

Zespół Szkolno-Przedszkolny w Sterkowcu

**Rozkład materiału zgodny z podręcznikiem „Lubię to!”
do INFORMATYKI dla klasy VIII**

rok szk. 2025/2026

mgr Rafał Góra

Rozkład materiału nauczania

Temat w podręczniku	Liczba godzin lekcyjnych proponowanych na dany temat	Numer wymagania
DZIAŁ 1. Arkusz kalkulacyjny (8 GODZIN)		
1.1. Formuły i adresowanie względne w arkuszu kalkulacyjnym	2	II.3c, II.4
1.2. Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	2	II.3c, II.4
1.3. Przedstawianie danych na wykresie	2	II.3c, II.4, II.5
1.4. Zastosowania arkusza kalkulacyjnego	2	II.3c, II.4, II.5
DZIAŁ 2. Programowanie w języku Python (13 GODZIN)		
2.1. Wprowadzenie do programowania w języku Python	3	I.1, III.3
2.2. Piszemy programy w języku Python	3	I.1, II.1
2.3. Algorytmy na liczbach naturalnych	3	I.1, I.2a
2.4. Algorytmy wyszukiwania	2	I.1, I.2b
2.5. Algorytmy porządkowania	2	I.1, I.2b
DZIAŁ 4. Projekty (5 GODZIN)		
4.1. Dokumentacja szkolnej imprezy sportowej	2	II.3c, II.4, IV.1
4.2. Sterowanie obiektem na ekranie	3	II.1, II.2, IV.1

Ogólne kryteria oceniania

Dopuszczająca	Dostateczna	Dobra	Bardzo dobra	Celująca
- Potrafi wprowadzić dane do arkusza kalkulacyjnego i wykonać proste obliczenia (II.3c). - Zna pojęcie formuły i adresowania względnego (II.3c). - Wie, że istnieją funkcje w arkuszu (II.4). - W Pythonie potrafi uruchomić środowisko i wykonać najprostsze polecenie (I.1).	- Stosuje formuły i adresowanie względne w arkuszu (II.3c, II.4). - Potrafi użyć jednej prostej funkcji (np. SUMA) i przedstawić dane na wykresie (II.5). - W Pythonie pisze proste programy z instrukcjami wejścia i wyjścia (I.1, II.1). - Rozumie proste algorytmy na liczbach naturalnych (I.2a).	- Stosuje formuły z adresowaniem względnym i bezwzględnym, korzysta z kilku funkcji (np. MIN, MAX, ŚREDNIA) (II.3c, II.4). - Tworzy poprawne wykresy i interpretuje dane (II.5). - W Pythonie stosuje zmienne, instrukcje warunkowe, proste pętle (I.1, II.1). - Potrafi zapisać algorytm w kodzie i rozwiązywać typowe problemy (np. NWD, suma cyfr) (I.2a, I.2b).	- W arkuszu kalkulacyjnym tworzy złożone formuły, stosuje adresowanie mieszane, używa wielu funkcji w zadaniu praktycznym (II.3c, II.4, II.5). - W Pythonie pisze programy z pętlami zagnieżdżonymi, obsługą list i prostymi algorytmami wyszukiwania i porządkowania (I.1, I.2a, I.2b, II.1). - Potrafi zaprojektować i zrealizować zadanie projektowe (np. dokumentacja imprezy, sterowanie obiektem na ekranie) (II.3c, II.4, IV.1, II.2).	- Twórczo i samodzielnie wykorzystuje arkusz kalkulacyjny – opracowuje dane, stosuje różne funkcje i prezentacje graficzne (II.3c, II.4, II.5). - W Pythonie projektuje i realizuje własne programy rozwiązujące złożone problemy, łączy różne struktury i algorytmy (I.1, I.2a, I.2b, II.1, II.2). - W projektach wykorzystuje kilka narzędzi jednocześnie (arkusz, programowanie, grafikę), opracowuje dokumentację i prezentuje wyniki (IV.1). - Potrafi wyjaśniać innym zasady działania algorytmów i programów.

Szczegółowy rozkład materiału dla klasy VIII

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:
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 stworzonych 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	- omawia algorytm Euklidesa i zapisuje go w wybranej postaci - wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje	- wyjaśnia różnice między instrukcją iteracyjną while a pętlą for - pisze programy obliczające NWD,	pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby

			instrukcję iteracyjną while	go w wybranej postaci	stosując algorytm Euklidesa, oraz wypisujące cyfry 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

Wymagania edukacyjne z informatyki w klasie 8 szkoły podstawowej

1. W zakresie rozumienia, analizowania i rozwiązywania problemów uczeń:
 - wymienia etapy rozwiązywania problemów,
 - wyjaśnia, czym jest algorytm,
 - buduje algorytmy do rozwiązywania problemów,
 - wskazuje specyfikację problemu (dane, wyniki),
 - przedstawia algorytm w postaci listy kroków,
 - tłumaczy, na czym polega sytuacja warunkowa w algorytmie,
 - omawia możliwości wykorzystania arkusza kalkulacyjnego w różnych dziedzinach.
2. W zakresie programowania i rozwiązywania problemów z wykorzystaniem komputera i innych urządzeń cyfrowych uczeń:
 - wyjaśnia, co to znaczy programować,
 - wyjaśnia, na czym polega iteracja (powtarzanie),
 - stosuje pętlę powtórzeniową w tworzonych programach,
 - stosuje sytuację warunkową w tworzonych programach,
 - wykorzystuje zmienne podczas programowania,
 - tworzy procedury z parametrami i bez parametrów,
 - oblicza największy wspólny dzielnik, wykorzystując algorytm Euklidesa,
 - wskazuje największą liczbę w zbiorze, stosując algorytm wyszukiwania,
 - porządkuje elementy w zbiorze metodą wybierania, połowienia i zliczania,
 - wskazuje różnice pomiędzy kodem źródłowym a kodem wynikowym (maszynowym),
 - wskazuje różnice pomiędzy kompilatorem a interpreterem,
 - wyjaśnia, czym jest arkusz kalkulacyjny, wiersz, kolumna i komórka tabeli,
 - wskazuje adres komórki oraz zakres komórek w arkuszu kalkulacyjnym,
 - samodzielnie buduje formuły do wykonywania prostych obliczeń w arkuszu kalkulacyjnym,
 - stosuje formuły wbudowane w program do wykonywania obliczeń w arkuszu kalkulacyjnym,
 - kopiuje formuły, stosując adresowanie względne, bezwzględne oraz mieszane,
 - sprawdza warunek logiczny w arkuszu kalkulacyjnym, korzystając z funkcji JEŻELI,
 - dodaje oraz usuwa wiersze i kolumny w tabeli arkusza kalkulacyjnego,
 - zmienia szerokość kolumn i wysokość wierszy tabeli arkusza kalkulacyjnego,
 - zmienia wygląd komórek w arkuszu kalkulacyjnym,
 - dodaje i formatuje obramowanie komórek tabeli arkusza kalkulacyjnego,
 - scala ze sobą wiele komórek tabeli arkusza kalkulacyjnego,
 - wykorzystuje funkcję zawijania tekstu, aby zmieścić w jednej komórce dłuższe teksty,
 - zmienia format danych wpisanych do komórek arkusza kalkulacyjnego,
 - drukuje tabele utworzone w arkuszu kalkulacyjnym,
 - przedstawia na wykresie dane zebrane w tabeli arkusza kalkulacyjnego,
 - dobiera odpowiedni typ wykresu do rodzaju danych zebranych w tabeli arkusza kalkulacyjnego,
 - wstawia do dokumentu tekstowego tabelę lub wykres arkusza kalkulacyjnego,

- wstawiając tabelę lub wykres arkusza kalkulacyjnego do dokumentu tekstowego, odróżnia obiekt osadzony od obiektu połączanego,
 - sortuje dane w tabeli arkusza kalkulacyjnego w określonym porządku,
 - wyświetla tylko wybrane dane w tabeli arkusza kalkulacyjnego, korzystając z funkcji filtrowania.
3. W zakresie posługiwania się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi uczniów:
- korzysta z różnych urządzeń peryferyjnych,
 - wyszukuje w internecie informacje i dane różnego rodzaju (tekst, obrazy, muzykę, filmy),
 - sprawnie posługuje się urządzeniami elektronicznymi takimi jak skaner, drukarka,
 - prawidłowo nazywa programy, narzędzia i funkcje, z których korzysta,
 - wyjaśnia działanie narzędzi, z których korzysta.
4. W zakresie rozwijania kompetencji społecznych uczniów:
- współpracuje z innymi, wykonując złożone projekty,
 - określa etapy wykonywania złożonego projektu grupowego,
- komunikuje się z innymi przez sieć lokalną oraz przez internet, wykorzystując komunikatory,
 - wysyła i odbiera pocztę elektroniczną,
 - selekcjonuje i ocenia krytycznie informacje znalezione w internecie,
5. W zakresie przestrzegania praw i zasad bezpieczeństwa uczniów:
- przestrzega zasad bezpiecznej i higienicznej pracy przy komputerze,
 - wymienia i opisuje rodzaje licencji na oprogramowanie,
 - przestrzega postanowień licencji na oprogramowanie i materiały pobrane z internetu,
 - przestrzega zasad etycznych, korzystając z komputera i internetu,
 - dba o swoje bezpieczeństwo podczas korzystania z internetu,
 - przestrzega przepisów prawa podczas korzystania z internetu,
 - wie, czym jest netykieta, i przestrzega jej zasad, korzystając z internetu.

Wymagania na poszczególne oceny

Wymagania na każdy stopień wyższy niż **dopuszczający** obejmują również wymagania na wszystkie stopnie niższe.

Ocena				
Stopień dopuszczający Uczeń:	Stopień dostateczny Uczeń:	Stopień dobry Uczeń:	Stopień bardzo dobry Uczeń:	Stopień celujący Uczeń:

• 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) • rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym • wstawia wykres do arkusza kalkulacyjnego • korzysta z arkusza kalkulacyjnego w celu stworzenia kalkulacji wydatków • definiuje pojęcia: algorytm, program, programowanie • podaje kilka sposobów przedstawienia algorytmu • tłumaczy, do czego używa się zmiennych w programach • pisze proste programy w trybie skryptowym języka Python z wykorzystaniem zmiennych • wyjaśnia działanie operatora modulo • wyjaśnia algorytm badania podzielności liczb • wyjaśnia potrzebę wyszukiwania informacji w zbiorze • sprawdza działanie programów wyszukujących element w zbiorze	• określa zasady wprowadzania danych do komórek arkusza kalkulacyjnego • dodaje i usuwa wiersze oraz kolumny w tabeli • stosuje w arkuszu podstawowe funkcje: (SUMA, ŚREDNIA), wpisuje je ręcznie oraz korzysta z kreatora • omawia i modyfikuje poszczególne elementy wykresu • zapisuje w tabeli arkusza kalkulacyjnego dane otrzymane z prostych doświadczeń i przedstawia je na wykresie • 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 • wykonuje obliczenia w języku Python • omawia działanie operatorów arytmetycznych • stosuje listy w języku Python oraz	• tworzy proste formuły obliczeniowe • wyjaśnia, czym jest adres względny • 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 • dobiera odpowiedni wykres do rodzaju danych • sortuje oraz filtruje dane w arkuszu kalkulacyjnym • wymienia przykładowe środowiska programistyczne • wyjaśnia, czym jest specyfikacja problemu • opisuje etapy rozwiązywania problemów • opisuje etapy powstawania programu komputerowego • zapisuje proste polecenia języka Python • 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	• kopiuje utworzone formuły obliczeniowe, wykorzystując adresowanie względne • korzysta z biblioteki funkcji, aby wyszukiwać potrzebne funkcje • stosuje adresowanie względne, bezwzględne lub mieszane w zaawansowanych formułach obliczeniowych • tworzy wykres dla więcej niż jednej serii danych • tworzy prosty model (na przykładzie rzutu sześcienną kostką do gry) w arkuszu kalkulacyjnym • stosuje filtry niestandardowe • pisze proste programy w trybie skryptowym języka Python • 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 • wyjaśnia różnice między instrukcją iteracyjną while a pętlą for • pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby • samodzielnie zapisuje w	• samodzielnie tworzy i kopiuje skomplikowane formuły obliczeniowe • stosuje zaawansowane funkcje arkusza w tabelach tworzonych na własne potrzeby • tworzy rozbudowane wykresy dla wielu serii danych • przygotowuje rozbudowane arkusze kalkulacyjne korzysta z arkusza kalkulacyjnego do analizowania doświadczeń z innych przedmiotów • zapisuje algorytmy różnymi sposobami oraz pisze programy o większym stopniu trudności • pisze programy w języku Python do rozwiązywania zadań matematycznych • tworzy program składający się z kilku funkcji wywoływanych w programie głównym • pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby • samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania • samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez
---	---	--	---	--

• wyjaśnia potrzebę porządkowania danych • sprawdza działanie programu sortującego dla różnych danych • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • testuje grę na różnych etapach • współpracuje w grupie podczas pracy nad projektem	operatory logiczne • zapisuje w postaci listy kroków algorytm badania podzielności liczb naturalnych • wykorzystuje w programach instrukcję iteracyjną while • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym elementu największego i najmniejszego • zapisuje w wybranej formie algorytm porządkowania metodą przez wybieranie • omawia implementację algorytmu sortowania przez wybieranie • stosuje pętle zagnieżdżone i wyjaśnia, jak działają • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej • wprowadza dane do zaprojektowanych tabel • bierze udział w pracach nad wypracowaniem koncepcji gry • współpracuje w grupie podczas pracy nad projektem	i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości • omawia algorytm Euklidesa i zapisuje go w wybranej postaci • wyjaśnia algorytm wyodrębniania cyfr danej liczby i zapisuje go w wybranej postaci • implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym • omawia funkcje zastosowane w kodzie źródłowym algorytmu sortowania przez wybieranie • przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy • współpracuje w grupie podczas pracy nad projektem • programuje wybrane funkcje i elementy gry • opracowuje opis gry	wybranej postaci algorytm wyszukiwania elementu w zbiorze • implementuje algorytm porządkowania metodą przez wybieranie • wprowadza modyfikacje w implementacji algorytmu porządkowania przez wybieranie • bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki • współpracuje w grupie podczas pracy nad projektem • implementuje i optymalizuje kod źródłowy gry, korzystając z wypracowanych założeń	wybieranie • 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 • rozbudowuje grę o nowe elementy • współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
---	---	--	---	--

PRZEDMIOTOWY SYSTEM OCENIANIA Z INFORMATYKI

ZASADY PRACY:

1. Prace klasowe, sprawdziany i inne formy sprawdzenia wiedzy po dziale są obowiązkowe.
2. Jeżeli uczeń opuścił pracę klasową itp. z przyczyn losowych, to powinien napisać ją w ciągu dwóch tygodni od dnia powrotu do szkoły.
3. Uczeń może poprawić ocenę pracy pisemnej w ciągu tygodnia od dnia oddania sprawdzonych prac. Może ją poprawić poza swoimi lekcjami.

4. Przy poprawianiu prac kasowych, kryteria ocen nie zmieniają się, a otrzymana ocena jest wpisana do dziennika.
5. Uczeń raz poprawiający daną pracę nie może poprawiać jej kolejny raz.
6. Krótkie sprawdziany (kartkówki) mogą obejmować materiał z najwyżej trzech ostatnich tematów.
7. Uczniowie nieobecni na krótkich sprawdzianach piszą je możliwie najbliższym terminie.
8. Krótkie sprawdziany nie podlegają poprawie.
9. Nie ocenia się uczniów do trzech dni po dłuższej usprawiedliwionej nieobecności w szkole.
10. Uczeń ma prawo trzykrotnie w ciągu semestru zgłosić nie przygotowanie do lekcji tzn. brak zeszytu, zadania domowego, zeszytu ćwiczeń, brak notatek lub przyborów.
11. Uczeń ma prawo zapoznać się z każdą pracą pisemną.
12. Prace klasowe, sprawdziany nauczyciel zbiera i przechowuje do końca roku szkolnego.

ODPOWIEDZI USTNE:

Ocena bardzo dobra	– uczeń samodzielnie i sprawnie rozwiązuje zadanie problemowe, komentuje rozwiązanie, jasno formułuje wypowiedzi w sposób zrozumiały dla innych uczniów, sprawnie operuje poświadczonymi symbolami.
Ocena dobra	– uczeń samodzielnie, z użyciem języka informatycznego rozwiązuje zadania o średnim stopniu trudności – typowe.
Ocena dostateczna	– uczeń, przy użyciu elementarnego języka informatycznego rozwiązuje proste zadania.
Ocena dopuszczająca	– uczeń za pomocą nauczyciela rozwiązuje najprostsze zadania – działania, zna

definicję, pojęcia, umie je wypowiedzieć „własnymi słowami”.

Ocena niedostateczna – uczeń przy pomocy nauczyciela nie potrafi wykonać najprostszych zadań.

PRACE PISEMNE:

1. Prace klasowe, sprawdziany, i inne formy sprawdzenia wiedzy – z danego działu, 1 godzina
2. Krótkie sprawdziany (kartkówki), 15 minut.

PRACE DOMOWE:

1. Wykonywanie zadań w zeszycie lub ćwiczeniach – pomaga przećwiczyć omawiany temat.

INNE FORMY SPRAWDZANIA WIEDZY:

1. Projekty multimedialne – uczniowie tworzą prezentacje, animacje, krótkie filmy czy proste strony internetowe, demonstrując umiejętność obsługi programów i zdobytą wiedzę.
2. Tworzenie gier – korzystając z wizualnych języków programowania (Scratch, Python), uczniowie projektują własne gry, co rozwija kreatywność, logiczne myślenie i umiejętności programistyczne.
3. Zadania praktyczne z oprogramowaniem – uczniowie wykonują konkretne zadania przy komputerze, np. tworzenie dokumentów, arkuszy kalkulacyjnych, grafik, czy prostych baz danych – co pozwala sprawdzić biegłość w pracy z aplikacjami.
4. Projekty zespołowe – uczniowie w parach lub małych grupach realizują wspólne zadania (np. przygotowanie plakatu, gazetki szkolnej w wersji cyfrowej, filmu promującego szkołę).

5. Quizy i gry edukacyjne online – korzystanie z platform takich jak Kahoot, Quizizz, Wordwall, które pozwalają szybko i atrakcyjnie sprawdzić wiedzę.
6. Kodowanie bez komputera (Unplugged) – zadania ruchowe, planszowe czy łamigłówki rozwijające logiczne myślenie i rozumienie podstaw algorytmiki.
7. Projekty badawcze i eksploracja Internetu – zadania polegające na wyszukiwaniu, analizie i ocenie wiarygodności informacji, a następnie zaprezentowaniu wyników.
8. Rozmowy i dyskusje – indywidualne lub grupowe rozmowy sprawdzające rozumienie zagadnień (np. bezpieczeństwo w sieci, netykieta, sztuczna inteligencja).
9. Ocenianie koleżeńskie – uczniowie oceniają prace swoich kolegów według ustalonych kryteriów, co uczy krytycznego myślenia i pracy z feedbackiem.
10. Samoocena – uczeń samodzielnie ocenia swoje umiejętności w danym zakresie (np. obsługa arkusza, tworzenie animacji) na specjalnej karcie lub w aplikacji.

SKALA OCENIANIA:

Każde zadanie jest punktowane. Suma punktów jest zamieniana na ocenę wg następującej skali:

0 - 29%	niedostateczny
30 - 31%	- dopuszczający
32 - 47%	dopuszczający
48 - 49%	+ dopuszczający
50 - 51%	- dostateczny
52 - 67%	dostateczny
68 - 69%	+ dostateczny
70 - 71%	- dobry
72 - 87%	dobry
88 - 89%	+ dobry

90 - 91%	- bardzo dobry
92 - 97%	bardzo dobry
98 - 99%	+ bardzo dobry
100%	celujący

INNE FORMY AKTYWNOŚCI UCZNIA:

1. Praca informatyczna (poza lekcjami informatyki):
 - awans do następnego etapu konkursu lub osiągnięcie tytułu laureata – częściowa ocena celująca.
 - wyniki wyższe niż przeciętne – ocena bardzo dobra,
 - za udział w konkursie ocena bardzo dobra.
2. Praca na lekcji – częste zgłaszanie się na lekcji i udzielanie prawidłowych odpowiedzi, umiejętność pracy w grupach, itp.
 - ocena: dostateczna - bardzo dobra.
3. Prace domowe i projekty: Wykonywanie dodatkowych zadań, np. tworzenie modeli brył geometrycznych.
4. Praca w grupach: Ocena współpracy i aktywności uczniów podczas pracy zespołowej.

Ocena końcowa nie jest średnią arytmetyczną.

INFORMACJA ZWROTNA:

Nauczyciel – uczeń – rodzic/opiekun/.

1. Każda ocena z odpowiedzi ustnej jest wpisana do zeszytu przy danym temacie. Nauczyciel przekazuje uczniowi komentarz ustny do każdej wystawionej oceny – przy każdej ocenie podpis rodzica/opiekuna/.
2. Informacja o ocenach odbywać się będzie za pośrednictwem Dziennika elektronicznego lub podczas konsultacji indywidualnych.

OCENA WYŻSZA NIŻ PROPONOWANA:

1. Uczeń może zwrócić się z prośbą o podwyższenie oceny proponowanej.
2. Nauczyciel udziela indywidualnie warunków uzyskania oceny wyższej, np. poprzez:
 - a) Poprawę konkretnych prac pisemnych,
 - b) Odpowiedź ustną z wyznaczonego zakresu materiału,

- c) Testu sprawdzającego obejmującego cały rok szkolny lub jego część.
3. Szczegółowe warunki i tryb uzyskania wyższej niż przewidywana oceny klasyfikacyjnej opisuje Statut.

Opracowanie: mgr Rafał Góra