**WYMAGANIA EDUKACYJNE NA POSZCZEGÓLNE OCENY  
 zgodne z treściami nauczania wskazanymi w ,,Podstawie programowej dla II etapu edukacyjnego (klasy 4–8)”   
i zawartymi w podręczniku ,,Lubię to!” dla klasy 8 (Python)**

Ocenę niedostateczną otrzymuje uczeń, który nie spełnia wymagań edukacyjnych na ocenę dopuszczającą.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tytuł w podręczniku** | **Numer i temat lekcji** | **Ocena dopuszczająca**  **Uczeń:** | **Ocena dostateczna**  **Uczeń:** | **Ocena dobra**  **Uczeń:** | **Ocena bardzo dobra**  **Uczeń:** | **Ocena celująca**  **Uczeń:** |
| **DZIAŁ 1. Arkusz kalkulacyjny** | | | | | | |
| 1.1. Formuły i adresowanie względne w arkuszu kalkulacyjnym | 1. i 2. 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 tekstu oraz wygląd czcionki) | * 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 |
| 1.2. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym | 3. i 4. 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 | 5. i 6. 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 | 7. 8. 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 | 9., 10. i 11. Wprowadzenie do programowania w języku Python | * definiuje pojęcia: algorytm, program, programowanie * podaje kilka sposobów przedstawienia algorytmu | * wymienia różne sposoby przedstawienia algorytmu: opis słowny, 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 | 12., 13. i 14. 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 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 rozwiązywanie zadań matematycznych * tworzy program składający się z kilku funkcji wywoływanych w programie głównym |
| 2.3. Algorytmy na liczbach naturalnych | 15., 16. i 17. 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 while | * 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 |
| 2.4. Algorytmy wyszukiwania | 18. i 19. Algorytmy wyszukiwania | * 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 | * implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym | * samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze | * samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania |
| 2.5. Algorytmy porządkowania | 20. i 21. Algorytmy porządkowania | * 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 | * 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 4. Projekty** | | | | | | |
| 4.1. Dokumentacja szkolnej imprezy sportowej | 22. i 23. 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 | 24., 25. i 26. 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 |