

## Wymagania edukacyjne z informatyki – poziom rozszerzony

---

### Klasa I

#### Ocena dopuszczająca (2)

- Definiuje podstawowe pojęcia: algorytm, program, zmienna.
- Potrafi zapisać liczby w systemie binarnym.
- Píše bardzo proste programy (np. obliczanie sumy dwóch liczb).

#### Ocena dostateczna (3)

- Tworzy krótkie algorytmy w postaci schematów blokowych.
- Wykonuje podstawowe operacje na liczbach w systemie binarnym i dziesiętnym.
- Stosuje pętle i instrukcje warunkowe w prostych programach.

#### Ocena dobra (4)

- Píše programy operujące na zmiennych i tablicach.
- Stosuje podstawowe operacje logiczne (koniunkcja, alternatywa).
- Tworzy algorytmy obliczające wartości w różnych systemach liczbowych.

#### Ocena bardzo dobra (5)

- Rozwiązuje zadania złożone przy użyciu pętli zagnieżdżonych.
- Korzysta z podstawowych funkcji języka programowania.
- Optymalizuje kod pod względem czytelności i wydajności.

#### Ocena celująca (6)

- Implementuje złożone algorytmy obliczeniowe.
- Wykazuje inicjatywę w tworzeniu własnych projektów programistycznych.
- Wykracza poza program nauczania, np. stosując rekurencję.

---

### Klasa II

#### Ocena dopuszczająca (2)

- **Podstawy programowania:**
  - Zrozumienie pojęć: program, instrukcja, algorytm.
  - Rozróżnianie różnych typów danych (np. liczby, tekst).
  - Tworzenie prostych programów składających się z kilku instrukcji.
- **Komputery:**
  - Opisanie podstawowych elementów komputera i ich funkcji.
  - Wyjaśnienie pojęcia systemu operacyjnego.
  - Bezpieczne korzystanie z komputera i Internetu.

### Ocena dostateczna (3)

- **Algorytmy:**
  - Tworzenie prostych algorytmów i przedstawianie ich w postaci schematów blokowych.
  - Rozwiązywanie prostych problemów algorytmicznych (np. znajdowanie największej liczby w zbiorze).
- **Programowanie:**
  - Zastosowanie zmiennych, operatorów arytmetycznych i logicznych w programach.
  - Pisanie prostych programów wykorzystujących instrukcje warunkowe (jeśli, to, inaczej).
  - Tworzenie prostych pętli (np. for, while).

### Ocena dobra (4)

- **Struktury danych:**
  - Korzystanie z tablic do przechowywania danych.
  - Wykonywanie podstawowych operacji na tablicach (np. sortowanie, wyszukiwanie).
- **Algorytmy:**
  - Projektowanie i implementacja algorytmów rozwiązujących bardziej złożone problemy (np. sortowanie bąbelkowe, wyszukiwanie liniowe).
- **Programowanie:**
  - Zastosowanie funkcji w programach.
  - Modułowe programowanie (dzielenie programu na mniejsze, samodzielne części).

### Ocena bardzo dobra (5)

- **Algorytmy i struktury danych:**
  - Zastosowanie bardziej zaawansowanych struktur danych (np. listy, stosy, kolejki).
  - Projektowanie i implementacja efektywnych algorytmów (np. sortowanie przez scalanie, wyszukiwanie binarne).
- **Programowanie:**
  - Optymalizacja kodu pod kątem szybkości działania i czytelności.
  - Korzystanie z narzędzi do debugowania programów.
- **Projektowanie:**
  - Projektowanie prostych interfejsów użytkownika.

### Ocena celująca (6)

- **Zaawansowane tematy:**
    - Rekurencja, programowanie obiektowe.
    - Algorytmy grafowe, algorytmy przeszukiwania
-

## **Klasa III**

### **Ocena dopuszczająca (2)**

- Zna podstawowe algorytmy sortowania i wyszukiwania.
- Tworzy schematy blokowe prostych problemów.
- Wczytuje dane z plików tekstowych.

### **Ocena dostateczna (3)**

- Implementuje algorytmy sortowania (np. bąbelkowe) w języku programowania.
- Stosuje proste struktury danych, takie jak tablice i listy.
- Wyjaśnia sposób działania wybranych algorytmów (np. wyszukiwania binarnego).

### **Ocena dobra (4)**

- Pisze programy wykorzystujące algorytmy dynamiczne.
- Optymalizuje algorytmy sortowania pod względem złożoności obliczeniowej.
- Tworzy projekty wykorzystujące współpracę z bazami danych.

### **Ocena bardzo dobra (5)**

- Rozwiązuje problemy związane z przetwarzaniem danych w dużej skali.
- Implementuje złożone algorytmy grafowe (np. DFS, BFS).
- Tworzy aplikacje z interfejsem użytkownika.

### **Ocena celująca (6)**

- Wdraża zaawansowane technologie, np. algorytmy sztucznej inteligencji.
  - Bierze udział w konkursach informatycznych, osiągając sukcesy.
  - Tworzy systemy rozproszone lub chmurowe.
- 

## **Klasa IV**

### **Ocena dopuszczająca (2)**

- Wykorzystuje języki programowania do tworzenia prostych aplikacji.
- Wykonuje operacje na relacyjnych bazach danych.

### **Ocena dostateczna (3)**

- Projektuje proste aplikacje internetowe w oparciu o HTML i CSS.
- Stosuje język SQL do obsługi baz danych.

### **Ocena dobra (4)**

- Tworzy aplikacje z backendem, korzystając z PHP lub Node.js.
- Analizuje dane z wykorzystaniem podstawowych bibliotek programistycznych.

### **Ocena bardzo dobra (5)**

- Tworzy złożone aplikacje webowe i mobilne.
- Optymalizuje systemy bazodanowe pod kątem wydajności.

### **Ocena celująca (6)**

- Tworzy projekty integrujące różne technologie (webowe, mobilne, bazodanowe).
  - Podejmuje inicjatywę w realizacji projektów zespołowych jako lider.
- 

## **Klasa V (jeśli obowiązuje)**

### **Ocena dopuszczająca (2)**

- Zna podstawy przetwarzania danych w chmurze.
- Implementuje proste modele analityczne.

### **Ocena dostateczna (3)**

- Tworzy aplikacje z wykorzystaniem bibliotek analizy danych.
- Stosuje modele predykcyjne na prostych zbiorach danych.

### **Ocena dobra (4)**

- Optymalizuje aplikacje pod kątem efektywności obliczeniowej.
- Wdraża proste rozwiązania chmurowe w projektach.

### **Ocena bardzo dobra (5)**

- Integruje aplikacje analityczne z systemami zewnętrznymi.
- Stosuje zaawansowane modele sztucznej inteligencji.

### **Ocena celująca (6)**

- Wykorzystuje technologie AI/ML w rozwiązaniach biznesowych.
- Osiąga sukcesy w projektach o zasięgu ogólnopolskim lub międzynarodowym.