**Wymagania edukacyjne - klasa 8**

Kryteria obejmują zakres ocen 2‒5, nie uwzględniając oceny 1 (niedostatecznej) i 6 (celującej). Ocenę celującą otrzymuje uczeń, który biegle posługuje się wiedzą i umiejętnościami wymaganymi na ocenę bardzo dobrą, zaś uczeń, który nie spełnia wymagań na ocenę dopuszczającą, otrzymuje ocenę niedostateczną.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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:** |
| **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 |
| **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 |
| **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 |
| **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 |
| **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, 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 |
| **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 | * 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 |
| **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 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 while a pętlą for * pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby * wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem |
| **2.4. Algorytmy wyszukiwania** | 18. i 19. 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 |
| **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 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 |
| **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 |
| **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 |
| **4.3. Historia i rozwój informatyki** | 27., 28. i 29. 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** | 30. 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 |

.