

Plan wynikowy dla klasy 8 szkoły podstawowej zgodny z podręcznikiem „Lubię to!” (Python)

Wymagania zamieszczone w planie wynikowym zostały dostosowane do poszczególnych jednostek lekcyjnych i mają na celu ułatwienie planowania lekcji i oceniania uczniów. Są one propozycją, którą każdy nauczyciel powinien zmodyfikować stosownie do możliwości swojego zespołu klasowego.

Tytuł w podręczniku	Numer i temat lekcji	Wymagania (ocena dopuszczająca) Uczeń:	Wymagania (ocena dostateczna) Uczeń:	Wymagania (ocena dobra) Uczeń:	Wymagania (ocena bardzo dobra) Uczeń:	Wymagania (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	<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 bezwzględne i mieszane w arkuszu kalkulacyjnym	3. i 4. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	<ul style="list-style-type: none"> rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym 	<ul style="list-style-type: none"> stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora 	<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 	<ul style="list-style-type: none"> korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych 	<ul style="list-style-type: none"> stosuje zaawansowane funkcje arkusza w tabelach stworzonych 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 programowania w języku Python	9., 10. i 11. Wprowadzenie do programowania w języku Python	<ul style="list-style-type: none"> definiuje pojęcia: algorytm, program, programowanie podaje kilka sposobów przedstawienia algorytmu 	<ul style="list-style-type: none"> 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 	<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"> pisze proste programy w trybie skryptowym języka Python 	<ul style="list-style-type: none"> 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	<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"> buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów konstruuje złożone sytuacje warunkowe (wiele warunków) w algorytmach 	<ul style="list-style-type: none"> pisze programy w języku Python do rozwiązywanie 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"> 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	<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 w wersji z odejmowaniem i z dzieleniem – 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 wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem 	<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 określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym 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 zapisuje algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia implementuje grę w zgadywanie liczby 	<ul style="list-style-type: none"> implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia implementuje algorytm wyszukiwania największej wartości w zbiorze 	<ul style="list-style-type: none"> 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 	<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 • sprawdza działanie programu sortującego dla różnych danych 	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • omawia implementację algorytmu sortowania przez zliczanie • omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie • wprowadza modyfikacje w implementacji algorytmów porządkowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie
• 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
4.3. Historia i rozwój informatyki	27., 28. i 29. Historia i rozwój informatyki	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone 	<ul style="list-style-type: none"> • współpracuje z innymi podczas pracy nad projektem 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu 	<ul style="list-style-type: none"> • współpracuje w grupie podczas pracy nad

		<p>zadania o niewielkim stopniu trudności – znalezienie informacji w internecie, umieszczenie ich w chmurze</p>	<ul style="list-style-type: none"> • analizuje zebrane dane • tworzy projekt prezentacji multimedialnej 	<p>realizuje powierzone zadania</p> <ul style="list-style-type: none"> • tworzy prezentację wg projektu zaakceptowanego przez zespół 	<ul style="list-style-type: none"> • analizuje i weryfikuje pod względem merytorycznym i technicznym przygotowaną prezentację 	<p>projektem, przyjmuje funkcję lidera</p> <ul style="list-style-type: none"> • wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne
4.4. Informatyka w moim przyszłym życiu	30. Informatyka w moim przyszłym życiu	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych 	<ul style="list-style-type: none"> • gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu • projektuje tabele do zapisywania informacji o zawodach • weryfikuje i formatuje przygotowane dokumenty tekstowe 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, przyjmuje rolę lidera • podczas dyskusji przyjmuje funkcję moderatora

Uczeń ma obowiązek wykonywać prace w trakcie lekcji. Jeżeli nie był obecny na lekcji, w następnym tygodniu po powrocie na zajęcia w szkole, ma obowiązek zgłosić do nauczyciela chęć uzupełnienia brakujących prac. Powinien dostarczyć wykonane prace do nauczyciela w następnym tygodniu po otrzymaniu wskazówek do wykonania zadania.

Po upływie dwóch tygodni od powrotu ucznia do szkoły po nieobecności, otrzymuje za brak pracy ocenę niedostateczną. Jeśli uczeń w ciągu tygodnia od otrzymania oceny niedostatecznej nie zgłosi do nauczyciela chęci poprawienia swojej pracy, nie będzie możliwości poprawienia oceny na wyższą od otrzymanej.

Poprawa oceny następuje poprzez wpisanie w kolejnej rubryce oceny wyższej. Ocena niższa pozostaje w dzienniku bez zmian.

Tzw. Średnia ocen ucznia, czy też średnia ważona, nie mają wpływu na wystawioną ocenę końcową na semestr czy też końcowo roczną.

Ocenę końcową wystawia nauczyciel na podstawie otrzymanych prac, sumienności i obowiązkowości ucznia, jego zaangażowania w pracę na lekcji.

Uczeń zawsze może liczyć na wsparcie nauczyciela podczas wykonywania swoich prac. Nauczyciel nie ocenia zdolności ucznia, a jedynie jego pracę i zaangażowanie.

Uczeń ubiegający się o ocenę celującą powinien wykazać się szerszą niż podstawowa wiedzą i umiejętnościami. Nauczyciel może oceniać dodatkowe prace ucznia, które ten wykonuje w domu w wybranym przez siebie programie. Prace należy dostarczać jeden raz w miesiącu.