

Wymagania edukacyjne

INFORMATYKA

KLASA VIII

Tytuł w podręczniku	Numer i temat lekcji	(ocena dopuszczająca) Uczeń:	(ocena dostateczna) Uczeń:	(ocena dobra) Uczeń:	(ocena bardzo dobra) Uczeń:	(ocena celująca) Uczeń:
I PÓŁROCZE						
DZIAŁ 1. Arkusz kalkulacyjny						
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	3. i 4. Funkcje oraz adresowanie	<ul style="list-style-type: none"> rozumie różnice między 	<ul style="list-style-type: none"> stosuje w arkuszu podstawowe funkcje: 	<ul style="list-style-type: none"> wykorzystuje funkcję JEŻELI do 	<ul style="list-style-type: none"> korzysta z biblioteki funkcji, 	<ul style="list-style-type: none"> stosuje zaawansowane

bezwzględne i mieszane w arkuszu kalkulacyjnym	bezwzględne i mieszane w arkuszu kalkulacyjnym	adresowaniem względnym, bezwzględnym i mieszanym	(SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora	tworzenia algorytmów z warunkami w arkuszu kalkulacyjnym <ul style="list-style-type: none"> ustawia format danych komórki odpowiadający jej zawartości w formułach stosuje adresowanie względne, bezwzględne i mieszane 	aby wyszukiwać potrzebne funkcje <ul style="list-style-type: none"> stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych 	funkcje arkusza w tabelach tworzonych na własne potrzeby
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 2. Programowanie w języku Python						
2.1. Wprowadzenie do	9., 10. i 11. Wprowadzenie do	<ul style="list-style-type: none"> definiuje pojęcia: algorytm, program, programowanie 	<ul style="list-style-type: none"> wymienia różne sposoby przedstawienia 	<ul style="list-style-type: none"> wymienia przykładowe środowiska programistyczne 	<ul style="list-style-type: none"> pisze proste programy w trybie 	<ul style="list-style-type: none"> zapisuje algorytmy różnymi sposobami oraz pisze programy

programowania w języku Python	programowania w języku Python	<ul style="list-style-type: none"> • podaje kilka sposobów przedstawienia algorytmu 	<p>algorytmu: opis słowny, lista kroków</p> <ul style="list-style-type: none"> • poprawnie formułuje problem do rozwiązania • 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"> • wyjaśnia, czym jest specyfikacja problemu • opisuje etapy rozwiązywania problemów • opisuje etapy powstawania programu komputerowego • zapisuje proste polecenia języka Python 	skryptowym języka Python	o większym stopniu trudności
2.2. Piszemy programy w języku Python	12., 13. i 14. Piszemy programy w języku Python	<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 	<ul style="list-style-type: none"> • 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 <code>if</code> oraz <code>if else</code> w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach instrukcję iteracyjną <code>for</code> 	<ul style="list-style-type: none"> • konstruuje złożone sytuacje warunkowe (wiele warunków) w algorytmach • pisze programy zawierające instrukcje warunkowe, pętle oraz funkcje 	<ul style="list-style-type: none"> • 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"> definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości 	<ul style="list-style-type: none"> wyjaśnia, jakie błędy zwraca interpreter czyta kod źródłowy i opisuje jego działanie 	
2.3. 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 instrukcję iteracyjną <code>while</code> 	<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 	<ul style="list-style-type: none"> wyjaśnia różnice między instrukcją iteracyjną <code>while</code> a pętlą <code>for</code> pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby 	<ul style="list-style-type: none"> pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby
2.4. Algorytmy wyszukiwania	18. i 19. Algorytmy wyszukiwania	<ul style="list-style-type: none"> wyjaśnia potrzebę wyszukiwania informacji w zbiorze 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 	<ul style="list-style-type: none"> implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym 	<ul style="list-style-type: none"> samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze 	<ul style="list-style-type: none"> samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
2.5. Algorytmy porządkowania	20. i 21. Algorytmy porządkowania	<ul style="list-style-type: none"> wyjaśnia potrzebę porządkowania danych 	<ul style="list-style-type: none"> zapisuje w wybranej formie algorytm porządkowania 	<ul style="list-style-type: none"> omawia funkcje zastosowane w kodzie źródłowym algorytmu 	<ul style="list-style-type: none"> implementuje algorytm porządkowania 	<ul style="list-style-type: none"> samodzielnie modyfikuje i optymalizuje programy sortujące

		<ul style="list-style-type: none"> • sprawdza działanie programu sortującego dla różnych danych 	<p>metodą przez wybieranie</p> <ul style="list-style-type: none"> • omawia implementację algorytmu sortowania przez wybieranie • stosuje pętle zagnieżdżone i wyjaśnia, jak działają 	<p>sortowania przez wybieranie</p>	<p>metodą przez wybieranie</p> <ul style="list-style-type: none"> • wprowadza modyfikacje w implementacji algorytmu porządkowania przez wybieranie 	<p>metodą przez wybieranie</p>
II PÓLROCZE						
<ul style="list-style-type: none"> • DZIAŁ 4. Projekty 						
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 • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera

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 • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
-------------------------------------	---	--	--	--	---	--