustaleniu ocen klasyfikacyjnych.

§ 6

1. Oceny klasyfikacyjne ustalają nauczyciel Języków i technik programowania.
2. Ustalona przez nauczyciela niedostateczna ocena klasyfikacyjna końcoworoczne (semestralna) może być zmieniona tylko w wyniku egzaminu poprawkowego.
3. Oceny bieżące i śródroczną ustala się w stopniach:
 - stopień celujący – 6;
 - stopień bardzo dobry – 5;
 - stopień dobry – 4;
 - stopień dostateczny – 3;
 - stopień dopuszczający – 2;
 - stopień niedostateczny – 1.

4. Skala oceniania:

Ocena Progi procentowe: wymagania podstawowe

Kartkówki:	Sprawdzianu:
100 - 95 bardzo dobry	100-96 celujący
94 – 75 dobry	95 – 91 bardzo dobry
74 – 55 dostateczny	90 – 75 dobry
54 – 41 dopuszczający	74 – 55 dostateczny
40 - 0 niedostateczny	54 – 41 dopuszczający

§ 7

1. Każdy sprawdzian pisemny jest zapowiedziany przynajmniej na tydzień przed terminem sprawdzania i odnotowany w dzienniku.
2. Za brak pracy na lekcji uczeń może otrzymać ocenę niedostateczną (trzy minusy) lub wezwanie do odpowiedzi.
3. Ocenę niedostateczną - ze sprawdzianu - można poprawić w ciągu dwóch tygodni lub w terminie ustalonym przez nauczyciela, na konsultacjach lub na lekcji; Dotyczy to również osób, które z różnych przyczyn nie pisały sprawdzianu.
4. Ocenione kartkówki i sprawdziany będą pokazywane uczniom, a po omówieniu przez nauczyciela i zapoznaniu z ocenami uczniów zostają w posiadaniu nauczyciela, który udostępnia je do wglądu rodzicom podczas konsultacji i zebrań.
5. Nieobecność na lekcji nie zwalnia ucznia z przygotowania do lekcji następnej i możliwości odpowiedzi lub pisania klasówki (chyba, że jest to dłuższa nieobecność - powyżej 3 ostatnich godzin lekcji Języki i techniki programowania lub późny powrót z zawodów do domu – to zwalnia z klasówki).
6. W przypadku pojedynczej nieobecności nieusprawiedliwionej na zapowiedzianej pracy pisemnej
czy też w przypadku usprawiedliwionej nieobecności na zapowiedzianej pracy pisemnej uczeń będzie musiał ją napisać na jednej z najbliższych lekcji bez wcześniejszej zapowiedzi ze strony nauczyciela.
7. Uczeń przyłapany na ściąganiu (lub jej próbie) otrzymuje ocenę niedostateczną, która nie podlega poprawie
8. Uczeń ma prawo być nieprzygotowany do lekcji : 1 raz w semestrze (nieprzygotowanie uczeń zgłasza na początku lekcji podczas sprawdzania obecności.)
9. Uczeń, który opuści więcej niż 50% lekcji w semestrze- jest nieklasyfikowany i zalicza u nauczyciela materiał z całego semestru w postaci egzaminu pisemnego i ustnego. Nie zaliczenie równe jest ocenie niedostatecznej.

§ 8

1. Badanie kompetencji ucznia z Języków i technik programowania odbywa się przy komputerze, dlatego każdy sprawdzian lub test planowany jest jako praktyczny – uczeń wykonuje zadania ko-

rzyszając z komputera i odpowiedniego oprogramowania; forma zadań nie odbiega od ćwiczeń, które uczniowie wykonują na zajęciach.

2. Sprawdzanie umiejętności dotyczy:

- zakresu wiadomości i umiejętności;
- rozumienia materiału naukowego;
- umiejętności stosowania wiedzy;
- kreatywności (twórcze myślenie);
- posługiwanie się językiem informatycznym;
- postawy (pracowitość, systematyczność, umiejętność pracy w grupie, aktywność);
- kultury przekazywania wiadomości.

3. Obszary podlegające ocenie:

- niezbędne części teoretyczne;
- rozwiązywanie problemów za pomocą komputera;
- łączenie umiejętności praktycznych z wiedzą teoretyczną oraz znajomości podstawowych metod pracy na

komputerze;

- znajomości wspólnych dla różnych programów mechanizmów i podstawowych pojęć i metod technologii

informacyjnej oraz zarządzania informacją;

- aktywności na lekcjach;
- praca twórcza wkraczająca poza zakres programowy (praca własna);
- udział w konkursach.

4. Oceny są jednocześnie informacją dla rodziców, wychowawcy klasy, dyrektora szkoły i nadzoru pedagogicznego o:

- efektywności procesu nauczania i uczenia się;
- wkładzie uczniów w pracę nad własnym rozwojem;
- postępach uczniów.

5. Ustala się minimum dwie prace pisemne w semestrze przeprowadzane zgodnie z harmonogramem, ustalającym nie więcej niż trzy prace pisemne w ciągu tygodnia dla danej klasy. Wszelkie zmiany uzgadniane będą między nauczycielem i uczniami, pod warunkiem, że jednego dnia nie będzie dwóch prac pisemnych, a w tygodniu trzech.

6. Tematy prac pisemnych powinny być sformułowane komunikatywnie.

7. Kartkówki – krótkie prace pisemne, obejmujące materiał z trzech ostatnich lekcji nie muszą być zapowiadane

i mogą odbywać się w dniu, w którym uczeń pisze zapowiedzianą pracę kontrolną z innego przedmiotu. Ilość

i zakres pytań muszą być tak dobrane, aby czas przewidziany na pisemną odpowiedź nie przekroczył 20 minut.

8. Oddanie i omówienie ocenionych prac pisemnych winno nastąpić w ciągu dwóch tygodni, kartkówek –

w ciągu tygodnia od dnia ich przeprowadzenia (z wyjątkiem wypadków losowych, choroby oraz okresu

sprawdzania egzaminów maturalnych w maju przez egzaminatorów). Uczeń otrzymuje ocenioną pracę pisemną do wglądu na lekcji, a rodzice na swój wniosek w czasie konsultacji lub wywiadówek. Nauczyciel przechowuje prace do końca danego roku szkolnego, za wyjątkiem prac uczniów zdających egzaminy po wakacjach.

§ 9

1. Nauczyciel jest zobowiązany, na podstawie pisemnej opinii poradni psychologiczno – pedagogicznej lub innej

poradni specjalistycznej, obniżyć wymagania edukacyjne w stosunku do ucznia, u którego stwierdzono specyficzne

trudności w uczeniu się lub deficyty rozwojowe, uniemożliwiające sprostanie wymaganiom edukacyjnym wynikającym z programu nauczania.

§ 10

Zasady współdziałania:

- Przestrzeganie zasad zawartych w PSO przez nauczyciela i ucznia;
- Indywidualne spotkania na konsultacjach (w ustalonym przez nauczyciela terminie lub w wyjątkowych sytuacjach innym);
- Indywidualizacja pracy z uczniem m.in. w przypadkach dłuższej usprawiedliwionej nieobecności, przygotowania do olimpiady, konkursów, dodatkowe zajęcia rozszerzające wiedzę geograficzną dla maturzystów;

Przedmiotowy system oceniania z Języków i technik programowania jest integralną częścią Wewnętrzny Szkolnego Systemu Oceniania w II Liceum Ogólnokształcącym im. A. Asnyka. Wszystkie sprawy nie ujęte w przedmiotowym systemie oceniania rozstrzygane będą zgodnie z wewnętrznym systemem oceniania.

Ocena jest jawna, sprawiedliwa i obiektywnie wystawiana.

Zabrania się w czasie lekcji korzystania z telefonów komórkowych (taki telefon może zostać przez nauczyciela odebrany po uprzednim wyłączeniu z prośbą o odbiór przez rodziców)

WYMAGANIA EDUKACYJNE

Kody binarne. Przyszłość informatyki

Wiedza. Uczeń:	Umiejętności. Uczeń:
Wie, jakich standardów używa się do szyfrowania poczty elektronicznej – PGP i S/MIME. Wie, czym są certyfikaty i klucze, w tym publiczny. Wie, na jakiej zasadzie funkcjonuje klucz publiczny symetryczny i asymetryczny.	Zabezpiecza plik w pakiecie MS Office i LibreOffice Writer (OpenOffice.org Writer). Umie szyfrować pocztę elektroniczną za pomocą darmowych programów szyfrujących.
Zna podstawy kodów binarnych. Wie, jaka jest ich budowa i jak zbudowane są kody wagowe. Zna metody zmiany postaci liczny z binarnej na dziesiętną i odwrotnie. Zna wielokrotności bajtu jako jednostki informacji. Zna charakterystyczne liczby związane z bajtem i jego wielokrotnością, np. liczbę możliwych do zapisania liczb za pomocą 8 bitów.	Zapisuje liczby dziesiętne w postaci binarnej w Naturalnym Kodzie Binarnym NKB. Konwertuje liczny z kodu NKB do dziesiętnego i odwrotnie. Używa kalkulatora dla informatyków, np. z systemu Windows, do konwersji liczb.
Wie, jak zapisywać liczby binarne mniejsze od 0	Zapisuje dowolne liczby całkowite mniejsze od zera w kodzie U2. Zamienia liczby dziesiętne na postać U2.
Zna podstawowe prawa arytmetyki binarnej	Umie dodawać, odejmować, mnożyć liczby w systemie dwójkowym NKB i U2.
Zna podstawowe prawa algebry Boole'a	Wykorzystuje prawa algebry Boole'a do wykonywania podstawowych działań, takich jak iloczyn i suma logiczna.
Wie, jakie znaczenie i zastosowanie w informatyce ma kod szesnastkowy, np. w zapisie adresów IP, MAC adresu, zawartości komórek pamięci itp. Zna	Zapisuje liczby w kodzie szesnastkowym i konwertuje je do postaci dziesiętnej i binarnej. Odczytuje za pomocą odpowiedniego programu lub polecenia systemowego, np. ipconfig, adresy sieciowe i przedstawia je

zależności pomiędzy liczbami w kodzie szesnastkowym i binarnym.	w postaci dziesiętnej.
Wie, w jaki sposób zapisać ułamkowe liczby binarne. Rozróżnia liczby stałe i zmiennoprzecinkowe.	Zapisuje ułamkowe liczby binarne i uzasadnia niedokładności takiego zapisu w porównaniu z liczbami dziesiętnymi.
Zna sposób cyfrowego zapisu znaków alfanumerycznych za pomocą kodu ASCII w odmianie ISO 8859-2 i CP-1250	Umie znaleźć kod ASCII danego znaku w tabeli oraz z wykorzystaniem edytora tekstu.
Zna historię maszyn liczących i komputerów i wie, jaki wpływ na współczesną technikę wywarły pierwsze konstrukcje mechaniczne i elektroniczne. Zna kierunki rozwoju współczesnej informatyki, w tym transmisje strumieniowe, telefonię internetową, zwiększanie przepustowości łączy i dynamiczny rozwój chmur internetowych.	Korzysta z nowych usług chmur informatycznych, posługuje się różnymi rodzajami komputerów, np. smartfonem, tabletem, netbookiem, czytnikiem e-booków itp.
Wie, czym jest e-learning, i wie, jakie znaczenie ma w życiu współczesnego człowieka. Zna strukturę lekcji i kursu elektronicznego.	Umie skonfigurować platformę zdalnego nauczania, np. Moodle. Tworzy strukturę kursów i lekcji w Moodle. Umieszcza w lekcjach materiały edukacyjne, tworzy testy, quizy itp. Umie zorganizować krótkie szkolenie w ramach e-learningu.
Zna prawa człowieka dotyczące prywatności i wykorzystania wizerunku. Wie, w jakich dokumentach należy szukać przepisów prawnych dotyczących bezpieczeństwa w sieci i ochrony dóbr osobistych.	Umie ocenić, czy dana informacja, zdjęcie itp. narusza czyjeś prawo do prywatności lub ochrony wizerunku.
Zna zasady, dzięki którym nie narusza się praw osobistych innych osób	Stosuje zasady bezpiecznego publikowania informacji i wizerunku.
Zna nazwy różnych rodzajów przestępczości elektronicznej	Umie opisać poszczególne rodzaje przestępczości elektronicznej i wie, na co jest narażona osoba, wobec której przestępca je zastosował.
Wie, jak zachować się w obliczu przestępstwa elektronicznego. Rozumie, na czym polegają zagrożenia związane z korzystaniem z sieci Internet, w tym uzależnienie od gier sieciowych, informacji, kontaktów towarzyskich w portalach społecznościowych, zakupów elektronicznych itp.	Stara się zabezpieczać przed atakami przestępców elektronicznych, wykorzystując wiedzę o sposobach ich działania.
Wie, że nielegalne udostępnianie plików z zawartością chronioną prawem autorskim bez zezwolenia jest przestępstwem. Zdaje sobie sprawę z tego, że nielegalne pobieranie plików chronionych prawem majątkowym i autorskim jest kradzieżą.	Umie legalnie kupować utwory muzyczne i inne pliki chronione prawem autorskim. Umie korzystać z kina internetowego. Posługuje się systemem płatności elektronicznej.
Wie, w jakim charakterze może pracować wykształcony informatyk. Wie, jakie wymagania stawiają przed in-	Umie opisać wymagania stawiane przed informatykiem pracującym na różnych stanowiskach, w tym programistą, administratorem sieci komputerowej i

formatykiem różne stanowiska pracy.	serwera, konserwatora sprzętu informatycznego itp.
Wie, jakimi cechami powinien charakteryzować się dobry informatyk	Umie ułożyć plan drogi do zdobycia zawodu informatyka.
Zna podstawowe rodzaje języków programowania i obszary ich zastosowania	Posługuje się edytorem środowiska programistycznego Free Pascal.
Wie, czym jest kompilator, interpreter oraz czym różni się edytor środowiska programistycznego od edytora tekstu	Korzysta z opcji edytora środowiska Free Pascal.
Zna nazwy kilku najpopularniejszych kompilatorów języków programowania, w tym Turbo Pascal, C++, Borland Pascal, Free Pascal	Stosuje prawidłową strukturę programu i umie odpowiednio i czytelnie ją zapisywać.
Zna przeznaczenie i działanie poszczególnych opcji edytora Free Pascal	
Wie, jak kompilować program	

Podstawy programowania

Wiedza. Uczeń:	Umiejętności. Uczeń:
Zna strukturę programu ułożonego w języku Pascal	Stosuje identyfikatory
Wie, czym są słowa kluczowe, procedury i funkcje, zmienne, stałe	Umie ułożyć, zapisać i skompilować prosty program, np. wykonując proste działania arytmetyczne z wprowadzaniem danych i wyprowadzaniem wyniku na ekran komputera.
Zna typy identyfikatorów i ich właściwości	Umie odczytać komunikaty kompilatora dotyczące wyników kompilacji, w tym rodzaje błędów, oraz poprawić błędy.
Wie, jak deklarować stałe i zmienne	Używa w programie prawidłowo zadeklarowanych stałych i zmiennych.
Wie, czym jest wyrażenie i zna jego typy	Układa proste programy z zastosowaniem różnych typów operacji: na liczbach całkowitych, rzeczywistych, typie znakowym i logicznym.
Zna podstawowe operacje arytmetyczne wykonywane przez program	Umie ułożyć program obliczający wartość według złożonego wzoru matematycznego z zachowaniem kolejności działań.
Zna rodzaje instrukcji, w tym: proste, warunkowe, wyboru, powtarzania	Układa proste programy z zastosowaniem różnych rodzajów instrukcji, w tym warunkowych, prostych, wyboru i powtarzania. Zagnieżdża w sobie instrukcje warunkowe.
Wie, jak zbudowana jest tablica, jakie typy danych można w niej zapisać. Zna typy tablicowe, w tym złożone.	Układa proste programy z zastosowaniem tablic.
Rozumie znaczenie i strukturę łańcuchów. Zna działanie funkcji dotyczących łańcuchów.	Stosuje w programach łańcuchy, np. do wyświetlania komunikatów.
Wie, czym są procedury i potrafi określić ich znaczenie w strukturze programu. Wie, czym różnią się zmienne lokalne od globalnych i na czym polega przekazywanie przez zmienną i przez	Definiuje procedury i podprogramy oraz umie je ułożyć oraz wywołać w programie.

wartość. Zna pojęcie podprogramu.	
Zna definicję i przeznaczenie funkcji i wie, czym różni się ona od procedury. Wie, czym są funkcje predefiniowane i jak znaleźć opis ich działania.	Prawidłowo używa funkcji w tworzeniu podprogramów. Używa funkcji predefiniowanych.
Wie, czym są złożone struktury danych, w tym rekordy	Używa rekordów do grupowania danych różnego typu.
Wie, czym są operacje wyjścia-wejścia i kiedy się je stosuje. Rozumie pojęcie fizycznych zbiorów danych i definiuje właściwości i rodzaje plików. Zna różnice pomiędzy dostępem sekwencyjnym a swobodnym do elementów pliku. Zna typy plików, w tym: tekstowe, zdefiniowane.	Układa programy wykorzystujące zapis i odczyt danych w pliku, w tym tekstowym. Deklaruje zmienne plikowe. Odczytuje poszczególne elementy z pliku, np. sekwencje znaków w pliku tekstowym. Stosuje procedury predefiniowane do operacji na plikach, w tym: Reset, Close, Seek, Eof.
Wie, czym jest debugowanie (śledzenie) i symulacja działania programu	Analizuje treść programu i znajduje błędy powodujące jego wadliwe działanie.
Definiuje pojęcie rekurencji w programie komputerowym	Układa program zawierający rekurencję, np. do obliczenia silni.
Wie, na czym polega modularyzacja programu	Stosuje moduły w programach złożonych.

Podstawy algorytmiki

Wiedza. Uczeń:	Umiejętności. Uczeń:
Definiuje pojęcie algorytmu oraz poszczególnych jego elementów	Podaje przykłady algorytmów różnych znanych czynności.
Wie, czym dla algorytmu są dane wejściowe i wyjściowe, oraz definiuje związek pomiędzy nimi. Wie, czym jest specyfikacja algorytmu.	Zapisuje specyfikację algorytmu: formułuje problem, sporządza jego opis, definiuje dane wejściowe i wyjściowe.
Zna sposoby zapisu algorytmu, w tym: lista kroków, pseudojęzyk, schemat blokowy	Opisuje kolejność czynności za pomocą różnych sposobów zapisu algorytmu.
Zna różne edytory schematów blokowych	Układa algorytmy za pomocą edytora np. Dia.
Wie, jakie są różnice pomiędzy algorytmem liniowym a rozgałęzionym	Edytuje i układa proste algorytmy rozgałęzione, używając edytora.
Zna pojęcie złożoności algorytmu, w tym złożoności pamięciowej i czasowej, pesymistycznej, oczekiwanej i optymistycznej	Analizuje algorytmy pod kątem złożoności.

Algorytmy działań na liczbach całkowitych

Wiedza. Uczeń:	Umiejętności. Uczeń:
Wie (powtórka z matematyki), czym są liczby pierwsze	Tworzy specyfikację problemu dotyczącą sprawdzania, czy dana liczba jest liczbą pierwszą. Układa algorytm programu testującego liczby i wykrywającego liczby pierwsze. Układa program realizujący ten algorytm.

Wie (powtórka z matematyki), czym są liczby doskonałe	Tworzy specyfikację problemu dotyczącą sprawdzania, czy dana liczba jest liczbą doskonałą. Układa algorytm programu testującego liczby i wykrywającego liczby doskonałe. Układa program realizujący ten algorytm.
Wie (powtórka z matematyki), czym jest największy wspólny dzielnik	Tworzy specyfikację problemu dotyczącą znajdowania największego wspólnego dzielnika metodą Euklidesa. Układa algorytm programu na podstawie metody Euklidesa. Układa program realizujący ten algorytm.
Definiuje ciąg Fibonacciego	Tworzy specyfikację problemu dotyczącą wyznaczania n-tego elementu ciągu Fibonacciego. Układa algorytm programu do obliczania n-tego elementu ciągu Fibonacciego. Układa program realizujący ten algorytm.
Wie, na czym polega metoda zachłanna wydawania reszty	Tworzy specyfikację problemu dotyczącą wydawania reszty w możliwie najmniejszej ilości nominałów (metodą zachłanną). Układa algorytm programu do wydawania reszty. Układa program realizujący ten algorytm.
Wie, na czym polega rozkładanie liczb na czynniki pierwsze	Tworzy specyfikację problemu dotyczącą rozkładu liczby na czynniki pierwsze. Układa algorytm programu znajdującego czynniki pierwsze danej liczby. Układa program realizujący ten algorytm.

Algorytmy wyszukiwania i sortowania

Wiedza. Uczeń:	Umiejętności. Uczeń:
Wie, czym różni się wyszukiwanie liniowe od wyszukiwania liniowego z wartownikiem	Tworzy specyfikację problemu dotyczącą wyszukiwania liniowego i liniowego z wartownikiem. Układa algorytm programu według specyfikacji wyszukiwania liniowego i liniowego z wartownikiem. Układa program realizujący ten algorytm.
Wie, czym jest tymczasowe minimum i maksimum i jakie mają one znaczenie dla algorytmu sortowania. Wie, czym jest proces rekurencji i na czym polega metoda dziel i zwyciężaj.	Tworzy specyfikację problemu dotyczącą największego lub najmniejszego elementu w zbiorze liczb. Układa algorytm programu według specyfikacji największego lub najmniejszego elementu w zbiorze liczb. Układa algorytm optymalny znajdujący największą i najmniejszą liczbę w zbiorze metodą rekurencyjną dziel i zwyciężaj. Układa program realizujący ten algorytm.
Wie, na czym polega sortowanie bąbelkowe i jaka jest złożoność obliczeniowa tej metody	Tworzy specyfikację problemu dotyczącą metody bąbelkowej sortowania liczb. Układa algorytm programu według specyfikacji metody bąbelkowej. Układa program realizujący ten algorytm.
Zna metodę sortowania przez wybieranie	Tworzy specyfikację problemu dotyczącą metody sortowania liczb przez wybieranie. Układa algorytm programu według specyfikacji metody sortowania przez wybieranie. Układa program realizujący ten algorytm.

Zna metodę sortowania przez wstawianie	<p>Tworzy specyfikację problemu dotyczącą metody sortowania liczb przez wstawianie.</p> <p>Układa algorytm programu według specyfikacji metody sortowania przez wstawianie.</p> <p>Układa program realizujący ten algorytm.</p>
Zna metodę sortowania QuickSort	<p>Tworzy specyfikację problemu dotyczącą metody sortowania szybkiego QuickSort.</p> <p>Układa algorytm programu według specyfikacji metody sortowania szybkiego.</p> <p>Układa program realizujący ten algorytm.</p>
Zna metodę sortowania przez scalanie metodą dziel i zwyciężaj	<p>Tworzy specyfikację problemu dotyczącą metody sortowania przez scalanie.</p> <p>Układa algorytm programu według specyfikacji metody sortowania przez scalanie.</p> <p>Układa program realizujący ten algorytm.</p>
Zna metodę sortowania przez zliczanie. Wie, że w tej metodzie nie występuje porównywanie liczb i że jest to metoda stabilna.	<p>Tworzy specyfikację problemu dotyczącą metody sortowania przez zliczanie.</p> <p>Układa algorytm programu według specyfikacji metody sortowania przez zliczanie.</p> <p>Układa program realizujący ten algorytm.</p>
Zna metodę sortowania kubełkowego	<p>Tworzy specyfikację problemu dotyczącą metody sortowania kubełkowego.</p> <p>Układa algorytm programu według specyfikacji metody sortowania kubełkowego.</p> <p>Układa program realizujący ten algorytm.</p>
Wie, czym jest zbiór uporządkowany i jaki ma to wpływ na wyszukiwanie. Wie, że można w tym przypadku stosować metodę rekurencyjną (dziel i rządź), jak i iteracyjną.	<p>Tworzy specyfikację problemu dotyczącą metody wyszukiwania binarnego w tablicy uporządkowanej.</p> <p>Układa algorytm programu według specyfikacji metody wyszukiwania binarnego dla tablic uporządkowanych.</p> <p>Układa program realizujący ten algorytm.</p>

Algorytmy numeryczne

Wiedza. Uczeń:	Umiejętności. Uczeń:
Zna metodę Newtona-Raphsona do obliczania wartości pierwiastka kwadratowego z zastosowaniem elementarnych działań matematycznych	Tworzy specyfikację problemu dotyczącą metody obliczania wartości pierwiastka kwadratowego metodą Newtona-Raphsona. Układa algorytm programu według specyfikacji metody obliczania wartości pierwiastka kwadratowego metodą Newtona-Raphsona. Układa program realizujący ten algorytm.
Zna schemat Hornera do obliczania wartości wielomianu. Zna jego niewielką złożoność obliczeniową.	Tworzy specyfikację problemu dotyczącą schematu Hornera. Układa algorytm programu według specyfikacji schematu Hornera. Układa program realizujący ten algorytm.
Zna metody numeryczne przedstawiania liczb za pomocą różnych kodów liczbowych, w tym dziesiętnego, NKB i Hex	Tworzy specyfikację problemu dotyczące zamiany postaci liczb. Układa algorytmy programów według specyfikacji zamiany postaci liczb dla różnych kodów liczbowych, w tym dziesiętnego, NKB i Hex. Układa program realizujący ten algorytm.
Zna schemat Hornera do szybkiego podnoszenia liczby do potęgi (od lewej do prawej)	Tworzy specyfikację problemu dotyczącego szybkiego podnoszenia liczby do potęgi za pomocą schematu Hornera. Układa algorytm programu według specyfikacji szybkiego podnoszenia liczby do potęgi za pomocą schematu Hornera. Układa program realizujący ten algorytm.
Wie, czym są miejsca zerowe funkcji, i wie, na czym polega metoda numeryczna ich wyznaczania, w tym metoda bisekcji – połowienia (oparta na twierdzeniu Bolzano-Cauchy`ego)	Tworzy specyfikację problemu dotyczącego znajdowania miejsca zerowego funkcji metodą bisekcji. Układa algorytm programu według specyfikacji znajdowania miejsca zerowego funkcji metodą bisekcji. Układa program realizujący ten algorytm.

Algorytmy operacji na tekstach

Wiedza. Uczeń:	Umiejętności. Uczeń:
Zna pojęcie palindromu i wie, jak sprawdzać, czy dany ciąg znaków go tworzy	Tworzy specyfikację problemu dotyczącego badania, czy ciąg znaków tworzy palindrom. Układa algorytm programu według specyfikacji badania, czy ciąg znaków tworzy palindrom. Układa program realizujący ten algorytm
Zna pojęcie anagramu i wie, jak sprawdzić, czy ciąg znaków go tworzy	Tworzy specyfikację problemu dotyczącego badania, czy ciąg znaków tworzy anagram. Układa algorytm programu według specyfikacji badania, czy ciąg znaków tworzy anagram. Układa program realizujący ten algorytm.
Wie, na czym polega porządkowanie leksykograficzne (alfabetyczne)	Tworzy specyfikację problemu dotyczącego porządkowania leksykograficznego metodą kubełkową. Układa algorytm programu według specyfikacji porządkowania leksykograficznego metodą kubełkową. Układa program realizujący ten algorytm.
Wie, na czym polegają metody naiwne	Tworzy specyfikację problemu dotyczącego wyszuki-

i Boyera-Moore'a stosowane do wyszukiwania wzorca w tekście	wania wzorca w tekście metodami naiwna i Boyera-Moore'a. Układa algorytm programu według specyfikacji wyszukiwania wzorca w tekście metodami naiwna i Boyera-Moore'a. Układa program realizujący ten algorytm.
Wie, czym jest wyrażenie ONP i jakie ma zastosowanie przy obliczaniu wartości wyrażenia	Tworzy specyfikację problemu dotyczącego obliczania wartości wyrażenia NOP. Układa algorytm programu według specyfikacji obliczania wartości wyrażenia NOP. Układa program realizujący ten algorytm.

Algorytmy kompresji i szyfrowania

Wiedza. Uczeń:	Umiejętności. Uczeń:
Zna zasadę stosowania szyfru przestawieniowego, np. Cezara i wieloalfabetowego	Tworzy specyfikację problemu dotyczącego szyfrowania metodami przestawieniowymi, np. Cezara. Układa algorytm programu według specyfikacji szyfrowania metodami przestawieniowymi, np. Cezara. Układa program realizujący ten algorytm.
Wie, czym są kody znaków o zmiennej długości na podstawie kodu Morse'a	Tworzy specyfikację problemu dotyczącego szyfrowania informacji kodem Morse'a jako przykładem kodu dla zmiennych o różnych długościach. Układa algorytm programu według specyfikacji kodowania kodem Morse'a. Układa program realizujący ten algorytm.
Wie, czym są kody znaków o zmiennej długości na podstawie kodu Huffmana i zna jego właściwości kompresowania	Tworzy specyfikację problemu dotyczącego szyfrowania informacji kodem Huffmana. Układa algorytm programu według specyfikacji kodowania kodem Huffmana. Układa program realizujący ten algorytm.
Wie, czym jest klucz publiczny i prywatny (RSA) oraz gdzie znajdują one zastosowanie. Zna wzory obliczania kluczy.	Umie zbudować klucz publiczny na podstawie wzorów.

Algorytmy badające własności geometryczne

Wiedza. Uczeń:	Umiejętności. Uczeń:
Zna metody badania warunków trójkąta	Tworzy specyfikację problemu dotyczącą badania warunków trójkąta. Układa algorytm programu badającego warunki trójkąta. Układa program realizujący ten algorytm.
Wie, na czym polega numeryczne badanie położenia punktu względem prostej i numeryczna prezentacja odcinka na płaszczyźnie	Tworzy specyfikację problemu dotyczącą badania położenia punktu względem prostej na płaszczyźnie i przynależności punktu do odcinka. Układa algorytm programu do badania położenia punktu względem prostej na płaszczyźnie i przynależności punktu do odcinka. Układa program realizujący ten algorytm.
Zna podstawy matematyczne dotyczące odcinków i prostych na płaszczyźnie	Tworzy specyfikację problemu dotyczącą badania odcinków na płaszczyźnie, a w szczególności ich przecina-

nie	nia się. Układa algorytm programu na podstawie specyfikacji. Układa program realizujący ten algorytm.
Wie, kiedy punkt należy do wielokąta na płaszczyźnie i jakie właściwości wielokąta pozwalają na opracowanie odpowiedniego algorytmu	Tworzy specyfikację problemu dotyczącą badania przynależności punktu do wielokąta. Układa algorytm programu do badania przynależności punktu do wielokąta. Układa program realizujący ten algorytm.
Wie, czym jest dywan Sierpińskiego, i wie, jak można go wykreślić na płaszczyźnie za pomocą metody rekurencyjnej	Tworzy specyfikację problemu dotyczącą wykreślenia dywanu Sierpińskiego za pomocą metody rekurencyjnej. Układa algorytm programu do wykreślenia dywanu Sierpińskiego za pomocą metody rekurencyjnej. Układa program realizujący ten algorytm.
Wie, czym jest płatek Kocha i jak można go wykreślić	Tworzy specyfikację problemu dotyczącą wykreślenia płatków Kocha na płaszczyźnie. Układa algorytm programu wykreślenia płatków Kocha na płaszczyźnie. Układa program realizujący ten algorytm.
Rozumie pojęcie drzewa binarnego i wie, jak ono powstaje	Tworzy specyfikację problemu dotyczącą wykreślenia drzewa binarnego na płaszczyźnie. Układa algorytm programu realizującego wykreślenie drzewa binarnego na płaszczyźnie. Układa program realizujący ten algorytm.

Indywidualna i zespołowa realizacja projektu informatycznego

Wiedza. Uczeń:	Umiejętności. Uczeń:
Wie, jak wygląda cykl tworzenia oprogramowania komputerowego i zna funkcje członków przykładowego zespołu	Planuje pracę zespołu informatyków pod kątem tematu i wielkości projektu.
Wie, jak tworzy się specyfikację wymagań na podstawie problemu przedstawionego przez klienta	Opracowuje specyfikacje prostych projektów informatycznych.
Wie, czym jest dokument projektu, jakie ma on znaczenie dla sprawnej pracy zespołu i jakie zasady obowiązują przy jego tworzeniu	Opracowuje dokument prostego projektu informatycznego z uwzględnieniem poznanych zasad.
Wie, jakie ustalenia należy poczynić, by praca zespołu przebiegała sprawnie, w tym: ujednoczenie nazw, formatu komentarzy, testowania i sposobu zapisu programu	Opracowuje tabele opisujące jednolite warunki sposobu tworzenia projektu
Wie, jak wielkie znaczenie ma testowanie gotowego programu i jak powinno być ono zorganizowane	Testuje swoje programy zgodnie z poznanymi zasadami. Układa scenariusz testowania programu.
Wie, co powinna zawierać dokumentacja techniczna programu komputerowego	Opracowuje dokumentację prostego programu komputerowego i zapisuje ją w dokumencie tekstowym.
Zna zasady tworzenia dokumentacji użytkownika, w tym: instrukcji obsługi programu. Zna procedury przekazania	Sporządza instrukcję obsługi programu z wykorzystaniem elementów graficznych, zrzutów ekranowych itp. Przeprowadza instalację programu i wstępny instruktaż

oprogramowania klientowi.	obsługi.
---------------------------	----------

Bazy danych

Wiedza. Uczeń:	Umiejętności. Uczeń:
Rozumie pojęcia: tabele, wiersze i klucze i wie, jakie role pełnią w bazie danych	Opisuje i przedstawia na przykładach znaczenie kluczy dla bazy danych.
Wie, czym różnią się pierwsza postać normalna tabeli od drugiej i dlaczego po przekształceniu powstaje więcej tabel. Wie, na czym polega redundancja danych w tabeli i jak ją eliminować.	Przekształca tabele z pierwszej postaci do drugiej, eliminuje redundancję w tabelach (przeprowadza proces normalizacji).
Wie, jak wygląda 3 i 4 postać normalna tabeli w bazie danych. Wie, jak eliminować dane nie należące do klucza.	Przekształca proste tabele do postaci 3 i 4 normalnej, eliminując zależności wielowarstwowe, nie dotyczące klucza głównego.
Wie, na czym polegają i jakie znaczenie dla budowy bazy mają relacje między tabelami. Rozumie relacje 1..1 i 1..wielu. Zna pojęcie klucza obcego.	Określa relacje pomiędzy tabelami w zależności o wymagań stawianych bazie.
Zna kilka programów, w których występują systemy baz danych pozwalające na ich tworzenie za pomocą kreatorów, np. OpenOffice.org Base, LibreOffice Base, MS Access	Posługuje się menu i kreatorem systemu do tworzenia relacyjnych baz danych, np. OpenOffice.org Base.
Zna pojęcia: formularz, kwerenda, raport	Tworzy podstawową strukturę bazy danych za pomocą kreatora, korzystając z własnego projektu prostej bazy.
Wie, jak utworzyć bazę danych z pominięciem kreatora	Wykorzystuje system np. OpenOffice.org Base do tworzenia relacyjnych baz danych z pominięciem kreatora. Modyfikuje tabele i inne elementy bazy danych.
Wie, czym są indeks i relacje i jakie mają znaczenie dla funkcjonowania bazy danych	Indeksuje tabele bazy danych i tworzy relacje pomiędzy nimi.
Wie, jak wprowadzać dane do bazy danych poprzez arkusz danych lub formularz	Opracowuje formularze do wprowadzania danych do bazy. Wprowadza dane do bazy danych.
Wie, jak dobrać kryteria podczas wyszukiwania danych w bazie	Korzysta z opcji wyszukiwania danych w systemie bazy danych, np. OpenOffice.org Base.
Wie, jak dobrać parametry sortowania i filtrowania danych w bazie	Sortuje i filtruje dane w tabelach bazy danych.
Wie, jak wykorzystać kreatora do budowania kwerendy dla bazy danych. Zna opcje pozwalające na budowanie kwerend bez użycia kreatora.	Buduje kwerendy dla bazy danych w systemie bazodanowym, np. OpenOffice Base, przy pomocy kreatora i z jego pominięciem. Odczytuje przy ich pomocy dane z bazy.
Wie, jak drukować dane z bazy danych	Kreuje formularze wydruku dla systemu bazy danych, w tym także z zastosowanie styli.
Wie, czym jest SQL i gdzie znajduje zastosowanie	Wykorzystuje opcje systemu bazy danych, np. OpenOffice.org Base, do opracowania kwerendy w języku SQL.