

Wymagania edukacyjne na poszczególne oceny z przedmiotu informatyka w klasie 8 szkoły podstawowej – podręcznik „Lubię to!” (język programowania C++), wydawnictwo Nowa Era

Uwzględnia zapisy podstawy programowej z 2017 r. oraz zmiany z 2024 r., wynikające z uszczuplonej podstawy programowej.
Nazwa programu nauczania: Program nauczania informatyki w szkole podstawowej „Lubię to!”

ocena dopuszczająca Uczeń:	ocena dostateczna Uczeń:	ocena dobra Uczeń:	ocena bardzo dobra Uczeń:	ocena celująca Uczeń:
DZIAŁ 1. Arkusz kalkulacyjny				
– 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)	– określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego – dodaje i usuwa wiersze oraz kolumny w tabeli	– tworzy proste formuły obliczeniowe – wyjaśnia, czym jest adres względny	– kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne	– samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe
– rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym	– stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora	– 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	– korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje – stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych	– stosuje zaawansowane funkcje arkusza w tabelach tworzonych na własne potrzeby
– wstawia wykres do	– omawia i modyfikuje	– dobiera odpowiedni	– tworzy wykres dla więcej	– tworzy rozbudowane

arkusza kalkulacyjnego	poszczególne elementy wykresu	wykres do rodzaju danych	niżej jednej serii danych	wykresy dla wielu serii danych
– korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków	– zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie	– sortuje oraz filtruje dane w arkuszu kalkulacyjnym	– tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym – stosuje filtry niestandardowe	– przygotowuje rozbudowane arkusze kalkulacyjne korzysta z arkusza kalkulacyjnego do analizowania doświadczeń z innych przedmiotów
DZIAŁ 2. Programowanie w języku C++				
– definiuje pojęcia: algorytm, program, programowanie – podaje kilka sposobów przedstawienia algorytmu	– 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	– wymienia przykładowe środowiska programistyczne – wyjaśnia, czym jest specyfikacja problemu – opisuje etapy rozwiązywania problemów – opisuje etapy powstawania programu komputerowego – zapisuje proste polecenia języka C++	– pisze proste programy w języku C++	– zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności
– tłumaczy, do czego używa się zmiennych w programach – pisze proste programy w języku C++ z wykorzystaniem zmiennych	– wykonuje obliczenia w języku C++ – omawia działanie operatorów arytmetycznych – stosuje tablice w języku C++ oraz operatory	– wykorzystuje instrukcję warunkową if oraz if else w programach – wykorzystuje iterację w konstruowanych algorytmach – wykorzystuje	– buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów – konstruuje złożone sytuacje warunkowe	– pisze programy w języku C++ do rozwiązywania zadań matematycznych – tworzy program składający się z kilku funkcji wywoływanych w programie głównym

	logiczne	w programach instrukcję iteracyjną for – definiuje funkcje w języku C++ i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości	(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 działanie operatora modulo – wyjaśnia algorytm badania podzielności liczb	– zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych – wykorzystuje w programach instrukcję iteracyjną while	– omawia algorytm Euklidesa i zapisuje go w wybranej postaci – wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci	– wyjaśnia różnice między instrukcją iteracyjną while a pętlą for – pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby	– pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby
– wyjaśnia potrzebę wyszukiwania informacji w zbiorze – sprawdza działanie programów wyszukujących element w zbiorze	– zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym – implementuje grę w zgadywanie liczby	– implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym	– samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze	– samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
– wyjaśnia potrzebę porządkowania danych – sprawdza działanie programu sortującego dla różnych danych	– zapisuje w wybranej formie algorytm porządkowania metodą przez wybieranie – omawia implementację algorytmu sortowania przez wybieranie – stosuje pętle	– omawia funkcje zastosowane w kodzie źródłowym algorytmu sortowania przez wybieranie	– implementuje algorytm porządkowania metodą przez wybieranie – wprowadza modyfikacje w implementacji algorytmu porządkowania przez wybieranie	– samodzielnie modyfikuje i optymalizuje program sortujący metodą przez wybieranie

	zagnieżdżone i wyjaśnia, jak działają			
DZIAŁ 3. Projekty				
– bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności	– bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej – wprowadza dane do zaprojektowanych tabel	– przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy – współpracuje w grupie podczas pracy nad projektem	– bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki – współpracuje w grupie podczas pracy nad projektem	– 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
– 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 pracach nad wypracowaniem koncepcji gry – współpracuje w grupie podczas pracy nad projektem	– programuje wybrane funkcje i elementy gry – opracowuje opis gry	– implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń	– rozbudowuje grę o nowe elementy – współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera

Ocena niedostateczna (1):

Uczeń nie opanował podstawowych wiadomości i umiejętności określonych w wymaganiach edukacyjnych z informatyki, nie potrafi samodzielnie wykonać prostych zadań praktycznych, nawet przy pomocy nauczyciela, oraz nie osiąga minimum programowego.