

Wymagania edukacyjne z informatyki w klasie VIII

Arkusz kalkulacyjny

Ocena dopuszczająca	Ocena dostateczna	Ocena dobra	Ocena bardzo dobra	Ocena celująca
<ul style="list-style-type: none"> • omawia zastosowanie oraz budowę arkusza kalkulacyjnego • określa adres komórki • wprowadza dane różnego rodzaju do komórek arkusza kalkulacyjnego • formatuje zawartość komórek (wyrównanie tekstu oraz wygląd czcionki) • rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym • wstawia wykres do arkusza kalkulacyjnego • korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków 	<ul style="list-style-type: none"> • określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego • dodaje i usuwa wiersze oraz kolumny w tabeli • stosuje w arkuszu podstawowe funkcje: (SUMA, SREDNIA), wpisuje je ręcznie oraz korzysta z kreatora • omawia i modyfikuje poszczególne elementy wykresu • zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie 	<ul style="list-style-type: none"> • tworzy proste formuły obliczeniowe • wyjaśnia, czym jest adres względny • wykorzystuje funkcję JEŻELI do tworzenia algorytmów z warunkami w arkuszu kalkulacyjnym • ustawia format danych komórki odpowiadający jej zawartości • w formułach stosuje adresowanie względne, bezwzględne i mieszane • dobiera odpowiedni wykres do rodzaju danych • sortuje oraz filtruje dane w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> • korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje • kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne • stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych • tworzy wykres dla więcej niż jednej serii danych • tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym • stosuje filtry niestandardowe 	<ul style="list-style-type: none"> • samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe • stosuje zaawansowane funkcje arkusza w tabelach tworzonych na własne potrzeby • tworzy rozbudowane wykresy dla wielu serii danych • przygotowuje rozbudowane arkusze kalkulacyjne • korzysta z arkusza kalkulacyjnego do analizowania doświadczeń z innych przedmiotów

Programowanie w języku C++

Ocena dopuszczająca	Ocena dostateczna	Ocena dobra	Ocena bardzo dobra	Ocena celująca
<ul style="list-style-type: none"> definiuje pojęcia: algorytm, program, programowanie podaje kilka sposobów przedstawienia algorytmu tłumaczy, do czego używa się zmiennych w programach pisze proste programy w języku C++ z wykorzystaniem zmiennych wyjaśnia działanie operatora modulo wyjaśnia algorytm badania podzielności liczb wyjaśnia potrzebę wyszukiwania informacji w zbiorze sprawdza działanie programów wyszukujących element w zbiorze wyjaśnia potrzebę porządkowania danych sprawdza działanie programu sortującego dla różnych danych 	<ul style="list-style-type: none"> wymienia różne sposoby przedstawienia algorytmu: opis słowny, lista kroków poprawnie formułuje problem do rozwiązania stosuje odpowiednie polecenie języka C++, aby wyświetlić tekst na ekranie omawia różnice pomiędzy kodem źródłowym a kodem wynikowym tłumaczy, czym jest środowisko programistyczne wykonuje obliczenia w języku C++ omawia działanie operatorów arytmetycznych stosuje tablice w języku C++ oraz operatory logiczne zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych wykorzystuje w programach instrukcję iteracyjną while zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym implementuje grę w zgadywanie liczby zapisuje w wybranej formie algorytm porządkowania metodą przez wybieranie 	<ul style="list-style-type: none"> wymienia przykładowe środowiska programistyczne wyjaśnia, czym jest specyfikacja problemu opisuje etapy rozwiązywania problemów opisuje etapy powstawania programu komputerowego zapisuje proste polecenia języka C++ wykorzystuje instrukcję warunkową if oraz if else w programach wykorzystuje iterację w konstruowanych algorytmach wykorzystuje w programach instrukcję iteracyjną for definiuje funkcje w języku C++ i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości omawia algorytm Euklidesa i zapisuje go w wybranej postaci wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym omawia funkcje zastosowane w 	<ul style="list-style-type: none"> buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów pisze proste programy w języku C++ konstruuje złożone sytuacje warunkowe (wiele warunków) w algorytmach pisze programy zawierające instrukcje warunkowe, pętle oraz funkcje wyjaśnia, jakie błędy zwraca interpreter czyta kod źródłowy i opisuje jego działanie wyjaśnia różnice między instrukcją iteracyjną while a pętlą for pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze implementuje algorytm porządkowania metodą przez wybieranie wprowadza modyfikacje w implementacji algorytmu porządkowania przez 	<ul style="list-style-type: none"> pisze programy w języku C++ do rozwiązywanie zadań matematycznych zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności tworzy program składający się z kilku funkcji wywoływanych w programie głównym pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania samodzielnie modyfikuje i optymalizuje program sortujący metodą przez wybieranie

	<ul style="list-style-type: none">• omawia implementację algorytmu sortowania przez wybieranie• stosuje pętle zagnieżdżone i wyjaśnia, jak działają	kodzie źródłowym algorytmu sortowania przez wybieranie	wybieranie	
--	--	--	------------	--

Projekty

Ocena dopuszczająca	Ocena dostateczna	Ocena dobra	Ocena bardzo dobra	Ocena celująca
<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • testuje grę na różnych etapach • współpracuje w grupie podczas pracy nad projektem bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej • wprowadza dane do zaprojektowanych tabel • bierze udział w pracach nad wypracowaniem koncepcji gry • współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy • współpracuje w grupie podczas pracy nad projektem • programuje wybrane funkcje i elementy gry • opracowuje opis gry 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki • współpracuje w grupie podczas pracy nad projektem • implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy zestawienia zawierające zaawansowane formuły, wykresy oraz elementy graficzne • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera • rozbudowuje grę o nowe elementy • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera

Ocenę celującą otrzymuje uczeń, który opanował wszystkie treści z podstawy programowej.