Wymagania edukacyjne z informatyki dla klasy: 2b

Liceum Ogólnokształcące im. B.Prusa w Skierniewicach
rok szkolny 2019/2020

nauczyciel: Aleksandra Daciów

Wymagania programowe na poszczególne oceny – IV etap edukacyjny – przygotowane na podstawie treści zawartych w podstawie programowej oraz w podręczniku „Informatyka Europejczyka” wydawnictwo Helion, zakres rozszerzony

Wymagania ogólne na poszczególne oceny

Ocena celująca (6) – uczeń bierze udział w konkursach związanych z informatyką i odnosi w nich sukcesy; samodzielnie i bezbłędnie wykonuje ćwiczenia z podręcznika oraz zadania dodatkowe, bierze udział w projektach; na lekcjach jest aktywny; posiada wiadomości i umiejętności wykraczające poza opisane w planie wynikowym; pomaga kolegom w pracy, nauczycielowi w prowadzeniu lekcji oraz nauczycielom na innych lekcjach w wykorzystaniu technologii informacyjnej i komunikacyjnej.

Ocena bardzo dobra (5) – uczeń samodzielnie i bezbłędnie wykonuje ćwiczenia z podręcznika oraz łatwiejsze zadania dodatkowe; na lekcjach jest aktywny; posiada wiadomości i umiejętności opisane w planie wynikowym; pomaga kolegom w pracy.

Ocena dobra (4) – uczeń samodzielnie wykonuje wszystkie ćwiczenia z podręcznika; na lekcjach jest aktywny; posiada wiadomości i umiejętności opisane w planie wynikowym.

Ocena dostateczna (3) – uczeń samodzielnie wykonuje łatwiejsze ćwiczenia z podręcznika, czasami z pomocą nauczyciela; stara się pracować systematycznie, robi postępy; posiada wiadomości i umiejętności opisane w planie wynikowym.

Ocena dopuszczająca (2) – uczeń wykonuje łatwe ćwiczenia z podręcznika z pomocą nauczyciela; posiada wiadomości i umiejętności opisane w planie wynikowym; ma problemy z systematycznością, niemniej jednak nie przekreśla to możliwości postępów w ciągu dalszej nauki.

Wymagania programowe na poszczególne oceny szkolne

Korelacja pomiędzy zestawem wymagań a oceną przedstawia się następująco:

– spełnienie wymagań koniecznych (K) odpowiada ocenie dopuszczającej;
– spełnienie wymagań koniecznych i podstawowych (P) odpowiada ocenie dostatecznej;
– spełnienie wymagań koniecznych, podstawowych i rozszerzających (R) odpowiada ocenie dobrej;
– spełnienie wymagań koniecznych, podstawowych, rozszerzających i dopełniających (D) odpowiada ocenie bardzo dobrej;
– spełnienie wymagań koniecznych, podstawowych, rozszerzających, dopełniających i wykraczających (W) odpowiada ocenie celującej.

Wymagania konieczne stanowią bazę do rozszerzania i pogłębiania dalszej wiedzy, nie ma zatem możliwości wystawienia uczniowi oceny pozytywnej, jeśli ich nie spełnia.

Umiejętności nabyte przez ucznia	Wymagania				
	K	D	P	R	W
Algorytmika i programowanie					
Rozumie pojęcie algorytm.	●	●	●	●	●
Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczanymi przedmiotami itp.		●	●	●	●
Poprawnie definiuje problem i formułuje jego specyfikację	●	●	●	●	●
Dokonuje analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją.			●	●	●
Realizuje algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków.			●	●	●
Wyodrębnia elementy składowe algorytmu	●	●	●	●	●
Rozumie pojęcia: język programowania, translator, kompilator, interpreter		●	●	●	●
Zna klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego poziomu i wysokiego poziomu.		●	●	●	●
Potrafi wymienić i sklasyfikować podstawowe języki programowania		●	●	●	●
Zna i rozumie podstawowe zasady i metody programowania		●	●	●	●
Korzysta z wybranego środowiska programistycznego (np. kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy			●	●	●
Zna i stosuje etapy rozwiązywania zadań za pomocą komputera.,				●	●

Zna różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.				●	●
Dobiera właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu				●	●
Kompiluje i uruchamia przykładowe programy napisane w różnych językach programowania.			●	●	●
Zna podstawową strukturę programu	●	●	●	●	●
Korzysta w programach z podstawowych operacji wejścia i wyjścia,		●	●	●	●
Potrafi deklarować zmienne i wykorzystywać je w programach			●	●	●
Potrafi deklarować stałe i wykorzystywać je w programach.			●	●	●
Stosuje komentarze przy pisaniu programów				●	●
Zna priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je podczas pisania programów				●	●
Zna i potrafi stosować podstawowe konstrukcje algorytmiczne, w tym instrukcję pustą, instrukcję przypisania, instrukcję złożoną,		●	●	●	●
Zna i potrafi stosować podstawowe konstrukcje algorytmiczne, w tym instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne		●	●	●	●
Stosuje w programach podstawowe konstrukcje algorytmiczne			●	●	●
Wykonuje operacje na prostych typach danych		●	●	●	●
Przedstawia algorytmy liniowe i warunkowe w postaci listy kroków, schematu blokowego i programu				●	●
Potrafi generować liczby losowe w wybranym języku programowania.					●
Realizuje algorytmy: sprawdzanie warunku trójkąta, określanie prostopadłości i równoległości prostych, obliczanie długości odcinka, obliczanie odległości punktu od prostej, obliczanie odległości punktów na płaszczyźnie, badanie położenia punktu względem prostej, badanie przynależności punktu do odcinka				●	●
Realizuje algorytmy: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe			●	●	●
Rozumie pojęcia: iteracja, wzór iteracyjny	●	●	●	●	●
Potrafi definiować iterację		●	●	●	●
Stosuje metodę iteracji przy realizacji algorytmów			●	●	●
Przedstawia algorytmy iteracyjne w postaci listy kroków, schematu blokowego i programu				●	●
Dobiera odpowiednie struktury danych umożliwiające rozwiązanie postawionego problemu					●

Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem			●	●	●
Rozumie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu	●	●	●	●	●
Rozumie podstawowe zasady i cel strukturalizacji programu		●	●	●	●
Zna sposoby przekazywania parametrów w funkcjach i świadomie stosuje je podczas pisania programów			●	●	●
Stosuje w programach funkcje z parametrami				●	●
Zna typy przeładowania funkcji i stosuje je w programach				●	●
Rozumie pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna	●	●	●	●	●
Potrafi podać przykłady zastosowania rekurencji		●	●	●	●
Identyfikuje wykorzystaną metodę rekurencji w przykładowych algorytmach			●	●	●
Potrafi definiować zależności rekurencyjne oraz odpowiednie funkcje rekurencyjne		●	●	●	●
Rozumie pojęcia: strukturalne typy danych, abstrakcyjne typy danych	●	●	●	●	●
Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe		●	●	●	●
Realizuje algorytm sprawdzający, czy dana liczba jest pierwsza		●	●	●	●
Rozumie pojęcie liczba pierwsza		●	●	●	●
Realizuje algorytmy rekurencyjne: wyznaczanie elementów rekurencyjnych ciągów liczbowych, wieże Hanoi itp				●	●
Potrafi dokonywać zamiany metody rekurencyjnej wykorzystanej w algorytmie na iteracyjną					●
Rozumie pojęcia: system liczbowy, pozycyjny system liczbowy		●	●	●	●
Realizuje algorytm sprawdzający, czy dana liczba jest pierwsza		●	●	●	●
Realizuje algorytm generujący liczby pierwsze — sito Eratostenesa			●	●	●
Wykorzystuje arkusz kalkulacyjny do realizacji algorytmów,				●	●
Dokonuje rozkładu liczby na czynniki pierwsze.			●	●	●
Zna reprezentacje danych liczbowych w komputerze, w tym reprezentację binarną liczb całkowitych i niecałkowitych, stałopozycyjną reprezentację liczb, zmiennopozycyjną reprezentację liczb, pojęcie mantysy i cechy			●	●	●
Realizuje algorytm zamiany liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera				●	●

Zna definicje systemu dwójkowego (binarnego), ósemkowego (oktalnego) i szesnastkowego (heksadecymalnego).		●	●	●	●
Wykorzystuje metody liniowe przy przeszukiwaniu ciągu liczbowego			●	●	●
Rozpoznaje metodę programowania liniowego w poznanych algorytmach		●	●	●	●
Realizuje algorytm znajdowania maksymalnego elementu w ciągu liczbowym				●	●
Rozumie pojęcie monotoniczność ciągu liczbowego	●	●	●	●	●
Realizuje algorytm znajdowania lidera w zbiorze			●	●	●
Realizuje algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny				●	●
Realizuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący, czy malejący		●	●	●	●
Wyjaśnia źródła błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny				●	●
Realizuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący, czy malejący	●	●	●	●	●
Rozumie pojęcie metoda „dziel i zwyciężaj	●	●	●	●	●
Potrafi definiować i stosować metodę „dziel i zwyciężaj” w odpowiednich sytuacjach		●	●	●	●
umie zaprojektować prezentację dostosowując ja do rodzaju odbiorców i prezentowanego tematu, na przykład najciekawszego miejsca regionu			●	●	●
Identyfikuje metodę „dziel i zwyciężaj” w przykładowych algorytmach		●	●	●	●
Realizuje algorytm jednoczesnego znajdowania minimalnego i maksymalnego elementu		●	●	●	●
Realizuje algorytm obliczający pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów			●	●	●
Potrafi definiować dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne			●	●	●
Pisze programy z wykorzystaniem dynamicznych struktur danych			●	●	●
Pisze programy z wykorzystaniem dynamicznych struktur danych				●	●
Konstruuje binarne drzewo poszukiwań oraz wykonuje operacje dodawania i sortowania elementów drzewa				●	●
Identyfikuje metodę programowania zachłannego w przykładowych algorytmach,			●	●	●
Realizuje algorytm znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metoda połowienia przedziałów				●	●

ealizuje algorytm obliczający wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona)					●
Rozumie pojęcie programowanie zachłanne.		●	●	●	●
Realizuje algorytm zachłanny dla problemu plecakowego,				●	●
Przetwarzanie informacji liczbowych w postaci tabelarycznej					
Rozumie pojęcia: kryptografia, kryptoanaliza, algorytmy kryptograficzne	●	●	●	●	●
Potrafi definiować typ łańcuchowy	●	●	●	●	●
Realizuje przykładowe algorytmy kryptograficzne, w tym algorytmy symetryczne, asymetryczne itp		●	●	●	●
Potrafi wymienić podstawowe metody szyfrowania			●	●	●
Zna zasady zapisu wyrażenia podanego w postaci ONP		●	●	●	●
Potrafi definiować tablice struktur			●	●	●
Zna podstawowe operacje na plikach i korzysta z nich.			●	●	●
Realizuje programy z wykorzystaniem tablic struktur				●	●
Rozumie pojęcie inżynieria oprogramowania	●	●	●	●	●
Potrafi określić kolejne fazy konstruowania oprogramowania		●	●	●	●
Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych i umiarkowanie złożonych zadań z różnych dziedzin				●	●
Znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin		●	●	●	●
Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach			●	●	●
Realizuje programy z wykorzystaniem typu plikowego, m.in. importuje dane z plików zewnętrznych i eksportuje wyniki do plików zewnętrznych				●	●
Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach		●	●	●	●
Zna i realizuje etapy przetwarzania plików				●	●