

**Wymagania edukacyjne na ocenę śródroczną i roczną z przedmiotu „Programowanie strukturalne i obiektowe” dla klasy 3p oraz sposoby sprawdzania osiągnięć edukacyjnych uczniów**

**Rok szkolny 2025/2026**

Program nauczania: PROGRAM NAUCZANIA DLA ZAWODU TECHNIK  
PROGRAMISTA 351406

Przedmiot	Programowanie strukturalne i obiektowe
Klasa	3p
Nauczyciel uczący	Katarzyna Maj

**Wymagania edukacyjne na oceny śródroczne:**

Ocenę **dopuszczającą** otrzymuje uczeń, który:

- tworzy proste algorytmy za pomocą:
  - a) schematów blokowych
  - b) listy kroków
  - c) drzew decyzyjnych
  - d) pseudokodu
- opisuje rodzaje prostych typów danych
- identyfikuje zmienne typu prostego i potrafi określić jej rodzaj
  - a) numeryczny
  - b) stałoprzecinkowy
  - c) zmiennoprzecinkowy
  - d) logiczny
  - e) znakowy i łańcuchowy
- charakteryzuje etapy kompilacji i interpretacji kodu

Ocenę **dostateczną** otrzymuje uczeń, który :

- projektuje złożone algorytmy za pomocą różnych metod (schemat blokowy, lista kroków, drzew decyzyjnych, pseudokodu)
- podaje przykłady wyrażeń wykorzystujących priorytety operatorów
- omawia strukturę tablic jednowymiarowych,
- omawia strukturę tablic dwuwymiarowych,

- omawia strukturę tablic rekordów
- podaje definicję funkcji
- opisuje prototyp funkcji

Ocenę **dobrą** otrzymuje uczeń, który:

- zapisuje złożone algorytmy w języku programowania
- identyfikuje tablice dynamiczne
- identyfikuje tablice asocjacyjne
- identyfikuje typ wskaźnikowy
- przesyła tablicę do funkcji za pomocą wskaźnika
- objaśnia pojęcie rekurencji
- definiuje funkcje rekurencyjne
- wyjaśnia standardowe zastosowania funkcji rekurencyjnych w rozwiązaniu i analizie typowych problemów algorytmicznych

Ocenę **bardzo dobrą** otrzymuje uczeń, który:

- charakteryzuje algorytmy sortowania:
  - a) przez wstawianie
  - b) bąbelkowe
  - c) wymianę/wybór
  - d) szybkie
  - e) scalanie
- wyjaśnia algorytmy szyfrowania, tekstowe

Ocenę **celującą** otrzymuje uczeń, który:

- implementuje poznane algorytmy sortowania w języku programowania
- w pełni opanował materiał
- w sposób kreatywny rozwiązuje zadania o wysokim stopniu trudności
- bierze udział i odnosi sukcesy w konkursach, olimpiadach informatycznych dotyczącej tematyki programowania

## **Wymagania edukacyjne na oceny roczne:**

Ocenę **dopuszczającą** otrzymuje uczeń, który:

- tworzy proste algorytmy za pomocą:
  - a) schematów blokowych
  - b) listy kroków
  - c) drzew decyzyjnych
  - d) pseudokodu
- opisuje rodzaje prostych typów danych
- identyfikuje zmienne typu prostego i potrafi określić jej rodzaj
  - a) numeryczny
  - b) stałoprzecinkowy
  - c) zmiennoprzecinkowy
  - d) logiczny
  - e) znakowy i łańcuchowy
- charakteryzuje etapy kompilacji i interpretacji kodu
- podaje proste operacje na zmiennych: wejścia i wyjścia, arytmetyczne oraz logiczne
- zna operatory arytmetyczne, przypisania, porównania, logiczne, operatory do obsługi łańcuchów, bitowe
- zna składnię instrukcji warunkowej „if”
- zna składnię instrukcję wyboru „switch”
- zna składnię pętli „for”, „while” oraz „do .. while”

Ocenę **dostateczną** otrzymuje uczeń, który :

- projektuje złożone algorytmy za pomocą różnych metod (schemat blokowy, lista kroków, drzew decyzyjnych, pseudokodu)
- podaje przykłady wyrażeń wykorzystujących priorytety operatorów
- omawia strukturę tablic jednowymiarowych,
- omawia strukturę tablic dwuwymiarowych,
- omawia strukturę tablic rekordów
- podaje definicję funkcji
- opisuje prototyp funkcji
- definiuje funkcje jednoargumentowe,
- definiuje funkcje wieloargumentowe,
- zapisuje program w postaci zestawu funkcji,
- objaśnia sposoby przekazywania parametrów do funkcji przez wartość i referencję,
- zapisuje proste algorytmy w języku programowania

Ocenę **dobrą** otrzymuje uczeń, który:

- zapisuje złożone algorytmy w języku programowania
- identyfikuje tablice dynamiczne
- identyfikuje tablice asocjacyjne
- identyfikuje typ wskaźnikowy
- przesyła tablicę do funkcji za pomocą wskaźnika
- objaśnia pojęcie rekurencji
- definiuje funkcje rekurencyjne
- wyjaśnia standardowe zastosowania funkcji rekurencyjnych w rozwiązaniu i analizie typowych problemów algorytmicznych
- wyjaśnia rolę programowania zorientowanego obiektowo
- wyjaśnia pojęcie klasy jako nowego typu zdefiniowanego przez użytkownika
- wyjaśnia pojęcie obiektu jako prototypu nowej klasy
- implementuje instancję klasy w języku programowania
- stosuje obiektowe podejście do rozwiązywania problemów

Ocenę **bardzo dobrą** otrzymuje uczeń, który:

- charakteryzuje algorytmy sortowania:
  - a) przez wstawianie
  - b) bąbelkowe
  - c) wymianę/wybór
  - d) szybkie
  - e) scalanie
- wyjaśnia algorytmy szyfrowania, tekstowe
- charakteryzuje algorytmy rekurencyjne, heurystyczne
- określa złożoność obliczeniową algorytmów
- ocenia efektywność różnych algorytmów sortowania
- omawia algorytmy wyszukiwania dla różnych typów danych
- definiuje klasy jako reprezentację obiektu ze świata rzeczywistego
- definiuje obiekty na rzecz danej klasy
- wyjaśnia składniki statyczne klasy
- definiuje składniki statyczne klasy

Ocenę **celującą** otrzymuje uczeń, który:

- implementuje poznane algorytmy sortowania w języku programowania
- wykorzystuje poznane zagadnienia programowania obiektowego w kontekście ich praktycznego wykorzystania (implementacja w języku programowania zorientowanym obiektowo),
- w pełni opanował materiał
- w sposób kreatywny rozwiązuje zadania o wysokim stopniu trudności
- bierze udział i odnosi sukcesy w konkursach, olimpiadach informatycznych dotyczącej tematyki programowania

Dla uczniów z opiniami z poradni psychologiczno – pedagogicznej wymagania edukacyjne zostaną dostosowane indywidualnie