

Wymagania edukacyjne z informatyki - poziom rozszerzony klasa 3

Ocenę **dopuszczającą** otrzymuje uczeń, który:

Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe. Wie, na czym polega powtarzanie tych samych operacji. Potrafi omówić, na przykładzie, algorytm znajdowania najmniejszego z trzech elementów. Potrafi zanalizować przebieg algorytmu dla przykładowych danych i ocenić w ten sposób jego poprawność. Zna klasyfikację języków programowania. Zna ogólną budowę programu i najważniejsze elementy języka - słowa kluczowe, instrukcje, wyrażenia, zasady składni. Potrafi zrealizować prosty algorytm liniowy w języku wysokiego poziomu; potrafi skompilować i uruchomić gotowy program. Wymienia przykłady prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.

Ocenę **dostateczną** otrzymuje uczeń, który opanował materiał na ocenę dopuszczającą oraz:

Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm w postaci listy kroków. Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Korzysta (w stopniu podstawowym) z programu edukacyjnego do symulacji działania algorytmu skonstruowanego w postaci schematu blokowego. Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami). Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń. Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy. Omawia i analizuje wybrane techniki sortowania w postaci gotowych schematów blokowych, skonstruowanych w programie edukacyjnym. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie. Zapisuje program w czytelnej postaci - stosuje wcięcia, komentarze. Rozumie pojęcia: implementacja, kompilacja, uruchomienie, testowanie. Rozumie znaczenie i działanie podstawowych instrukcji wybranego języka programowania wysokiego poziomu. Rozróżnia i poprawia błędy kompilacji i błędy wykonania. Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu. Zna podstawowe zasady poprawnego programowania; testuje tworzone programy; wie, jak unikać problemów, takich jak np. zapętlenie się programu. Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady. Deklaruje typy złożone.

Ocenę **dobrą** otrzymuje uczeń, który opanował materiał na ocenę dostateczną oraz:

Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów. Przedstawia dokładną specyfikację dowolnego zadania. Zna znaczenie i działanie instrukcji symbolicznego języka programowania (pseudojęzyka). Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci. Potrafi skonstruować algorytm z warunkami zagnieżdżonymi i pętlą za pomocą programu edukacyjnego. Analizuje algorytmy, w których występują powtórzenia (iteracje). Zna sposoby zakończenia iteracji. Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego. Zna iteracyjną postać algorytmu Euklidesa. Zna przynajmniej dwie techniki sortowania, np. bąbelkowe i przez wybór. Określa problemy, w których występuje rekurencja i podaje przykłady "zjawisk rekurencyjnych" - wziętych z życia i zadań szkolnych. Zna rekurencyjną realizację wybranego algorytmu, np. silni. Rozumie, co to jest złożoność algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania. Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna rekurencyjne realizacje prostych algorytmów. Rozumie i stosuje zasady programowania strukturalnego. Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach. Wie, jakie znaczenie ma zasięg działania zmiennej. Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą. Zna zasady działania wybranych algorytmów sortowania. Zna podstawowe procedury graficzne, potrafi narysować na ekranie wykres funkcji i podstawowe figury geometryczne.

Ocenę **bardzo dobrą** otrzymuje uczeń, który opanował materiał na ocenę dobrą oraz:

Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji), m.in. w pseudojęzyku. Zapisuje algorytmy z pętlą zagnieżdżoną. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Stosuje swobodnie oprogramowanie edukacyjne do graficznej prezentacji i analizy algorytmów. Zna metodę "dziel i zwyciężaj", algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji. Zna inne algorytmy sortowania, np. kulekowe, przez wstawianie. Zna przynajmniej jeden algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego. Wskazuje różnicę między rekurencją a iteracją. Zna rekurencyjną realizację wybranych algorytmów, np. silnię i algorytm Euklidesa. Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną. Potrafi porównać złożoność różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest uniwersalny. Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów. Potrafi realizować nawet bardzo złożone algorytmy, stosować procedury graficzne w realizacji skomplikowanych zadań - np. tworzyć własne animacje. Potrafi prezentować algorytmy rekurencyjne w postaci programu; potrafi zamienić rozwiązanie iteracyjne algorytmu na rekurencyjne. Zapisuje w postaci programu wybrane algorytmy sortowania. Opracowuje złożony program w kilkusobowej grupie - umie podzielić zadania, ustalić sposoby przekazywania danych pomiędzy procedurami. Zabezpiecza tworzone programy przed wprowadzeniem przez użytkownika błędnych danych. Rozumie, na czym polega dobór struktur danych do algorytmu. Potrafi zastosować rekordowy typ danych.

Ocenę **celującą** otrzymuje uczeń, który opanował materiał na ocenę bardzo dobrą oraz:

Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Potrafi trafnie dobrać do algorytmu sposób prezentacji. Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi zaproponować własny pseudojęzyk (postać instrukcji i zasady składni). Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję. Zna trudniejsze algorytmy, np. algorytm trwałego małżeństwa, wieże Hanoi, problem ośmiu hetmanów. Zna inne techniki sortowania, np. sortowanie przez scalanie ciągów i metodę szybką. Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu). Zna inne algorytmy numeryczne, np. wyznaczanie miejsca zerowego funkcji. Korzysta samodzielnie z dodatkowej literatury. Ocenia złożoność czasową i pamięciową algorytmu. Zna odpowiednie wzory. Ocenia efektywność działania programu. Wie, na czym polega programowanie obiektowe i zdarzeniowe. Potrafi stosować techniki programowania dynamicznego lub programowania obiektowego. Zna i rozumie podobieństwa i różnice w strukturze programu zapisanego w różnych językach programowania - w deklaracji zmiennych i procedur, w składni i zasadach działania poszczególnych procedur. Sprawnie korzysta z dodatkowej, fachowej literatury. Zna dynamiczne struktury danych. Potrafi zastosować zmienne typu wskaźnikowego w zadaniach. Zna struktury listowe, np. stos, kolejkę, listę. Rozumie i potrafi zastosować typ obiektowy.

