

Publiczna Szkoła Podstawowa im. Tadeusza Kościuszki w Bładowie

WYMAGANIA EDUKACYJNE

Przedmiot: Informatyka

Klasa 8

Nauczyciel: mgr Krystyna Kaźmierczak

Program nauczania informatyki w szkole podstawowej „Lubię to!” - autor: Grażyna Koba, Nowa Era

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:
-	Zapoznanie z programem nauczania oraz wymaganiami edukacyjnymi. Bezpieczna praca przy komputerze.	Określa, za co może uzyskać daną ocenę; wymienia możliwości poprawy oceny niedostatecznej oraz zasady pracy na zajęciach komputerowych. Wymienia zasady bezpieczeństwa obowiązujące w pracowni komputerowej. Stosuje poznane zasady bezpieczeństwa w pracowni oraz podczas pracy na komputerze i w Internecie.				
-	O usłudze GSuite Jak pracować w Classroom. Dziennik elektroniczny.	- pracuje w Classroom - korzysta z dziennika elektronicznego				
DZIAŁ 1. Arkusz kalkulacyjny						
1.1. Formuły i adresowanie względne w arkuszu kalkulacyjnym	Formuły i adresowanie względne w arkuszu kalkulacyjnym	· omawia zastosowanie oraz budowę arkusza kalkulacyjnego · określa adres komórki · wprowadza dane różnego rodzaju do komórek arkusza kalkulacyjnego · formatuje zawartość komórek (wyrównanie	· 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

		tekstu oraz wygląd czcionki)				
1.2. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	· 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
1.3. Przedstawianie danych na wykresie	Przedstawianie danych na wykresie	· wstawia wykres do arkusza kalkulacyjnego	· omawia i modyfikuje poszczególne elementy wykresu	· dobiera odpowiedni wykres do rodzaju danych	· tworzy wykres dla więcej niż jednej serii danych	· tworzy rozbudowane wykresy dla wielu serii danych
1.4. Zastosowania arkusza kalkulacyjnego	Zastosowania arkusza kalkulacyjnego	· 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 Python						

2.1. Wprowadzenie do programowania w języku Python	Wprowadzenie do programowania w języku Python	· definiuje pojęcia: algorytm, program, programowanie · podaje kilka sposobów przedstawienia algorytmu	· 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 Python, 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 Python ·	· pisze proste programy w trybie skryptowym języka Python	· zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności
--	---	---	---	---	---	---

2.2. Piszemy programy w języku Python	Piszemy programy w języku Python	· 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	· 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> · definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości	· 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	· pisze programy w języku Python do rozwiązywanie zadań matematycznych · tworzy program składający się z kilku funkcji wywoływanych w programie głównym
2.3. Algorytmy na liczbach naturalnych	Algorytmy na liczbach naturalnych	· wyjaśnia działanie operatora modulo · wyjaśnia algorytm badania podzielności liczb	· zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych · wykorzystuje w programach instrukcję iteracyjną <code>while</code>	· omawia algorytm Euklidesa w wersji z odejmowaniem i z dzieleniem – zapisuje go w wybranej postaci · wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci	· 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 · wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem	· pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby

2.4. Algorytmy wyszukiwania	Algorytmy wyszukiwania	· wyjaśnia potrzebę wyszukiwania informacji w zbiorze · określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym · sprawdza działanie programów wyszukujących element w zbiorze	· 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	· implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym · omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia · implementuje algorytm wyszukiwania największej wartości w zbiorze	· 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	· samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
2.5. Algorytmy porządkowania	Algorytmy porządkowania	· wyjaśnia potrzebę porządkowania danych · sprawdza działanie programu sortującego dla różnych danych	· zapisuje w wybranej formie algorytm porządkowania metodami przez wybieranie oraz przez zliczanie · omawia implementację algorytmu sortowania przez wybieranie · stosuje pętle zagnieżdżone i wyjaśnia, jak działają	· omawia implementację algorytmu sortowania przez zliczanie · omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez wybieranie oraz przez zliczanie	· implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie · wprowadza modyfikacje w implementacji algorytmów porządkowania przez wybieranie oraz przez zliczanie	· samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie
· DZIAŁ 4. Projekty						

4.1. Dokumentacja szkolnej imprezy sportowej	Dokumentacja szkolnej imprezy sportowej	· 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
4.2. Sterowanie obiektem na ekranie	Sterowanie obiektem na ekranie	· 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
4.3. Historia i rozwój informatyki	Historia i rozwój informatyki	· aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności – znalezienie informacji w internecie, umieszczenie ich w chmurze	· współpracuje z innymi podczas pracy nad projektem · analizuje zebrane dane · tworzy projekt prezentacji multimedialnej	· aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania · tworzy prezentację wg projektu zaakceptowanego przez zespół	· aktywnie uczestniczy w pracach zespołu · analizuje i weryfikuje pod względem merytorycznym i technicznym przygotowaną prezentację	· współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera · wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne

4.4. Informatyka w moim przyszłym życiu	Informatyka w moim przyszłym życiu	· aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności · bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych	· gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych	· aktywnie uczestniczy w pracach zespołu · projektuje tabele do zapisywania informacji o zawodach · weryfikuje i formatuje przygotowane dokumenty tekstowe	· aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość	· aktywnie uczestniczy w pracach zespołu, przyjmuje rolę lidera · podczas dyskusji przyjmuje funkcję moderatora
---	------------------------------------	--	--	--	---	--