

PRZEDMIOTOWE ZASADY OCENIANIA Z INFORMATYKI W KLASIE ÓSMEJ
z uwzględnieniem lekcji programowania w języku C++

WYMAGANIA EDUKACYJNE NIEZBĘDNE DO UZYSKANIA POSZCZEGÓLNYCH OCEN BIEŻĄCYCH, ŚRÓDROCZNYCH I ROCZNYCH Z INFORMATYKI W KLASIE ÓSMEJ						
Tytuł w podręczniku	Numer i temat lekcji	Wymagania konieczne (ocena dopuszczająca) Uczeń:	Wymagania podstawowe (ocena dostateczna) Uczeń:	Wymagania rozszerzające (ocena dobra) Uczeń:	Wymagania dopełniające (ocena bardzo dobra) Uczeń:	Wymagania wykraczające (ocena celująca) Uczeń:
Informacja: Aby uczeń otrzymał ocenę wyższą musi opanować szczegółowe wymagania na oceny niższe.						
DZIAŁ 1. Arkusz kalkulacyjny						
Wymagania podstawy programowej: II.3c, II.4, II.5						
1.1. Formuły i adresowanie względne w arkuszu kalkulacyjnym	1. i 2. Formuły i adresowanie względne w arkuszu kalkulacyjnym	<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) 	<ul style="list-style-type: none"> ● określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego ● dodaje i usuwa wiersze oraz kolumny w tabeli 	<ul style="list-style-type: none"> ● tworzy proste formuły obliczeniowe ● wyjaśnia, czym jest adres względny 	<ul style="list-style-type: none"> ● kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne 	<ul style="list-style-type: none"> ● samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe
1.2. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	3. i 4. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	<ul style="list-style-type: none"> ● rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym 	<ul style="list-style-type: none"> ● stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora 	<ul style="list-style-type: none"> ● wykorzystuje funkcję JEŻELI do tworzenia algorytmów z warunkami w arkuszu kalkulacyjnym ● ustawia format danych komórki odpowiadający jej zawartości 	<ul style="list-style-type: none"> ● korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje ● stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych 	<ul style="list-style-type: none"> ● stosuje zaawansowane funkcje arkusza w tabelach tworzonych na własne potrzeby

				<ul style="list-style-type: none"> • w formułach stosuje adresowanie względne, bezwzględne i mieszane 		
1.3. Przedstawianie danych na wykresie	5. i 6. Przedstawianie danych na wykresie	<ul style="list-style-type: none"> • wstawia wykres do arkusza kalkulacyjnego 	<ul style="list-style-type: none"> • omawia i modyfikuje poszczególne elementy wykresu 	<ul style="list-style-type: none"> • dobiera odpowiedni wykres do rodzaju danych 	<ul style="list-style-type: none"> • tworzy wykres dla więcej niż jednej serii danych 	<ul style="list-style-type: none"> • tworzy rozbudowane wykresy dla wielu serii danych
1.4. Zastosowania arkusza kalkulacyjnego	7. 8. Zastosowania arkusza kalkulacyjnego	<ul style="list-style-type: none"> • korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków 	<ul style="list-style-type: none"> • zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie 	<ul style="list-style-type: none"> • sortuje oraz filtruje dane w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> • tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym • stosuje filtry niestandardowe 	<ul style="list-style-type: none"> • przygotowuje rozbudowane arkusze kalkulacyjne korzysta z arkusza kalkulacyjnego do analizowania doświadczeń z innych przedmiotów
DZIAŁ 3. Programowanie w języku C++						
Wymagania podstawy programowej: I.1, I.2a, I.2b, II.1, II.2						
3.1. Wprowadzenie do programowania w języku C++	9., 10. i 11. Wprowadzenie do programowania w języku C++	<ul style="list-style-type: none"> • definiuje pojęcia: algorytm, program, programowanie • podaje kilka sposobów przedstawienia algorytmu 	<ul style="list-style-type: none"> • wymienia różne sposoby przedstawienia algorytmu: opis słowny, schemat blokowy, lista kroków • poprawnie formułuje problem do rozwiązania • wyjaśnia różnice między interaktywnym a skryptowym trybem pracy • stosuje odpowiednie polecenie języka 	<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++ 	<ul style="list-style-type: none"> • pisze proste programy w trybie skryptowym języka C++ 	<ul style="list-style-type: none"> • zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności

			<p>C++, aby wyświetlić tekst na ekranie</p> <ul style="list-style-type: none"> • omawia różnice pomiędzy kodem źródłowym a kodem wynikowym • tłumaczy, czym jest środowisko programistyczne 			
3.2. Piszemy programy w języku C++	12., 13. i 14. Piszemy programy w języku C++	<ul style="list-style-type: none"> • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka C++ z wykorzystaniem zmiennych 	<ul style="list-style-type: none"> • wykonuje obliczenia w języku C++ • omawia działanie operatorów arytmetycznych • stosuje tablice w języku C++ oraz operatory logiczne 	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów • 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 	<ul style="list-style-type: none"> • pisze programy w języku C++ do rozwiązywanie zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych w programie głównym
3.3. Algorytmy na liczbach naturalnych	15., 16. i 17. Algorytmy na liczbach naturalnych	<ul style="list-style-type: none"> • wyjaśnia działanie operatora modulo • wyjaśnia algorytm badania podzielności liczb 	<ul style="list-style-type: none"> • zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych • wykorzystuje w programach 	<ul style="list-style-type: none"> • omawia algorytm Euklidesa w wersji z odejmowaniem i z dzieleniem – zapisuje go w wybranej postaci 	<ul style="list-style-type: none"> • wyjaśnia różnice między instrukcją iteracyjną while a pętlą for • pisze programy obliczające NWD, stosując algorytm 	<ul style="list-style-type: none"> • pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby

			instrukcję iteracyjną while	<ul style="list-style-type: none"> • wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci 	<p>Euklidesa, oraz wypisujące cyfry danej liczby</p> <ul style="list-style-type: none"> • wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem 	
3.4. Algorytmy wyszukiwania	18. i 19. Algorytmy wyszukiwania	<ul style="list-style-type: none"> • wyjaśnia potrzebę wyszukiwania informacji w zbiorze • określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym • sprawdza działanie programów wyszukujących element w zbiorze 	<ul style="list-style-type: none"> • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elementu największego i najmniejszego • zapisuje algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia • implementuje grę w zgadywanie liczby 	<ul style="list-style-type: none"> • implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia • implementuje algorytm wyszukiwania największej wartości w zbiorze 	<ul style="list-style-type: none"> • samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze metodą połowienia, w tym elementu największego i najmniejszego • implementuje algorytm wyszukiwania elementu w zbiorze metodą połowienia 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
3.5. Algorytmy porządkowania	20. i 21. Algorytmy porządkowania	<ul style="list-style-type: none"> • wyjaśnia potrzebę porządkowania danych • sprawdza działanie programu sortującego dla różnych danych 	<ul style="list-style-type: none"> • zapisuje w wybranej formie algorytm porządkowania metodami przez wybieranie oraz przez zliczanie • omawia implementację algorytmu sortowania przez wybieranie 	<ul style="list-style-type: none"> • omawia implementację algorytmu sortowania przez zliczanie • omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez 	<ul style="list-style-type: none"> • implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie • wprowadza modyfikacje w implementacji algorytmów porządkowania 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie

			<ul style="list-style-type: none"> • stosuje pętle zagnieżdżone i wyjaśnia, jak działają • 	wybieranie oraz przez zliczanie	przez wybieranie oraz przez zliczanie	
DZIAŁ 4. Projekty						
Wymagania podstawy programowej: II.3c, II.4, IV.1						
4.1. Dokumentacja szkolnej imprezy sportowej	22. i 23. Dokumentacja szkolnej imprezy sportowej	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy • współpracuje w grupie podczas pracy nad projektem 	<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 	<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy zestawienia zawierające zaawansowane formuły, wykresy oraz elementy graficzne
4.2. Sterowanie obiektem na ekranie	24., 25. i 26. Sterowanie obiektem na ekranie	<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 	<ul style="list-style-type: none"> • bierze udział w pracach nad wypracowaniem koncepcji gry • współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> • programuje wybrane funkcje i elementy gry • opracowuje opis gry 	<ul style="list-style-type: none"> • implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń 	<ul style="list-style-type: none"> • rozbudowuje grę o nowe elementy
4.3. Historia i rozwój informatyki	27., 28. i 29. Historia i rozwój informatyki	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności – znalezienie informacji w Internecie, umieszczenie ich w chmurze 	<ul style="list-style-type: none"> • współpracuje z innymi podczas pracy nad projektem • analizuje zebrane dane • tworzy projekt prezentacji multimedialnej 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania • tworzy prezentację wg projektu zaakceptowanego przez zespół 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu • analizuje i weryfikuje pod względem merytorycznym i technicznym 	<ul style="list-style-type: none"> • wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne

					przygotowaną prezentację	
4.4. Informatyka w moim przyszłym życiu	30. Informatyka w moim przyszłym życiu	<ul style="list-style-type: none"> ● aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności ● bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych 	<ul style="list-style-type: none"> ● gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych 	<ul style="list-style-type: none"> ● aktywnie uczestniczy w pracach zespołu ● projektuje tabele do zapisywania informacji o zawodach ● weryfikuje i formatuje przygotowane dokumenty tekstowe 	<ul style="list-style-type: none"> ● aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość 	

Każdy uczeń oceniany jest zgodnie z indywidualnymi dostosowaniami edukacyjnymi zawartymi w opinii/ orzeczeniu z poradni Psychologiczno Pedagogicznej. Uczeń, który nie opanował wymagań na ocenę dopuszczającą (konieczną) otrzymuje ocenę niedostateczną. Opracowano na podstawie Podstawy Programowej nauczania informatyki na rok 2024.

Metodyka pracy na lekcjach informatyki w klasie 8 szkoły podstawowej.

Klasę 8 uczniowie rozpoczynają od wykonywania obliczeń w arkuszu kalkulacyjnym. Poznają jego zaawansowane funkcje. Następnie przechodzą do algorytmiki i programowania. Dokładnie poznają wybrane algorytmy i je stosują. Poznają jeden z dwóch tekstowych języków programowania i tworzą w nich pierwsze programy. Na zakończenie klasy 8 przypominają sobie etapy rozwoju informatyki i technologii. Zastanawiają się nad możliwą ścieżką rozwoju zawodowego, związaną z informatyką.

W związku z uszczupleniem przez MEN podstawy programowej, w rozkładzie materiału zmniejszyła się liczba godzin na realizację obowiązkowych zagadnień. Uzyskane w ten sposób dodatkowe godziny pozostają do dyspozycji nauczyciela w trakcie roku szkolnego. Zgodnie z założeniami MEN: Ograniczony zakres treści nauczania – wymagań szczegółowych – da nauczycielom i uczniom więcej czasu na spokojniejszą i bardziej dogłębną realizację programów nauczania.