Wymagania edukacyjne z informatyki

klasa 4

poziom rozszerzony

Przyjmujemy, że uczeń spełnia wymagania na ocenę wyższą, jeśli spełnia

jednocześnie wymagania na ocenę niższą oraz dodatkowe wymagania.

Proponujemy:

→Wymagania na ocenę dopuszczającą.

→Wymagania na ocenę dostateczną zawierają wymagania na ocenę dopuszczającą.

→Wymagania na ocenę dobrą zawierają wymagania na ocenę dostateczną i dopuszczającą.

→Wymagania na ocenę bardzo dobrą zawierają wymagania na ocenę dobrą, dostateczną i dopuszczającą.

→Wymagania na ocenę celującą zawierają wymagania na oceną bardzo dobrą, dobrą, dostateczną i dopuszczającą.

Rozdział 1. Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych

Ocenę celującą otrzymuje uczeń, który:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, implementuje w języku C++ algorytm Euklidesa, stosując iterację i rekurencję,
- implementuje w języku C++ algorytm wyszukiwania binarnego w wersji rekurencyjnej,
- pisze programy sortujące dane różnego typu w plikach tekstowych (liczby, napisy, pary),
- stosuje zaawansowane algorytmy i struktury danych do wyszukiwania spójnych podciągów,
- stosuje zaawansowane algorytmy wyszukiwania, np. najlepszego wyboru (trwałych par), stosując rekurencję,

- pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- stosuje w programach algorytmy sortowania inne niż omawiane na lekcjach (np. heapsort),
- bierze udział w olimpiadach i konkursach, zajmując punktowane miejsca,
- w projektach zespołowych przyjmuje rolę lidera.
- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych,
- 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,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- wyszukuje spójne podciągi w plikach tekstowych, stosując optymalne algorytmy (w tym programowanie dynamiczne), wyjaśnia ich działanie,
- pisze programy obliczające liczbę operacji przenoszenia krążków w problemie wież Hanoi, stosując iterację i rekurencję,
- do implementacji rozszerzonego algorytmu Euklidesa stosuje zarówno iterację, jak i rekurencję,
- stosuje metody zachłanną i dynamiczną w problemach kasjera, harmonogramu wykorzystania sali, pakowania plecaka i wyszukiwania drogi, wskazuje wady i zalety obu metod, szacuje złożoność czasową,
- aktywnie uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, prezentuje efekty wspólnej pracy.
- 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,
- implementuje algorytmy badające własności geometryczne,

Ocenę dobrą otrzymuje uczeń, który:

- stosuje algorytm wyszukiwania binarnego i oszacowuje jego złożoność czasową,
- pisze programy sortujące metodami prostymi z zastosowaniem funkcji typu void,
- stosuje algorytmy sortowania szybkiego i przez scalanie,

- stosuje metodę zachłanną w programach – problem kasjera, harmonogram wykorzystania sali, wyszukiwanie drogi, pakowanie plecaka,
- uczestniczy w realizacji projektu zespołowego na wszystkich jego etapach, bierze czynny udział w tworzeniu dokumentacji projektowej.
- 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 wszerz (BFS),
- omawia algorytm Dijkstry,

Ocenę dostateczną otrzymuje uczeń, który:

- formułuje algorytm wydawania reszty minimalną liczbą monet, harmonogramu wykorzystania sali, pakowania plecaka, znajdowania drogi metodami zachłanną i dynamiczną,
- uczestniczy w realizacji projektu zespołowego, wykonując powierzone mu zadania.
- 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 wszerz (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,
- 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,
- wyjaśnia, na czym polegają metoda zstępująca i metoda wstępująca,

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),

Rozdział 2. Algorytmy numeryczne

Ocenę celującą otrzymuje uczeń, który:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów

Ocenę bardzo dobrą otrzymuje uczeń, który:

- 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,
- pisze algorytmy stabilne dla problemów generujących przekroczenia błędów zaokrąglenia,

Ocenę dobrą otrzymuje uczeń, który:

- 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,

- oblicza błąd względny i bezwzględny dla reprezentacji rzeczywistej liczby i podstawowych działań arytmetycznych,
- znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i liczby podwójnej precyzji,

Ocenę dostateczną otrzymuje uczeń, który:

- 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,

Ocenę dopuszczającą otrzymuje uczeń, który:

- 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,
- omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze,
- wymienia rodzaje błędów w obliczeniach komputerowych, rozróżnia błąd względny i bezwzględny,

Rozdział 3. Zaawansowane algorytmy i techniki programistyczne

Ocenę celującą otrzymuje uczeń, który:

- 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:

- w programowaniu obiektowym stosuje hierarchię klas, wyjaśnia, na czym polega hermetyzacja danych i jakie jest zastosowanie operatora zasięgu,
- 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.
- stosuje programowanie obiektowe, definiując własne klasy, obiekty, atrybuty i metody, deklaruje konstruktory w klasach, wyjaśnia, na czym polega polimorfizm i czym są metody wirtualne,
- 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:

- 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 dostateczną otrzymuje uczeń, który:

- w programowaniu obiektowym definiuje własne klasy, korzystając ze specyfikatorów dostępu,

Ocenę dopuszczającą otrzymuje uczeń, który:

- definiuje programowanie strukturalne,
- definiuje programowanie obiektowe i podstawowe pojęcia z nim związane,
- zna podstawowe pojęcia dotyczące relacyjnych baz danych,
- 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.
- wie, na czym polegają podstawowe metody obliczeń przybliżonych,

Rozdział 4. Rozwiązywanie różnych problemów z wykorzystaniem komputera

Ocenę celującą otrzymuje uczeń, który:

- charakteryzuje skomplikowane sytuacje algorytmiczne, proponuje optymalne rozwiązanie sytuacji problemowej z zastosowaniem złożonych struktur danych i biblioteki STL języka C++,
- pisze programy o wysokim stopniu trudności: z olimpiad przedmiotowych, konkursów informatycznych, implementuje w języku C++,

Ocenę bardzo dobrą otrzymuje uczeń, który:

- pisze program znajdujący punkty kratowe zawarte w okręgu w czasie liniowym,
- pisze program znajdujący punkty kratowe zawarte w figurze,
- pisze program obliczający iloczyn wektorowy,
- pisze program badający przecięcie dwóch odcinków,
- pisze program badający wypukłość figury,

Ocenę dobrą otrzymuje uczeń, który:

- pisze program znajdujący punkty kratowe zawarte w odcinku,
- pisze program obliczający punkty przecięcia okręgów
- pisze program sortujący podane punkty względem odległości od początku układu,

Ocenę dostateczną otrzymuje uczeń, który:

- pisz program obliczający pole wielokąta,
- pisze program badający współliniowość trzech punktów,
- pisze program znajdujący punkty kratowe zawarte w okręgu metodą naiwną,

Ocenę dopuszczającą otrzymuje uczeń, który:

- znajduje wartość wielomianu algorytmem naiwnym,
- umie reprezentować wektor na płaszczyźnie,
- zna proste algorytmy badające własności geometryczne (np. położenie punktu względem prostej),
- wyjaśnia, co to jest fraktal, wskazuje przykłady struktur fraktalnych występujących w przyrodzie,

* opracowano na podstawie materiałów wydawnictwa Nowa Era – Informatyka na czasie.