

WYMAGANIA EDUKACYJNE

z informatyki dla klasy 8 szkoły podstawowej

Rok szkolny 2026/2027

Zasada ogólna

Wymagania mają charakter narastający. Uczeń ubiegający się o daną ocenę powinien spełniać również wymagania przewidziane dla ocen niższych.

Zasady oceniania

Ocenie podlegają

- wiadomości i umiejętności ucznia sprawdzane podczas pracy na lekcji, wykonywania zadań i projektów oraz odpowiedzi ustnych;
- aktywność i zaangażowanie ucznia podczas zajęć.

Sposoby sprawdzania wiedzy i umiejętności

Obowiązuje skala ocen od 1 do 6. Uczeń może otrzymywać oceny za:

- sprawdziany, testy, zadania i projekty przeprowadzane lub wykonywane po zakończeniu danego tematu albo działu;
- odpowiedzi ustne oraz pracę na lekcji;
- aktywność i zaangażowanie podczas zajęć;
- udział i osiągnięcia w konkursach przedmiotowych.

Zasady poprawiania ocen częściowych

Uczeń może poprawić ocenę z kartkówki, zadania lub projektu w terminie wyznaczonym przez nauczyciela. Chęć poprawy zgłasza nauczycielowi. Jeżeli uczeń jest nieobecny w ustalonym terminie, na przykład z powodu choroby, nowy termin poprawy ustalany jest indywidualnie.

Pozostałe ustalenia

1. Sprawdziany, testy, zadania i projekty są obowiązkowe. Uczeń nieobecny w czasie sprawdzianu lub testu pisze go w terminie wyznaczonym przez nauczyciela, nie później niż w ciągu dwóch tygodni od dnia powrotu do szkoły.
2. Jeżeli uczeń jest nieobecny podczas oceniania indywidualnego zadania lub projektu, wykonuje pracę samodzielnie i przedstawia ją nauczycielowi w wyznaczonym terminie, nie później niż w ciągu dwóch tygodni od dnia powrotu do szkoły.
3. Uczeń ma prawo jeden raz w semestrze zgłosić nieprzygotowanie do lekcji. Nie dotyczy to zapowiadanych sprawdzianów, testów, zadań i projektów.
4. Po nieobecności uczeń ma obowiązek uzupełnić wiadomości i zadania realizowane podczas lekcji. Wykonywane zadania projektowe przekazuje nauczycielowi w uzgodnionej formie.

Ocena dopuszczająca – 2

Uczeń:

- omawia zastosowanie i budowę arkusza kalkulacyjnego;
- określa adres komórki;
- wprowadza do komórek dane różnego rodzaju;
- formatuje zawartość komórek, w tym wyrównanie tekstu i wygląd czcionki;
- rozróżnia adresowanie względne, bezwzględne i mieszane;
- wstawia wykres do arkusza kalkulacyjnego;
- wykorzystuje arkusz do przygotowania prostej kalkulacji wydatków;
- wyjaśnia pojęcia: algorytm, program i programowanie;
- podaje sposoby przedstawiania algorytmu;
- wyjaśnia zastosowanie zmiennych w programach;
- pisze proste programy w języku Python z wykorzystaniem zmiennych;
- wyjaśnia działanie operatora modulo i algorytmu badania podzielności liczb;
- sprawdza działanie gotowych programów wyszukujących i sortujących dane;
- wykonuje powierzone zadania w projekcie zespołowym o niewielkim stopniu trudności.

Ocena dostateczna – 3

Uczeń:

- określa zasady wprowadzania danych do komórek arkusza;
- dodaje i usuwa wiersze oraz kolumny w tabeli;
- stosuje funkcje SUMA i ŚREDNIA, wpisując je ręcznie lub korzystając z kreatora;
- omawia i modyfikuje elementy wykresu;
- zapisuje w arkuszu dane z prostych doświadczeń i przedstawia je na wykresie;
- wymienia sposoby przedstawiania algorytmu, w tym opis słowny i listę kroków;
- poprawnie formułuje problem do rozwiązania;
- rozróżnia interaktywny i skryptowy tryb pracy w języku Python;
- wyświetla tekst i wykonuje obliczenia w języku Python;
- stosuje operatory arytmetyczne i logiczne oraz listy;
- wykorzystuje instrukcję iteracyjną while;
- zapisuje algorytm badania podzielności, wyszukiwania elementu i sortowania przez wybieranie;
- stosuje pętle zagnieżdżone;
- wprowadza dane do przygotowanych tabel i współpracuje przy tworzeniu projektu.

Ocena dobra – 4

Uczeń:

- tworzy proste formuły obliczeniowe w arkuszu;
- wyjaśnia, czym jest adres względny;
- wykorzystuje funkcję JEŻELI do tworzenia algorytmów z warunkami;
- ustawia format danych zgodny z zawartością komórki;
- stosuje w formułach adresowanie względne, bezwzględne i mieszane;
- dobiera odpowiedni typ wykresu do rodzaju danych;
- sortuje i filtruje dane w arkuszu;
- wymienia środowiska programistyczne i wyjaśnia pojęcie specyfikacji problemu;
- opisuje etapy rozwiązywania problemu i powstawania programu;
- stosuje w Pythonie instrukcje warunkowe if i if else;
- wykorzystuje pętlę for oraz definiuje funkcje;
- omawia algorytm Euklidesa i algorytm wyodrębniania cyfr liczby;
- implementuje wyszukiwanie elementu w zbiorze nieuporządkowanym;
- przygotowuje dokumentację projektu, wykonuje obliczenia, projektuje tabele i wykresy oraz programuje wybrane elementy.

Ocena bardzo dobra – 5

Uczeń:

- kopiuje formuły, wykorzystując adresowanie względne;
- wyszukuje potrzebne funkcje w bibliotece funkcji;
- stosuje adresowanie względne, bezwzględne i mieszane w zaawansowanych formułach;
- tworzy wykres dla więcej niż jednej serii danych;
- tworzy prosty model w arkuszu kalkulacyjnym;
- stosuje filtry niestandardowe;
- pisze programy w trybie skryptowym języka Python;
- konstruuje złożone sytuacje warunkowe;
- tworzy programy zawierające instrukcje warunkowe, pętle i funkcje;
- rozpoznaje błędy zwracane przez interpreter, czyta kod źródłowy i opisuje jego działanie;
- wyjaśnia różnicę między pętlami while i for;
- pisze programy obliczające NWD algorytmem Euklidesa i wyodrębniające cyfry liczby;
- implementuje i modyfikuje algorytm sortowania przez wybieranie;
- przygotowuje zestawienia, drukuje wyniki i współpracuje przy implementacji projektu.

Ocena celująca – 6

Uczeń:

- spełnia wymagania przewidziane na ocenę bardzo dobrą;
- samodzielnie stosuje poznane wiadomości i umiejętności w sytuacjach trudnych, złożonych i nietypowych;
- projektuje rozbudowane arkusze kalkulacyjne wykorzystujące złożone formuły, funkcje, różne rodzaje adresowania oraz filtry;
- samodzielnie analizuje dane i dobiera właściwy sposób ich prezentacji;
- opracowuje własne algorytmy rozwiązujące nietypowe problemy;
- tworzy w języku Python autorskie programy wykorzystujące złożone warunki, pętle, funkcje i listy;
- modyfikuje oraz optymalizuje algorytmy wyszukiwania i porządkowania danych;
- samodzielnie diagnozuje i poprawia błędy w kodzie;
- łączy arkusz kalkulacyjny, algorytmikę i programowanie w jednym złożonym projekcie;
- pełni odpowiedzialną rolę w zespole i wspiera innych uczestników projektu;
- przygotowuje kompletną dokumentację oraz prezentuje rezultaty pracy;
- osiąga sukcesy w konkursach lub podejmuje dodatkowe aktywności informatyczne.

Zakres materiału

Wymagania obejmują następujące obszary: arkusz kalkulacyjny, algorytmika, programowanie w języku Python, wyszukiwanie i porządkowanie danych oraz realizacja projektów zespołowych.