

Wymagania edukacyjne na ocenę śródroczną i roczną z przedmiotu „Programowanie strukturalne i obiektowe” dla klasy 4p 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	4pT
Nauczyciel uczący	Katarzyna Maj

Wymagania edukacyjne na oceny śródroczne:

Ocenę **dopuszczającą** otrzymuje uczeń, który:

- Zna wybrane biblioteki językaC++, C#
- Analizuje błędy w kodzie za pomocą debuggera
- Implementuje algorytmy w programie
- Charakteryzuje pojęcie biblioteki
- Projektuje program o strukturze modułowej
- Określa sposoby wymiany danych z programem (operacje wejścia–wyjścia)
- Określa klasy reprezentujące zbiory obiektów w danym zagadnieniu
- Potrafi wyjaśnić proste typy danych

Ocenę **dostateczną** otrzymuje uczeń, który:

- Określa klasy reprezentujące zbiory obiektów w danym zagadnieniu
- projektuje program o strukturze modułowej
- zna zastosowanie złożonych typów danych
- deklaruje zmienne różnych typów danych
- zna operacje na złożonych typach danych
- wykorzystuje priorytety operatorów do właściwego budowania programów
- potrafi zaprojektować proste klasy

Ocenę **dobrą** otrzymuje uczeń, który:

- deklaruje własne typy danych
- zna zastosowanie wybranych bibliotek języków programowania C++, C#, Java: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami
- potrafi wyjaśnić obiektowe podejście do rozwiązywania wybranego zagadnienia
- potrafi zaprojektować schemat aplikacji z zastosowaniem metod hermetyzacji
- potrafi zaprojektować schemat aplikacji z zastosowaniem metod dziedziczenia

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

- rozumie zastosowanie i wykorzystanie metod oraz pól dla odpowiednich klas w hierarchii dziedziczenia
- potrafi wyjaśnić definicję klasy pochodnej
- projektuje szablony(wzorce) klas

Ocenę **celującą** otrzymuje uczeń, który:

- projektuje obsługę wyjątków
- 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:

- zna wybrane biblioteki języka C++, C#
- analizuje błędy w kodzie za pomocą debuggera
- implementuje algorytmy w programie
- charakteryzuje pojęcie biblioteki
- projektuje program o strukturze modułowej
- określa sposoby wymiany danych z programem (operacje wejścia–wyjścia)
- określa klasy reprezentujące zbiory obiektów w danym zagadnieniu
- potrafi wyjaśnić proste typy danych
- wykonuje proste operacje na zmiennych: wejścia i wyjścia, arytmetyczne oraz logiczne
- zna zastosowanie operatorów arytmetycznych, przypisania, porównania, logicznych, operatorów do obsługi łańcuchów, bitowych
- zna składnię oraz konstrukcję instrukcji warunkowej „if..else”
- zna składnię oraz konstrukcję instrukcji wyboru „switch”
- zna składnię i konstrukcję pętli

Ocenę **dostateczną** otrzymuje uczeń, który:

- określa klasy reprezentujące zbiory obiektów w danym zagadnieniu
- projektuje program o strukturze modułowej
- zna zastosowanie złożonych typów danych
- deklaruje zmienne różnych typów danych
- zna operacje na złożonych typach danych
- wykorzystuje priorytety operatorów do właściwego budowania programów
- potrafi zaprojektować proste klasy
- dobiera składowe klasy w zależności od jej zastosowania
- określa składniki statyczne klasy
- definiuje obiekty na rzecz danej klasy
- rozumie funkcję i zastosowanie konstruktora na etapie tworzenia nowego obiektu na rzecz klasy
- zna i potrafi zastosować instrukcje inicjujące konstruktor danej klasy

Ocenę **dobrą** otrzymuje uczeń, który:

- deklaruje własne typy danych
- zna zastosowanie wybranych bibliotek języków programowania C++, C#, Java: biblioteka standardowa, biblioteka z funkcjami matematycznymi, biblioteka z podstawowymi algorytmami
- potrafi wyjaśnić obiektowe podejście do rozwiązywania wybranego zagadnienia
- potrafi zaprojektować schemat aplikacji z zastosowaniem metod hermetyzacji
- potrafi zaprojektować schemat aplikacji z zastosowaniem metod dziedziczenia
- potrafi zaprojektować schemat aplikacji z zastosowaniem metod polimorfizmu
- rozumie pojęcie dotyczące zakresu widoczności pól klasy i potrafi zdefiniować kwalifikatory dostępu
- określa zakres widoczności metod klasy
- określa kwalifikatory dostępu
- zna i rozumie zastosowanie i wykorzystanie klas dziedziczonych oraz zaprzyjaźnionych
- zna i rozumie zastosowanie funkcji zaprzyjaźnionych z klasą
- wyjaśnia hierarchię dziedziczenia klas w programie

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

- rozumie zastosowanie i wykorzystanie metod oraz pól dla odpowiednich klas w hierarchii dziedziczenia
- potrafi wyjaśnić definicję klasy pochodnej
- projektuje szablony(wzorce) klas
- projektuje szablony klas dla obsługi typów liczbowych
- projektuje klasy bazowe i pochodne
- stosuje metody wirtualne
- projektuje klasy abstrakcyjne

Ocenę **celującą** otrzymuje uczeń, który:

- projektuje obsługę wyjątków
- rozumie zastosowanie i potrafi zastosować szkielet obsługi wyjątków z instrukcjami „try” i „catch”
- 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