

Programowanie

dla szkół ponadpodstawowych.

**Opis założonych osiągnięć ucznia – przykłady wymagań
na poszczególne oceny szkolne
w klasie pierwszej**

II LO im. E. Plater

Klasa 1C

Jedna godzina tygodniowo

mgr E. Łyczko-Siller

Rok szkolny 2023/2024

Prezentacja algorytmu w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>Wie, co to jest algorytm.</p> <p>Określa dane do zadania oraz wyniki.</p> <p>Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym.</p> <p>Potrafi narysować (odręcznie) schemat blokowy algorytmu.</p> <p>Potrafi napisać prosty program, wyświetlający napis na ekranie monitora.</p>	<p>Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy.</p> <p>Zna pojęcie specyfikacji zadania.</p> <p>Zna wybrane sposoby prezentacji algorytmów.</p> <p>Przedstawia algorytm w postaci listy kroków.</p> <p>Określa pojęcia <i>program komputerowy, język programowania</i>.</p> <p>Zapisuje prosty algorytm w wybranym języku programowania. Potrafi go skompilować i uruchomić.</p>	<p>Określa zależności między problemem, algorytmem a programem komputerowym.</p> <p>Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady.</p> <p>Przedstawia dokładną specyfikację dowolnego zadania.</p> <p>Analizuje poprawność budowy schematu blokowego.</p> <p>Wyjaśnia pojęcia: <i>program wynikowy, kompilacja, translacja, interpretacja</i>.</p> <p>Realizuje przykładowy algorytm liniowy w języku programowania.</p> <p>Wykonuje program i testuje go, podstawiając różne dane.</p>	<p>Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji).</p> <p>Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych.</p> <p>Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego.</p> <p>Analizuje działanie algorytmu dla przykładowych danych.</p> <p>Potrafi posłużyć się kompilatorem języka.</p> <p>Potrafi wskazać i poprawić błędy w programie.</p>	<p>Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji).</p> <p>Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia.</p> <p>Potrafi samodzielnie zapoznać się z kompilatorem wybranego języka programowania.</p> <p>Samodzielnie pisze program realizujący algorytm.</p>

Podstawowe zasady programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>Zna klasyfikację języków programowania.</p> <p>Zna ogólną budowę programu i najważniejsze elementy języka programowania – słowa kluczowe, instrukcje, wyrażenia, zasady składni.</p> <p>Potrafi zrealizować prosty algorytm liniowy i z warunkami w języku wysokiego poziomu; potrafi skompilować i uruchomić program.</p>	<p>Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze.</p> <p>Rozumie znaczenie i działanie podstawowych instrukcji (m.in. iteracyjnych, warunkowych) wybranego języka programowania wysokiego poziomu.</p> <p>Wie, na czym polega programowanie strukturalne.</p> <p>Rozróżnia i poprawia błędy kompilacji i błędy wykonania.</p> <p>Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu.</p>	<p>Wymienia i omawia modele programowania.</p> <p>Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania.</p> <p>Zna i stosuje instrukcje wyboru.</p> <p>Prezentuje wybrane algorytmy iteracyjne w postaci programu komputerowego.</p> <p>Zna rekurencyjne realizacje prostych algorytmów.</p> <p>Rozumie i stosuje zasady programowania strukturalnego.</p> <p>Deklaruje procedury i funkcje bez parametrów.</p> <p>Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach i funkcjach.</p>	<p>Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów.</p> <p>Deklaruje procedury i funkcje z parametrami.</p> <p>Wie, jakie znaczenie ma zasięg zmiennej.</p> <p>Definiuje funkcje rekurencyjne. Potrafi prezentować algorytmy rekurencyjne w postaci programu.</p> <p>Zapisuje w postaci programu wybrane algorytmy sortowania, algorytmy na tekstach, definiując odpowiednie procedury lub funkcje.</p>	<p>Ocenia efektywność działania programu.</p> <p>Wskazuje podobieństwa i różnice dotyczące tworzenia programów zapisanych w różnych językach programowania; wyjaśnia działanie poszczególnych instrukcji, sposób deklaracji zmiennych.</p> <p>Sprawnie definiuje i stosuje procedury i funkcje w programach.</p> <p>Sprawnie korzysta z dodatkowej, fachowej literatury.</p> <p>Rozwiązuje przykładowe zadania z matury i olimpiady informatycznej.</p>
<p>Wymienia przykłady prostych struktur danych.</p> <p>Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.</p>	<p>Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości.</p> <p>Rozróżnia struktury danych: proste i złożone. Podaje przykłady.</p>	<p>Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach.</p> <p>Deklaruje typ tablicowy i łańcuchowy.</p>	<p>Rozumie, na czym polega dobór struktur danych do algorytmu. Wczytuje i wyprowadza elementy tablicy. Wprowadza dane tekstowe.</p> <p>Tworzy programy, dobierając odpowiednie struktury danych do programu.</p>	<p>Dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu.</p>

Techniki algorytmiczne i wybrane algorytmy				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>Określa sytuacje warunkowe.</p> <p>Podaje przykłady zadań, w których występują sytuacje warunkowe.</p> <p>Wie, na czym polega powtarzanie tych samych operacji.</p> <p>Potrafi omówić na konkretnym przykładzie algorytm znajdowania najmniejszego z trzech elementów.</p> <p>Potrafi zapisać prosty algorytm i program realizujący pętlę.</p>	<p>Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami).</p> <p>Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady.</p> <p>Wie, od czego zależy liczba powtórzeń.</p> <p>Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych.</p> <p>Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych.</p> <p>Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy.</p> <p>Omawia wybrane algorytmy sortowania.</p> <p>Omawia wybrany algorytm na tekstach (np. tworzenie anagramów).</p>	<p>Analizuje algorytmy, w których występują powtórzenia (iteracje).</p> <p>Ocenią zgodność algorytmu ze specyfikacją.</p> <p>Zna sposoby zakończenia iteracji. Określa kroki iteracji.</p> <p>Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego.</p> <p>Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci.</p> <p>Zna iteracyjną postać algorytmu Euklidesa.</p> <p>Zna rekurencyjną realizację wybranego algorytmu, np. silni.</p> <p>Zna przynajmniej dwie techniki sortowania (np. bąbelkowe, przez wybór) i zapisuje wybrany algorytm w postaci programu komputerowego.</p> <p>Omawia wybrane algorytmy na tekstach.</p> <p>Potrafi wyjaśnić, na czym polega wydawanie reszty metodą zachłanną i napisać listę kroków tego algorytmu.</p>	<p>Zapisuje algorytmy z pętlą zagnieżdżoną.</p> <p>Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji.</p> <p>Zna inne algorytmy sortowania, np. pozycyjne, przez wstawianie.</p> <p>Wskazuje różnicę między rekurencją a iteracją.</p> <p>Zna rekurencyjną realizację wybranych algorytmów, np. obliczania silni i algorytm Euklidesa.</p> <p>Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną.</p> <p>Zapisuje wybrany algorytm na tekstach (np. tworzenie anagramów, zliczanie znaków w tekście, sprawdzanie, czy dany ciąg jest palindromem) w postaci programu komputerowego.</p> <p>Zapisuje algorytm wydawania reszty metodą zachłanną w postaci programu komputerowego.</p>	<p>Rozumie dokładnie technikę rekurencji (znaczenie stosu).</p> <p>Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję.</p> <p>Zna trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, szukanie wzorca w tekście. Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu).</p> <p>Korzysta samodzielnie z dodatkowej literatury fachowej.</p>

Elementy analizy algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wymienia własności algorytmów. Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.	Zna i omawia własności algorytmów. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją.	Określa liczbę prostych działań zawartych w algorytmie. Rozumie, co to jest złożoność czasowa algorytmu . Rozróżnia złożoność czasową i pamięciową.	Wie, jak ocenić złożoność pamięciową algorytmu. Wie, kiedy algorytm jest efektywny.	Określa złożoność czasową i pamięciową wybranych algorytmów

Ocenę niedostateczną otrzymuje uczeń, który nie osiągnął poziomu wymagań na ocenę dopuszczającą.

Ocenę celującą otrzymuje uczeń , który :

- Wykazuje szczegółowe zainteresowanie przedmiotem,
- Samodzielnie formułuje problemy, jest dociekliwy i konsekwentnie dąży do ich rozwiązywania,
- Bierze udział w konkursach i olimpiadach przedmiotowych.