

WYMAGANIA EDUKACYJNE Z INFORMATYKI W KLASIE 8

UMIĘTNOŚCI KONIECZNE, PODSTAWOWE Uczeń:	UMIĘTNOŚCI ROZSZERZAJĄCE, DOPEŁNIAJĄCE, WYKRACZAJĄCE Uczeń:
Dział 1. Arkusz kalkulacyjny	
<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) • 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 • kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne • samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe
<ul style="list-style-type: none"> • rozumie różnicę między adresowaniem względnym, bezwzględnym i mieszanym • stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora • wstawia wykres do arkusza kalkulacyjnego • omawia i modyfikuje poszczególne elementy wykresu 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków • zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie 	<ul style="list-style-type: none"> • dobiera odpowiedni wykres do rodzaju danych • tworzy wykres dla więcej niż jednej serii danych • tworzy rozbudowane wykresy dla wielu serii danych • 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 Python	
<ul style="list-style-type: none"> • definiuje pojęcia: algorytm, program, programowanie • podaje kilka sposobów przedstawienia algorytmu • wymienia różne sposoby przedstawienia algorytmu: opis słowny, lista kroków • poprawnie formułuje problem do rozwiązania 	<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 Python

<ul style="list-style-type: none"> • wyjaśnia różnice między interaktywnym a skryptowym trybem pracy • stosuje odpowiednie polecenie języka Python, aby wyświetlić tekst na ekranie • omawia różnice pomiędzy kodem źródłowym a kodem wynikowym • tłumaczy, czym jest środowisko programistyczne 	<ul style="list-style-type: none"> • pisze proste programy w trybie skryptowym języka Python • zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności
<ul style="list-style-type: none"> • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka Python z wykorzystaniem zmiennych • wykonuje obliczenia w języku Python • omawia działanie operatorów arytmetycznych • stosuje listy w języku Python 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 Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości • 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 • pisze programy w języku Python do rozwiązywania zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych w programie głównym
<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • wyjaśnia potrzebę wyszukiwania informacji w zbiorze • sprawdza działanie programów wyszukujących element w zbiorze • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elementu największego i najmniejszego 	<ul style="list-style-type: none"> • implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze • samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
<ul style="list-style-type: none"> • 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 zagnieżdżone i wyjaśnia, jak działają 	<ul style="list-style-type: none"> • 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 programy sortujące metodą przez wybieranie
Dział 3. Projekty	
<ul style="list-style-type: none"> • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej 	<ul style="list-style-type: none"> • 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

<ul style="list-style-type: none"> • wprowadza dane do zaprojektowanych tabel 	<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
<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 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 • 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