

Wymagania edukacyjne z informatyki
dla klasy drugiej czteroletniego liceum
(zakres podstawowy)

Wymagania programowe na poszczególne oceny przygotowane na podstawie treści zawartych w podstawie programowej oraz w podręczniku „Informatyka na czasie 2” wydawnictwo Nowa Era, zakres podstawowy. Numer dopuszczenia MEN 990/2019

Temat lekcji	Zagadnienia	Wymagania edukacyjne na poszczególne oceny				
		Ocena dopuszczająca Uczeń:	Ocena dostateczna Uczeń:	Ocena dobra Uczeń:	Ocena bardzo dobra Uczeń:	Ocena celująca Uczeń:
Wymagania edukacyjne niezbędne do otrzymania przez ucznia śródrocznej oceny klasyfikacyjnej						
Rozdział I. Arkusz kalkulacyjny i bazy danych						
Jak pobierać dane do arkusza kalkulacyjnego	<ul style="list-style-type: none"> • Formuły i funkcje • Typy adresów w arkuszu kalkulacyjnym • Importowanie danych do arkusza 	<ul style="list-style-type: none"> — wyjaśnia, dlaczego warto stosować narzędzia wymiany danych — wymienia podstawowe zastosowania arkusza kalkulacyjnego 	<ul style="list-style-type: none"> — wyjaśnia podstawowe pojęcia związane z arkuszem kalkulacyjnym: skoroszyt, arkusz, adres komórki, formuła, funkcja, zakres adresów — kopiuje formuły — stosuje funkcje: SUMA, ŚREDNIA, MAX, MIN, DŁ, JEŻELI 	<ul style="list-style-type: none"> — modyfikuje dane w arkuszu — wykorzystuje adresy komórek w formułach obliczeniowych — korzysta z wbudowanych funkcji arkusza kalkulacyjnego — kopiuje dane z komórek i wkleja je na różne sposoby, również między arkuszami 	<ul style="list-style-type: none"> — stosuje różne sposoby zaznaczania zakresów komórek — przedstawia dane w postaci wykresów — pobierana dane z różnych źródeł i przetwarza je — wyjaśnia różnice między formułami i funkcjami 	<ul style="list-style-type: none"> — buduje złożone formuły pozwalające wykonywać obliczenia, rozwiązujące określone problemy — dobiera typ wykresu do rodzaju danych — interpretuje otrzymane wyniki zgodnie z ustalonymi założeniami

<p>Wyciągamy wiedzę z danych</p>	<ul style="list-style-type: none"> • Tabele przestawne • Wykresy przestawne 	<ul style="list-style-type: none"> — omawia różnicę między filtrowaniem i sortowaniem danych — filtruje i sortuje dane 	<ul style="list-style-type: none"> — tworzy tabele i stosuje w nich sortowanie i filtrowanie danych — stosuje formuły arkusza kalkulacyjnego do losowego generowania zbiorów danych 	<ul style="list-style-type: none"> — stosuje funkcję INDEKS do zwracania wartości określonych komórek 	<ul style="list-style-type: none"> — opisuje możliwości tabel przestawnych — generuje zestawy danych za pomocą narzędzi online 	<ul style="list-style-type: none"> — modyfikuje style tabel przestawnych — buduje tabele przestawne dla dużych zbiorów danych — tworzy fragmentatory i korzysta z osi czasu — interpretuje wyniki tabel i wykresów przestawnych
<p>Korespondencja seryjna</p>	<ul style="list-style-type: none"> • Tworzenie korespondencji seryjnej • Filtrowanie adresów • Personalizacja listów 	<ul style="list-style-type: none"> — tworzy dokument główny korespondencji seryjnej 	<ul style="list-style-type: none"> — umieszcza pola korespondencji seryjnej w tworzonych dokumentach — tworzy bazę adresatów 	<ul style="list-style-type: none"> — stosuje reguły warunkowe do personalizacji listów seryjnych — poprawnie scala dokumenty seryjne 	<ul style="list-style-type: none"> — zarządza danymi adresatów korespondencji seryjnej w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> — tworzy zestawy dokumentów seryjnych (listy, etykiety, koperty) — drukuje listy seryjne — wysyła korespondencję seryjną za pomocą poczty elektronicznej

Relacyjne bazy danych	<ul style="list-style-type: none"> • Pojęcia związane z bazami danych • Formularze • Raporty • Kwerendy 	<ul style="list-style-type: none"> — wyjaśnia podstawowe pojęcia związane z bazami danych: tabela, atrybut, rekord, pole, klucz główny, klucz obcy, relacja — wymienia różne zastosowania baz danych 	<ul style="list-style-type: none"> — projektuje nieduże bazy danych — zarządza danymi w bazie danych w programie MS Access 	<ul style="list-style-type: none"> — modyfikuje dane zawarte w bazie danych — tworzy tabele i definiuje relacje między nimi — tworzy raporty z użyciem kreatora 	<ul style="list-style-type: none"> — tworzy kwerendy, formularze oraz raporty w programie MS Access 	<ul style="list-style-type: none"> — dostosowuje raport do aktualnych potrzeb — drukuje i eksportuje raporty do pliku
Współdziałanie aplikacji – projekt zespołowy	<ul style="list-style-type: none"> • Opracowywanie projektów • Praca w grupie • Przetwarzanie danych w arkuszu kalkulacyjnym 	<ul style="list-style-type: none"> — współpracuje w grupie, korzystając z narzędzi online — korzysta z programów graficznych podczas pracy nad zadaniem projektowym 	<ul style="list-style-type: none"> — stosuje funkcje arkusza kalkulacyjnego do przetwarzania danych — korzysta z zasobów internetowych, wyszukując potrzebne informacje 	<ul style="list-style-type: none"> — testuje rozwiązania wypracowane w grupie — stosuje zasady netykiety i korzysta z niej w komunikacji zdalnej 	<ul style="list-style-type: none"> — prezentuje efekty pracy grupowej na forum klasy 	<ul style="list-style-type: none"> — przyjmuje rolę lidera odpowiedzialnego za zespół i projekt

Wymagania edukacyjne niezbędne do otrzymania przez ucznia rocznej oceny klasyfikacyjnej (obejmują wymagania edukacyjne niezbędne do otrzymania przez ucznia śródrocznej oceny klasyfikacyjnej).

Rozdział II Algorytmika i programowanie w języku C++

<p>Od problemu do programu</p>	<ul style="list-style-type: none"> • Algorytmy • Podstawy języka C++ 	<ul style="list-style-type: none"> — wyjaśnia pojęcie algorytmu — podaje przykłady algorytmów w życiu codziennym — kompiluje zapisany kod źródłowy 	<ul style="list-style-type: none"> — wymienia cechy poprawnego algorytmu — wyjaśnia na przykładzie pojęcie specyfikacji problemu — tworzy algorytm wyznaczania pierwiastka kwadratowego 	<ul style="list-style-type: none"> — zapisuje algorytm Herona w postaci listy kroków — wyjaśnia pojęcia związane z algorytmiką i programowaniem: schemat blokowy, lista kroków, kod źródłowy, kod wynikowy, kompilator, interpreter, słowa kluczowe, funkcje, plik wykonywalny — zapisuje algorytm w postaci kodu źródłowego 	<ul style="list-style-type: none"> — znajduje i poprawia błędy w kodzie źródłowym programu — wyjaśnia pojęcie zmiennej i typu zmiennej — wymienia zasady tworzenia kodu źródłowego w wybranym języku programowania — stosuje podstawowe konstrukcje wybranego języka programowania: instrukcje wejścia i wyjścia, operatory 	<ul style="list-style-type: none"> — tworzy samodzielnie programy, wykorzystując poznane instrukcje wybranego języka programowania — stosuje w swoich programach zagnieżdżone instrukcje warunkowe — pisze programy rozwiązujące zadania matematyczne i fizyczne oraz problemy z napisami
--------------------------------	--	---	--	---	---	--

					<p>arytmetyczne i logiczne oraz instrukcję warunkową</p> <ul style="list-style-type: none"> — tworzy program sprawdzający warunek trójkąta 	
Wyszukujemy i sumujemy	<ul style="list-style-type: none"> • Pętla while • Pętla for 	<ul style="list-style-type: none"> — wyjaśnia, na czym polega iteracyjne rozwiązywanie problemu — stosuje w swoich programach podstawowe rodzaje pętli: while i for 	<ul style="list-style-type: none"> — opisuje zasady użycia pętli w programach — analizuje w tabeli działanie algorytmu krok po kroku 	<ul style="list-style-type: none"> — opisuje, jak komputer porównuje dwie wartości — tworzy program wyszukiwania największej liczby z danego ciągu liczb 	<ul style="list-style-type: none"> — stosuje różne rodzaje pętli, dostosowując wybór do rozwiązywanego problemu — stosuje w kodzie operatory inkrementacji i dekrementacji 	<ul style="list-style-type: none"> — używa w algorytmach rozwiązania z wartownikiem, czyli specjalną wartością wskazującą koniec ciągu wczytywanych do programu wartości — analizuje kod źródłowy i poprawia ewentualne błędy
Binarny system liczbowy	<ul style="list-style-type: none"> • Liczby dziesiętne i dwójkowe • Systemy reprezentacji liczb 	<ul style="list-style-type: none"> — wyjaśnia pojęcia związane z systemami 	<ul style="list-style-type: none"> — zapisuje liczby w binarnym systemie liczbowym oraz w systemie 	<ul style="list-style-type: none"> — omawia algorytm konwersji liczb między 	<ul style="list-style-type: none"> — zapisuje algorytm konwersji między systemami 	<ul style="list-style-type: none"> — tworzy algorytm konwersji między

	<ul style="list-style-type: none"> • Algorytm zamiany reprezentacji liczb 	<p>liczbowymi: system pozycyjny, podstawa systemu liczbowego</p>	<p>szesnastkowym</p>	<p>systemami dziesiętnym i binarnym</p>	<p>liczbowymi w postaci programu komputerowego</p> <ul style="list-style-type: none"> — stosuje operację dzielenia całkowitego w rozwiązywaniu problemów 	<p>różnymi systemami liczbowymi</p> <ul style="list-style-type: none"> — programuje algorytmy konwersji między różnymi systemami liczbowymi, stosując strukturę tablicy lub listy
<p>Czy ta liczba jest pierwsza?</p>	<ul style="list-style-type: none"> • Liczby złożone i pierwsze • Podzielność liczb 	<ul style="list-style-type: none"> — definiuje liczby złożone i liczby pierwsze — podaje przykłady użycia liczb pierwszych 	<ul style="list-style-type: none"> — omawia algorytmy sprawdzające podzielność liczb 	<ul style="list-style-type: none"> — bada podzielność wybranych liczb, programując poznane algorytmy w wybranym języku — grupuje instrukcje w funkcje i wyjaśnia cel stosowania funkcji 	<ul style="list-style-type: none"> — tworzy samodzielnie programy dla poznanych algorytmów 	<ul style="list-style-type: none"> — wyjaśnia pojęcia liczb doskonałych, bliźniaczych, zaprzyjaźnionych — pisze programy wykorzystujące poznane rodzaje liczb pierwszych — wyjaśnia praktyczne znaczenie liczb pierwszych w informatyce

<p>Algorytm Euklidesa i działania na ułamkach</p>	<ul style="list-style-type: none"> • Największy wspólny dzielnik • Działania na ułamkach zwykłych 	<ul style="list-style-type: none"> — wyjaśnia pojęcia: NWD, NWW 	<ul style="list-style-type: none"> — podaje przykłady zastosowania algorytmu Euklidesa — zapisuje algorytm Euklidesa w postaci listy kroków 	<ul style="list-style-type: none"> — tworzy program pozwalający na dodawanie ułamków — stosuje odpowiednie konstrukcje wybranego języka programowania do implementacji omawianych zagadnień (w tym: funkcję, która nie zwraca wartości) 	<ul style="list-style-type: none"> — tworzy programy realizujące działania na ułamkach 	<ul style="list-style-type: none"> — opisuje algorytm Euklidesa i tworzy realizujący go program w wybranym języku programowania — opisuje różnicę w sprawności dwóch wersji algorytmu Euklidesa: z odejmowaniem i z dzieleniem — poznaje inne zastosowania algorytmu Euklidesa, wykorzystując informacje zawarte w internecie lub innych źródłach
---	---	--	---	---	---	--

UWAGI:

1. Ocenę wyższą otrzymuje uczeń spełniający łącznie wymagania edukacyjne określone dla ocen niższych np. ocenę dobrą otrzymuje uczeń spełniający wymagania edukacyjne na ocenę dopuszczającą, dostateczną oraz dobrą.
2. Ocenę niedostateczną otrzymuje uczeń, który nie spełnia wymagań na poszczególne pozytywne oceny.
3. W przypadku nie zrealizowania tematów lekcji (zagadnień) w I okresie będą one realizowane po klasyfikacji śródrocznej. W tym przypadku obowiązują również wymagania edukacyjne dla tych tematów (zagadnień).