

Pracownia podstaw programowania – wymagania edukacyjne

Ocena dopuszczająca (2):

Uczeń:

- uruchamia środowisko programistyczne i tworzy nowy projekt;
- pisze proste programy sekwencyjne (bez rozgałęzień i pętli);
- deklaruje zmienne podstawowych typów danych;
- realizuje podstawowe operacje wejścia/wyjścia (wprowadzanie i wyświetlanie danych);
- stosuje podstawowe operatory arytmetyczne;
- kompiluje program i uruchamia go;
- rozpoznaje podstawowe błędy składniowe wskazane przez kompilator;
- zapisuje i odczytuje pliki z kodem źródłowym.

Ocena dostateczna (3):

Uczeń spełnia wymagania na ocenę dopuszczającą oraz:

- implementuje proste instrukcje warunkowe (if, if-else);
- tworzy programy z podstawowymi pętlami (for, while);
- stosuje operatory relacyjne i logiczne w warunkach;
- deklaruje i inicjalizuje tablice jednowymiarowe;
- pisze programy realizujące proste algorytmy (suma liczb, średnia, minimum/maksimum);
- tworzy proste funkcje bez parametrów lub z parametrami przekazywanymi przez wartość;
- formatuje kod zgodnie z podstawowymi konwencjami (wcięcia, nawiasy);
- usuwa proste błędy składniowe na podstawie komunikatów kompilatora.

Ocena dobra (4):

Uczeń spełnia wymagania na ocenę dostateczną oraz:

- implementuje zagnieżdżone instrukcje warunkowe i pętle;
- tworzy programy wykorzystujące tablice wielowymiarowe;

- pisze funkcje z parametrami i wartościami zwracanymi;
- implementuje przekazywanie parametrów przez referencję;
- stosuje wskaźniki w podstawowym zakresie;
- realizuje walidację danych wejściowych;
- debuguje programy używając podstawowych technik (wypisywanie wartości zmiennych);
- dokumentuje kod za pomocą komentarzy;
- tworzy programy rozwiązujące bardziej złożone problemy (sortowanie bąbelkowe, wyszukiwanie liniowe);
- obsługuje podstawowe przypadki błędnych danych.

Ocena bardzo dobra (5):

Uczeń spełnia wymagania na ocenę dobrą oraz:

- implementuje funkcje rekurencyjne (silnia, Fibonacciego, NWD);
- sprawnie posługuje się wskaźnikami i arytmetyką wskaźników;
- stosuje dynamiczną alokację pamięci;
- tworzy programy modularne z podziałem kodu na funkcje;
- implementuje różne algorytmy sortowania;
- realizuje algorytmy przeszukiwania;
- optymalizuje kod eliminując zbędne operacje;
- testuje programy z różnymi zestawami danych testowych;
- obsługuje pliki tekstowe (odczyt, zapis);
- tworzy czytelny i dobrze zorganizowany kod.

Ocena celująca (6):

Uczeń spełnia wymagania na ocenę bardzo dobrą oraz:

- implementuje zaawansowane struktury danych (listy jednokierunkowe, stosy, kolejki);
- tworzy złożone projekty składające się z wielu modułów;
- implementuje zaawansowane algorytmy;

- stosuje wskaźniki do funkcji;
- efektywnie zarządza pamięcią dynamiczną (alokacja, dealokacja, unikanie wycieków);
- pisze programy obsługujące argumenty wiersza poleceń;
- tworzy własne biblioteki funkcji;
- implementuje obsługę wyjątków i zaawansowaną walidację danych;
- optymalizuje programy pod względem złożoności czasowej i pamięciowej;
- rozwiązuje zadania z olimpiad i konkursów programistycznych;
- samodzielnie realizuje projekty wykraczające poza program nauczania;
- pomaga innym uczniom w debugowaniu i ulepszaniu ich kodu.

UWAGI:

- Przedmiot ma charakter praktyczny - ocenie podlega umiejętność pisania kodu
- Wszystkie programy muszą być samodzielnie napisane przez ucznia
- Ocena uwzględnia nie tylko poprawność, ale też styl, czytelność i efektywność kodu
- Plagiat skutkuje oceną niedostateczną