

KRYTERIA OCENIANIA Z INFORMATYKI

I. Formy oceniania wiadomości i umiejętności uczniów

- | | |
|---------------------|------------------------|
| a) odpowiedzi ustne | e) aktywność na lekcji |
| b) kartkówki | f) projekty |
| c) sprawdziany | g) referaty |
| d) prace domowe | h) udział w konkursach |

II. Zasady oceniania

1. Sprawdziany obejmują zakresem cały dział materiału, mogą być poprzedzone lekcją powtórzeniową. Zapowiedziane na tydzień wcześniej przed zaplanowanym terminem ich napisania.
2. Odpowiedzi ustne i kartkówki są oceniane równorzędnie, mogą być zapowiedziane i niezapowiedziane. Obejmują zakres bieżącego materiału, omawiany aktualnie dział, ew. 3 ostatnie tematy.
3. Aktywność na lekcji obejmuje:
 - odpowiedzi na krótkie pytania przy powtórce na początku lekcji,
 - odpowiedzi na pytania problemowe,
 - aktywny udział w trakcie lekcji,
 - poszukiwanie materiałów związanych z bieżącymi tematami zajęć.
4. Opracowanie referatu polega na jego przygotowaniu i samodzielnej prezentacji.
5. Wykonanie prac, projektów dotyczy zagadnień omawianych podczas zajęć lekcyjnych; kształci umiejętności zdobyte podczas lekcji.
6. Udział w konkursach oceniany jest w zależności od uzyskanego wyniku. Zdobywanie tytułu finalisty lub laureata umożliwia uzyskanie oceny celującej na zakończenie roku.

III. Inne

1. Uczeń nie ma prawa do niesamodzielnej pracy na sprawdzianach i klasówkach czy przygotowaniu zadań i referatów, takie ewidentne próby kończą się wpisaniem do dziennika oznaczenia P.
2. Uczeń, który był nieobecny w szkole podczas sprawdzianu otrzymuje ocenę „0” i ma obowiązek napisania go w terminie ustalonym indywidualnie z nauczycielem.

IV. Informacje o terminach

1. Uczniowie są informowani na bieżąco o zdobytych punktach z odpowiedzi ustnych, plusach i minusach.
2. Termin oddawania sprawdzonych prac przez nauczyciela – dwa tygodnie.
3. Rodzice są informowani o punktach poprzez dziennik elektroniczny, na wywiadówkach szkolnych oraz na spotkaniach indywidualnych na prośbę rodzica, nauczyciela lub ucznia.

V. Kryteria ocen

Prace pisemne oceniane są za pomocą skali procentowej:

- 97-100% celujący, - 86-96% bardzo dobry, - 70-85% dobry, - 51-69% dostateczny, - 40-50% dopuszczający, - 0 - 39% niedostateczny

VI. Ocenianie semestralne/roczne

Ocena semestralna (roczna) jest oceną oceniającą prace ucznia przez cały semestr (rok). Ocena roczna zależy od wszystkich ocen bieżących (także ocen kształtujących), jakie uczeń otrzymał w ciągu roku szkolnego. **Ocena semestralna i roczna nie jest średnią arytmetyczną wszystkich ocen otrzymanych przez ucznia.**

Przy wystawianiu oceny semestralnej i rocznej brane są również pod uwagę: systematyczność pracy, zaangażowanie, wypełnienie założonych celów.

Przy wystawianiu oceny rocznej brane są pod uwagę oceny zarówno z I jak i II semestru.

VII. Ubieganie się o wyższą niż przewidywana ocena semestralna lub roczna.

W celu uzyskania wyższej niż przewidywana przez nauczyciela oceny rocznej uczeń zobowiązany jest do poprawy wskazanych przez nauczyciela ocen cząstkowych, wykonania zadania dodatkowego na przykład w postaci opracowania referatu bądź dłuższej wypowiedzi pisemnej na zadany temat.

INFORMATYKA – PRYWATNA SZKOŁA PODSTAWOWA ACADEMOS 2025/2026 Ogólne wymagania edukacyjne z informatyki na poszczególne oceny szkolne

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

Wymagania na ocenę celującą obejmują stosowanie przyswojonych informacji i umiejętności w sytuacjach trudnych, złożonych i nietypowych.

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				
<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
<ul style="list-style-type: none"> rozumie różnice między adresowaniem względnym, bezwzględnym i mieszany 	<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 tworzonych na własne potrzeby
<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
<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				
<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, lista kroków 	<ul style="list-style-type: none"> wymienia przykładowe środowiska programistyczne wyjaśnia, czym jest specyfikacja problemu 	<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

	<ul style="list-style-type: none"> 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"> opisuje etapy rozwiązywania problemów opisuje etapy powstawania programu komputerowego zapisuje proste polecenia języka 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> definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości 	<ul style="list-style-type: none"> 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 	<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"> 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 i 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 	<ul style="list-style-type: none"> pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby
<ul style="list-style-type: none"> wyjaśnia potrzebę wyszukiwania informacji w zbiorze 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 	<ul style="list-style-type: none"> implementuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym 	<ul style="list-style-type: none"> samodzielnie zapisuje w wybranej postaci algorytm wyszukiwania elementu w zbiorze 	<ul style="list-style-type: none"> samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania
<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 metodą przez wybieranie omawia implementację algorytmu sortowania przez wybieranie stosuje pętle zagnieżdżone i wyjaśnia, jak działają 	<ul style="list-style-type: none"> omawia funkcje zastosowane w kodzie źródłowym algorytmu sortowania przez wybieranie 	<ul style="list-style-type: none"> implementuje algorytm porządkowania metodą przez wybieranie wprowadza modyfikacje w implementacji algorytmu porządkowania przez wybieranie 	<ul style="list-style-type: none"> samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie
• DZIAŁ 4. Projekty				
<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej 	<ul style="list-style-type: none"> przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy zestawienia

mu zadania o niewielkim stopniu trudności	<ul style="list-style-type: none"> wprowadza dane do zaprojektowanych tabel 	<ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem 	<p>sportowej, przygotowuje zestawienia, drukuje wyniki</p> <ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem 	<p>zawierające zaawansowane formuły, wykresy oraz elementy graficzne</p> <ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
<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