

WYMAGANIA EDUKACYJNE Z INFORMATYKI

Klasa 8 Szkoły Podstawowej (Program „Teraz bajty” – wariant Python)

Zasady oceniania i konstrukcja wymagań

Przedmiotowy System Oceniania (PSO) został opracowany w oparciu o podstawę programową. Ocenianie ma charakter wspierający (elementy oceniania kształtującego, jasne kryteria sukcesu, informacja zwrotna). Wymagania edukacyjne mają charakter narastający (hierarchiczny) – osiągnięcie wyższego stopnia szkolnego zakłada opanowanie umiejętności z poziomów niższych.

Dostosowanie wymagań edukacyjnych dla uczniów ze SPE

Dla uczniów ze specjalnymi potrzebami edukacyjnymi (posiadających orzeczenie, opinię PPP lub objętych pomocą psychologiczno-pedagogiczną) stosuje się zindywidualizowane zasady i formy dostosowania wymagań:

- **Wsparcie w zakresie arkusza kalkulacyjnego i programowania w języku Python:** Uproszczenie złożonych procedur i konstrukcji programistycznych do podstawowych kroków; udostępnianie instrukcji obrazkowych/kart pracy z wzorami funkcji oraz gotowych fragmentów kodu Python do analizy i modyfikacji zamiast pisania od zera.
- **Wydłużony czas i podział zadań:** Dzielenie obszernych zadań projektowych (np. dokumentacja imprezy sportowej, skrypty sterowania obiektem) na mniejsze, etapowe ćwiczenia z bieżącą kontrolą i wsparciem nauczyciela.
- **Specyficzne trudności w uczeniu się (dysleksja, dysgrafia, zaburzenia motoryki):** Nieocenianie błędów składniowych i literowych w nazwach zmiennych, komendach Pythona czy adresach komórek; ocena skupiona na logice i pomyśle działania.
- **Docenianie wkładu pracy i aktywności:** Uwzględnianie stopnia zaangażowania, wysiłku i indywidualnego postępu ucznia w stosunku do jego możliwości psychofizycznych.

Ocena śródroczna / roczna	Wymagania edukacyjne (poziom działań ucznia)
2 – dopuszczający (poziom zapamiętywania / odtwarzania)	Uczeń rozpoznaje podstawowe pojęcia i wykonuje proste czynności według podanego wzoru: • wprowadza dane do arkusza kalkulacyjnego oraz wykonuje proste obliczenia z użyciem podstawowych formuł, • rozpoznaje adresowanie względne w komórkach arkusza kalkulacyjnego, • tworzy podstawowy wykres na podstawie wskazanych danych w arkuszu kalkulacyjnym, • uruchamia środowisko programistyczne Python oraz wpisuje i uruchamia proste komendy (np. <code>print()</code>, <code>input()</code>), • z pomocą nauczyciela modyfikuje proste skrypty w języku Python, • rozpoznaje proste algorytmy na liczbach naturalnych zaprezentowane na lekcji, • bierze udział w tworzeniu prostej dokumentacji projektu (np. szkolnej imprezy sportowej) według podanego wzoru, • wykorzystuje gotowy skrypt w języku Python do sterowania obiektem na ekranie.
3 – dostateczny (poziom rozumienia)	Uczeń rozumie zastosowanie narzędzi informatycznych i stosuje je w typowych sytuacjach: • stosuje funkcje arkusza kalkulacyjnego (np. SUMA, ŚREDNIA, MAX, MIN) w prostych obliczeniach, • stosuje adresowanie względne oraz wyjaśnia zasadę działania adresowania bezwzględnego w arkuszu, • tworzy i czytelnie formatuje wykresy (dobiera tytuł, opisy osi, legendę), • stosuje zmienne, podstawowe typy danych oraz instrukcje wejścia i wyjścia w języku Python, • buduje w Pythonie proste programy wykorzystujące instrukcje warunkowe (if, else) oraz podstawowe pętle (for, while),

Ocena śródroczna / roczna	Wymagania edukacyjne (poziom działań ucznia)
	• zapisuje i uruchamia proste algorytmy na liczbach naturalnych w języku Python (np. sprawdzanie parzystości liczby), • współpracuje w zespole projektowym przy przygotowywaniu dokumentacji i prezentacji wyników, • tworzy prosty skrypt do sterowania obiektem na ekranie w wybranym środowisku / języku Python.
4 – dobry <i>(poziom zastosowania)</i>	Uczeń sprawnie wykorzystuje zdobyte wiadomości do samodzielnego rozwiązywania zadań: • stosuje w praktyce adresowanie bezwzględne i mieszane do rozwiązywania problemów obliczeniowych w arkuszu kalkulacyjnym, • dobiera odpowiedni typ wykresu do charakteru analizowanych danych oraz dostosowuje jego parametry, • projektuje i pisze w języku Python programy z wykorzystaniem złożonych warunków logicznych i pętli, • samodzielnie implementuje algorytmy na liczbach naturalnych (np. badanie podzielności, rozkład liczby na cyfry), • implementuje w języku Python podstawowe algorytmy wyszukiwania (np. wyszukiwanie liniowe), • opisuje przebieg algorytmów porządkowania (sortowania) i wykonuje ich prostą implementację, • samodzielnie przygotowuje przejrzystą dokumentację projektu z wykorzystaniem odpowiednich narzędzi cyfrowych, • programuje złożone sterowanie obiektem na ekranie z reagowaniem na zdarzenia i klawisze.
5 – bardzo dobry <i>(poziom analizy i rozwiązywania problemów)</i>	Uczeń analizuje problemy, dobiera optymalne metody i wyciąga wnioski: • tworzy zaawansowane modele w arkuszu kalkulacyjnym łącznie z wykorzystaniem złożonych funkcji oraz przeliczaniem danych przy użyciu adresowania mieszanego, • przeprowadza analizę danych tabelarycznych i przedstawia wyciągnięte wnioski w postaci czytelnych wykresów i podsumowań, • biegle posługuje się strukturami danych (np. listami) i pętlami w języku Python do rozwiązywania nietypowych zadań, • samodzielnie implementuje i analizuje efektywność algorytmów wyszukiwania oraz sortowania (np. sortowanie przez wybieranie / bąbelkowe), • projektuje i realizuje kompleksowy projekt zespołowy (np. dokumentację imprezy sportowej) z zachowaniem wysokiej estetyki i czytelności, • tworzy dopracowaną aplikację / skrypt sterujący obiektem z rozbudowaną logiką i obsługą kolizji / granic ekranu.
6 – celujący <i>(poziom tworzenia i ewaluacji)</i>	Uczeń wykazuje się twórczą postawą, realizuje zadania nietypowe i wykraczające poza program: • samodzielnie formułuje i rozwiązuje złożone, wielowątkowe problemy analityczne i programistyczne, • tworzy autorskie, zaawansowane programy w języku Python z optymalizacją kodu, wykorzystaniem własnych funkcji lub zewnętrznych bibliotek, • projektuje i wdraża rozbudowane projekty informatyczne integrujące arkusz kalkulacyjny, programowanie oraz multimedia,

Ocena śródroczna / roczna	Wymagania edukacyjne (poziom działań ucznia)
	• wykazuje się wzorową postawą w zakresie etyki cyfrowej, poszanowania praw autorskich i bezpieczeństwa w sieci, • osiąga sukcesy w konkursach informatycznych / programistycznych lub występuje w roli lidera i mentora wspierającego innych uczniów w zespole.