

# Przedmiotowy system oceniania (propozycja)

Przedmiotowy system oceniania (PSO) to podstawowe zasady wewnątrzszkolnego oceniania uczniów z danego przedmiotu. Powinien być zgodny z podstawą programową oraz wewnątrzszkolnym systemem oceniania (WSO). Prezentowany materiał może posłużyć nauczycielom jako pomoc w opracowaniu własnych systemów zgodnych z wytycznymi obowiązującymi w szkole.

**Ocenę celującą** otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych,
- bierze udział w konkursach i olimpiadach informatycznych i zajmuje punktowane miejsca,
- pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych,
- optymalizuje programy, szacuje ich efektywność,
- wykorzystuje poznane algorytmy do rozwiązywania problemów nieomawianych na lekcjach, np. sprawdzanie spójności grafu,
- projektuje rozbudowane relacyjne bazy danych, zarządza nimi, wykorzystując zaawansowane narzędzia oraz klauzule języka SQL,
- tworzy rozwinięte responsywne aplikacje internetowe wymagające dużego nakładu pracy i znajomości nowoczesnych technologii,
- tworzy infografiki, korzystając z zaawansowanych narzędzi graficznych,
- programuje roboty wg własnych projektów, wykazując się przy tym kreatywnością, oraz wykorzystuje aplikacje mobilne do sterowania nimi,
- tworzy podcasty i publikacje wideo wymagające znajomości zaawansowanych narzędzi i dużego nakładu pracy,
- w projektach zespołowych przyjmuje rolę lidera, opracowuje złożone problemy, posługując się aplikacjami w stopniu zaawansowanym,
- w dyskusjach panelowych przyjmuje funkcję eksperta.

**Ocenę bardzo dobrą** otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- charakteryzuje sytuacje algorytmiczne, proponuje sposoby ich rozwiązania,
- pisze programy o podwyższonym stopniu trudności: rozwiązuje zadania oznaczone trzema gwiazdkami w podręczniku,
- optymalizuje rozwiązania,
- stosuje zaawansowane funkcje środowiska i języka programowania,
- dobiera struktury danych i metody do rodzaju problemu,
- szacuje złożoność algorytmów,
- implementuje algorytmy grafowe – BFS, DFS, algorytm Dijkstry,
- w reprezentacji liczb rzeczywistych w komputerze stosuje reprezentację stało- lub zmiennoprzecinkową zgodnie ze specyfikacją algorytmu, minimalizując błędy w obliczeniach,
- stosuje schemat Hornera do szybkiego podnoszenia do potęgi,

- implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą Newtona–Raphsona, obliczania pola obszaru zamkniętego metodą prostokątów i metodą trapezów, znajdowania przybliżenia liczby pi oraz symulacja ruchów Browna metodą Monte Carlo,
- implementuje algorytmy badające własności geometryczne,
- implementuje w języku JavaScript algorytmy generujące fraktale danego stopnia,
- stosuje metodę IFS do tworzenia fraktali w arkuszu kalkulacyjnym,
- stosuje funkcję haszującą oraz algorytm Karpa–Rabina w programach wyszukujących wzorców w tekście,
- pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA,
- stosuje programowanie obiektowe, definiując własne klasy, obiekty, atrybuty i metody, deklaruje konstruktory w klasach, wyjaśnia, na czym polega polimorfizm i czym są metody wirtualne,
- projektuje zaawansowane relacyjne bazy danych, zarządza nimi, tworzy tabele pomocnicze, formularze, kwerendy i raporty,
- formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych,
- tworzy aplikacje internetowe z przejrzystym interfejsem użytkownika korzystające z sieciowej bazy danych, testuje je i wprowadza poprawki,
- testuje i optymalizuje programy sterujące robotem,
- tworzy interesujące podcasty i publikacje wideo, dba o właściwy format plików, stosuje kompresję, stosuje zasady pracy z kamerą i mikrofonem,
- tworzy infografiki dostosowane do odbiorców, wykazując się dużymi umiejętnościami korzystania z narzędzi graficznych,
- aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, opracowuje prezentacje multimedialne, filmy przedstawiające efekty wspólnej pracy,
- w dyskusji panelowej przyjmuje rolę moderatora.

**Ocenę dobrą** otrzymuje uczeń, który spełnia kryteria ocen niższych, a ponadto:

- pisze programy o różnym stopniu trudności, szacuje ich efektywność,
- dobiera typy danych do rozwiązania problemu,
- do przeglądania grafu stosuje algorytm przeszukiwania w głąb (DFS) oraz algorytm przeszukiwania grafu wszerek (BFS),
- omawia algorytm Dijkstry,
- znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji,
- świadomie używa typów `float` i `double` w zadaniach,
- stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną,
- stosuje metodę Monte Carlo w obliczeniach przybliżonych,
- w algorytmach badających własności geometryczne wykorzystuje macierz oraz regułę Sarrusa do obliczania wyznacznika macierzy,

- omawia algorytm Karpa–Rabina do wyszukiwania wzorca w tekście z zastosowaniem funkcji haszującej,
- pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA,
- w programowaniu obiektowym stosuje hierarchię klas, wyjaśnia, na czym polega hermetyzacja danych i jakie jest zastosowanie operatora zasięgu,
- projektuje i tworzy proste bazy danych,
- przy tworzeniu aplikacji internetowej projektuje i tworzy interfejs użytkownika, zapewnia komunikację aplikacji z bazą danych,
- programuje roboty według określonych wytycznych, np. zatrzymanie przed przeszkodą,
- tworzy podcasty i publikacje wideo, stosując elementy przyciągające uwagę użytkowników, montuje materiały, wykorzystując specjalistyczne oprogramowanie,
- projektuje infografiki, umiejętnie stosując tekst i obraz, wykazuje się przy tym znajomością doboru barw i funkcji koloru, zwraca uwagę na dostosowanie treści do odbiorców,
- uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej oraz dyskusji panelowej.

**Ocenę dostateczną** otrzymuje uczeń, który spełnia kryteria oceny dopuszczającej, a ponadto:

- wyróżnia operacje, które można wykonywać na dynamicznych strukturach danych (stosie, kolejce, liście, typie `vector`),
- omawia zastosowanie dynamicznych struktur danych na różnych przykładach,
- zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w postaci odwrotnej notacji polskiej,
- oblicza wartość wyrażenia arytmetycznego zapisanego w odwrotnej notacji polskiej,
- omawia algorytmy znajdowania wyjścia z labiryntu z wykorzystaniem iteracji i rekurencji,
- symuluje problem Flawiusza,
- sortuje dane leksykograficznie,
- stosuje typ `vector` do reprezentacji grafu w postaci list sąsiedztwa,
- omawia algorytm przeszukiwania grafu w głąb (DFS),
- omawia algorytm przeszukiwania grafu wszcz (BFS),
- wyjaśnia, do czego służy algorytm Dijkstry,
- wyjaśnia różnicę między przekazywaniem parametrów do funkcji przez wartość i przez referencję,
- wykorzystuje pliki tekstowe do wczytywania danych i zapisywania wyników,
- omawia algorytm znajdujący rozwinięcie binarne nieskracalnego ułamka właściwego,
- zapisuje liczby w postaci znormalizowanej,
- definiuje liczby pojedynczej precyzji i liczby podwójnej precyzji,
- wykonuje działania na liczbach zmiennoprzecinkowych,
- wskazuje różnice między algorytmem stabilnym a algorytmem niestabilnym,
- znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym,
- implementuje algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera,
- stosuje w algorytmach numerycznych metody: bisekcji, Newtona–Raphsona, trapezów, prostokątów,

- omawia algorytmy badające własności geometryczne – położenie punktu względem prostej, przecinania się odcinków, przynależności punktu do figury,
- podaje przykłady fraktali (zbiór Cantora, drzewo binarne, dywan Sierpińskiego, płatek Kocha), wyjaśnia sposób tworzenia tych fraktali,
- implementuje algorytm naiwny wyszukiwania wzorca w tekście,
- wyjaśnia metodę haszowania,
- wyjaśnia, jak generuje się klucze publiczny i prywatny oraz szyfruje i deszyfruje informacje w algorytmie RSA,
- wyjaśnia, na czym polegają metoda zstępująca i metoda wstępująca,
- w programowaniu obiektowym definiuje własne klasy, korzystając ze specyfikatorów dostępu,
- wyszukuje informacje w bazach danych, tworzy formularze, kwerendy i raporty,
- wykorzystuje język SQL do tworzenia i usuwania baz danych, dodawania tabel do baz danych, usuwania tabel z baz, dodawania rekordów do tabel, importowania danych do tabel, edycji rekordów,
- tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy, używając języka SQL,
- formułuje zapytania zwracające określone dane, sortuje wyniki zapytań,
- wyjaśnia, na czym polega praca nad aplikacją internetową, instaluje i konfiguruje niezbędne oprogramowanie, przygotowuje bazę danych na potrzeby projektu,
- tworzy proste programy sterujące robotem, korzysta przy tym ze środowiska dostępnego dla fizycznego modelu robota lub z symulatora,
- opracowuje treści internetowe, korzystając z narzędzi graficznych i multimedialnych, dbając o identyfikację wizualną,
- projektuje proste poprawne infografiki zawierające uporządkowane informacje,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.

**Ocenę dopuszczającą** otrzymuje uczeń, który:

- pisze programy o niewielkim stopniu trudności,
- wyjaśnia, co to jest notacja infiksowa, notacja prefiksowa, odwrotna notacja polska, drzewo wyrażenia algebraicznego,
- definiuje pojęcie dynamicznej struktury danych,
- definiuje dynamiczne struktury danych takie jak: stos, kolejka, lista, **vector**,
- wymienia rodzaje list,
- wyjaśnia, na czym polega sortowanie leksykograficzne,
- definiuje graf, wymienia elementy i rodzaje grafów, wymienia sposoby reprezentacji grafu (macierz sąsiedztwa, lista sąsiedztwa),
- omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze,
- wymienia rodzaje błędów w obliczeniach komputerowych, rozróżnia błąd względny i bezwzględny,
- znajduje wartość wielomianu algorytmem naiwnym,
- wie, na czym polegają podstawowe metody obliczeń przybliżonych,

- zna proste algorytmy badające własności geometryczne (np. położenie punktu względem prostej),
- wyjaśnia, co to jest fraktal, wskazuje przykłady struktur fraktalnych występujących w przyrodzie,
- wyszukuje wzorzec w tekście algorytmem naiwnym,
- rozumie działanie funkcji haszującej,
- wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje pojęcia klucz publiczny i klucz prywatny,
- wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym),
- definiuje programowanie strukturalne,
- definiuje programowanie obiektowe i podstawowe pojęcia z nim związane,
- zna podstawowe pojęcia dotyczące relacyjnych baz danych,
- wie, co to jest język SQL, zna podstawowe klauzule tego języka,
- zna zasady tworzenia zapytań do bazy z wykorzystaniem języka SQL,
- wyróżnia etapy pracy nad aplikacją internetową, rozróżnia technologie back-end i front-end,
- definiuje pojęcie robota, omawia funkcje wybranych robotów i ich budowę,
- rozróżnia pojęcia webcast, webinarium, screencast i podcast,
- definiuje pojęcie grafiki informacyjnej, wymienia przykłady grafiki narracyjnej i wizualizacji danych,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania o niewielkim stopniu trudności.

**Ocenę niedostateczną** otrzymuje uczeń, który:

- nie wyjaśnia podstawowych pojęć – notacja infiksowa, notacja prefiksowa, odwrotna notacja polska, dynamiczna struktura danych, graf, stało- i zmiennoprzecinkowa reprezentacja liczb rzeczywistych, błąd zaokrąglenia, błąd przybliżenia, błąd reprezentacji, błąd względny, błąd bezwzględny, metody obliczeń przybliżonych, fraktal, metoda haszowania, kryptografia symetryczna, kryptografia asymetryczna, klucz publiczny, klucz prywatny, programowanie strukturalne, programowanie obiektowe, klasa, obiekt, atrybut, metoda, relacyjna baza danych, język SQL, system zarządzania bazą danych,
- nie zna podstawowych algorytmów – obliczania wartości wielomianu (algorytm naiwny), znajdowania miejsc zerowych funkcji, obliczania pól obszarów zamkniętych metodami przybliżonymi, badania własności geometrycznych, tworzenia przykładowych fraktali, wyszukiwania wzorca w tekście (algorytm naiwny), szyfrowania z kluczem publicznym (algorytm RSA),
- nie umie utworzyć relacji między tabelami w bazie, korzystać z formularzy i raportów, wykorzystywać danych pochodzących z kwerend,
- nie potrafi konstruować prostych zapytań w języku SQL,
- nie wie, na czym polega tworzenie aplikacji internetowych,
- nie zna pojęcia robota ani jego budowy, nie potrafi utworzyć prostego programu sterującego robotem, dysponując narzędziami, w tym online,

- nie umie opracować prostych treści internetowych, nie posługuje się narzędziami graficznymi i multimedialnymi do wzbogacania treści,
- nie bierze czynnego udziału w lekcjach, nie odrabia prac domowych,
- nie rozwiązuje najprostszych zadań,
- nie uczestniczy w projektach zespołowych.