

Wymagania Edukacyjne z Informatyki dla Klasy VII

Autorski program nauczania "Świat cyfrowych możliwości"

Autorzy: Damian Nowak, Iwona Ślężona-Plawgo

Wstęp i Zasady Oceniania

Chcemy, aby wiedzieli Państwo, że w naszej szkole ocenianie jest narzędziem, które ma pomóc uczniom w nauce. To nie tylko stopień w dzienniku, ale przede wszystkim informacja zwrotna dla ucznia, Państwa i dla mnie — co uczeń już umie, co robi dobrze, a nad czym warto jeszcze popracować. Skupiamy się na rozwoju dziecka i na tym, co potrafi zrobić z nową wiedzą. Zasady te są również dostępne na klasowym Padlecie, do którego dostęp uczeń otrzyma na lekcji.

Wymagania na ocenę wyższą obejmują również wymagania na niższe oceny:

- aby uzyskać ocenę dostateczną należy również spełnić wymagania na ocenę dopuszczającą,
- aby uzyskać ocenę dobrą należy również spełnić wymagania na ocenę dopuszczającą i dostateczną,
- aby uzyskać ocenę bardzo dobrą należy również spełnić wymagania na ocenę dopuszczającą, dostateczną i dobrą,
- aby uzyskać ocenę celującą należy również spełnić wymagania na ocenę dopuszczającą, dostateczną, dobrą i bardzo dobrą.

Uczeń otrzymuje ocenę niedostateczną, jeżeli nie spełnił wymagań na ocenę dopuszczającą.

Ocena śródroczna nie jest oceną klasyfikacyjną, lecz informacyjną. Jest ustalana na podstawie ocen cząstkowych, analizy osiągnięć edukacyjnych ucznia oraz odnosi się do stopnia realizacji wymagań edukacyjnych w I półroczu danego roku szkolnego.

Ocena końcoworoczna jest oceną klasyfikacyjną i jest ustalana na podstawie całorocznych osiągnięć edukacyjnych ucznia i odnosi się do realizacji wymagań edukacyjnych w całym roku szkolnym.

Co oznaczają poszczególne oceny? Wyjaśnienie na przykładzie

Wyobraźmy sobie, że zadaniem ucznia jest stworzenie w programie graficznym logo dla kółka zainteresowań. Zobaczmy, jak praca nad tym zadaniem przełoży się na ocenę.

- Ocena celująca (6): Zastosowanie wiedzy w sposób twórczy.

Otrzyma ją uczeń, który nie tylko perfekcyjnie opanuje materiał, ale także wykaże się własną inicjatywą. Oznacza to, że potrafi analizować, modyfikować i tworzyć zupełnie nowe rozwiązania.

Przykład: Uczeń tworzy oryginalne logo od podstaw, a dodatkowo przygotowuje jego dwie inne wersje (np. czarno-białą i uproszczoną) oraz w kilku zdaniach uzasadnia swój projekt.

- Ocena bardzo dobra (5): Zastosowanie wiedzy w sytuacjach nietypowych.
To ocena za umiejętne wykorzystanie wiedzy do rozwiązywania nowych problemów.
Przykład: Uczeń tworzy logo całkowicie od podstaw, używając tylko prostych kształtów i linii (bez gotowych ikon). Projekt jest oryginalny i pomysłowy.
- Ocena dobra (4): Zastosowanie wiedzy w sytuacjach typowych.
To ocena za praktyczne wykorzystanie wiedzy w zadaniach podobnych do tych z lekcji.
Przykład: Uczeń nie korzysta z gotowego szablonu, ale samodzielnie komponuje logo, łącząc co najmniej 3 gotowe elementy (np. tło, ikonę i tekst). Projekt jest estetyczny i czytelny.
- Ocena dostateczna (3): Zrozumienie wiadomości.
To ocena za zrozumienie podstawowych zasad i pojęć.
Przykład: Uczeń wybiera gotowy szablon logo, który pasuje do tematu, a następnie potrafi zmienić w nim tekst, kolory i rodzaj czcionki.
- Ocena dopuszczająca (2): Zapamiętanie wiadomości.
To ocena za opanowanie najważniejszych faktów, niezbędnych do dalszej nauki.
Przykład: Uczeń potrafi uruchomić program, wyszukać gotowy szablon logo i zmienić w nim tylko tekst. Wie, jak zapisać projekt.

Formy Sprawdzania Wiedzy i Umiejętności

Jak sprawdzane są umiejętności ucznia?

- **Sprawdziany i testy:** Będą zawsze zapowiedziane z co najmniej tygodniowym wyprzedzeniem. W jednym dniu uczeń może mieć tylko jeden taki sprawdzian, a w tygodniu – maksymalnie trzy (Zespół Szkolno-Przedszkolny w Ciasnej, 2024, § 20, ust. 7, pkt. 1-2).
- **Kartkówki:** Mogą odbyć się bez zapowiedzi i obejmują materiał z trzech ostatnich lekcji. Mogą mieć formę testu, quizu lub krótkiego zadania praktycznego (Zespół Szkolno-Przedszkolny w Ciasnej, 2024, § 20, ust. 6, pkt. 4).
- **Prace praktyczne:** To projekty, zadania wykonane przy komputerze, grafiki czy programy. Są one kluczowym elementem oceny na informatyce.
- **Aktywność na lekcji:** Zaangażowanie ucznia, jego pomysły i pomoc kolegom również mogą być docenione i nagrodzone oceną.

Nieprzygotowanie i poprawa ocen

- **Nieprzygotowanie (NP):** Uczeń ma prawo do jednego nieprzygotowania w semestrze. Należy je zgłosić na samym początku lekcji. Nie dotyczy to zapowiedzianych sprawdzianów (Zespół Szkolno-Przedszkolny w Ciasnej, 2024, § 20, ust. 6, pkt. 6).

- **Braki w wiedzy:** Obowiązkiem ucznia jest systematyczna nauka. Jeśli pojawią się trudności, prosimy o zgłoszenie tego faktu – szkoła oferuje pomoc (Zespół Szkolno-Przedszkolny w Ciasnej, 2024, § 22, ust. 1, pkt. 1, 5 oraz § 20, ust. 8, pkt. 2).
- **Poprawa ocen:** Uczeń ma prawo do jednokrotnej poprawy oceny niedostatecznej lub dopuszczającej z testu lub sprawdzianu w terminie i formie ustalonej z nauczycielem (Zespół Szkolno-Przedszkolny w Ciasnej, 2024, § 20, ust. 8, pkt. 4).

Warunki uzyskania oceny celującej

Aby uczeń otrzymał na semestr lub koniec roku ocenę **celującą**, musi wziąć udział przynajmniej raz w konkursie związanym z informatyką, posiadać oceny bardzo dobre z pozostałych aktywności, a także wykazać się przynajmniej jedną aktywnością w semestrze na ocenę celującą (zadaniem dodatkowym). Jeśli uczeń nie brał udziału w żadnym konkursie, musi w semestrze wykonać 3 dodatkowe zadania na ocenę celującą i posiadać oceny bardzo dobre. Uczeń może otrzymać ocenę celującą na koniec roku, jeśli przynajmniej w jednym semestrze otrzymał ocenę celującą, a w drugim ma ocenę bardzo dobrą.

Warunki otrzymania oceny za zadanie

Aby otrzymać ocenę, uczeń ma obowiązek zapisać pracę we właściwy sposób na komputerze, a następnie poinformować o jej wykonaniu na klasowym Padlecie, wpisując umówiony komentarz. Prace niewykonane lub nieprzesłane do nauczyciela w ciągu miesiąca od ich realizacji będą ocenione na ocenę niedostateczną. Uczeń nieobecny przez dłuższy czas ma możliwość poproszenia o zadanie zastępcze.

Szczegółowe Wymagania Edukacyjne

Numer i temat lekcji	Wymagania na ocenę dopuszczającą (2)	Wymagania na ocenę dostateczną (3)	Wymagania na ocenę dobrą (4)	Wymagania na ocenę bardzo dobrą (5)	Wymagania na ocenę celującą (6)
Blok: W świecie kodu i logiki					
1. Nowy etap...	Uczeń potrafi podać, jakie zagadnienia będą realizowane na lekcjach.	Uczeń potrafi przygotować swoje środowisko pracy (zalogować	Uczeń potrafi zaplanować swoją naukę, wyznaczając proste	Uczeń rozumie i potrafi omówić zasady bezpieczeństwa,	Uczeń aktywnie uczestniczy w ustalaniu zasad współpracy w zespole

		się, uporządko wać pliki).	cele.	netykiety i praw autorskich.	projektowy m i proponuje własne rozwiązania .
2. Co wiem, czego chcę...	Uczeń potrafi podać, czym są dane wejściowe i wyjściowe w prostym problemie.	Uczeń potrafi rozbić problem na mniejsze kroki.	Uczeń potrafi wskazać potrzebne wejścia, oczekiwane wyjścia i kryteria sukcesu.	Uczeń samodzielni e tworzy specyfikacj ę dla zadanego problemu.	Uczeń tworzy specyfikacj ę dla złożonego, wieloetapo wego problemu, uwzględniaj ąc możliwe przypadki brzegowe.
3. Krok po kroku do celu...	Uczeń potrafi podać, że algorytm to precyzyjny opis kroków.	Uczeń potrafi zapisać prosty algorytm w języku naturalnym.	Uczeń potrafi użyć w opisie prostych warunków i powtórzeń.	Uczeń potrafi zweryfikow ać poprawnoś ć algorytmu, przeprowad zając go "na sucho".	Uczeń tworzy algorytm rozwiązania nietypoweg o problemu, a następnie optymalizuj e go, zmniejszają c liczbę kroków.
4. Jak myśli komputer?.. .	Uczeń potrafi podać, że komputer używa zer i jedynek (bitów).	Uczeń potrafi rozróżnić jednostki pamięci (bit, bajt, kB, MB, GB).	Uczeń potrafi wyjaśnić, dlaczego zapis binarny jest używany w komputerac h.	Uczeń potrafi podać przykłady danych (tekst, obraz, dźwięk) i wyjaśnić, że	Uczeń tworzy prezentację lub infografikę wyjaśniając ą, jak różne typy danych są

				są one reprezentowane binarnie.	kodowane w systemie binarnym.
5. Liczymy jak komputer...	Uczeń potrafi podać różnicę między zapisem dziesiętnym a binarnym.	Uczeń potrafi zamienić małą liczbę z dziesiętnej na binarną.	Uczeń potrafi zamienić liczbę z binarnej na dziesiętną.	Uczeń potrafi sprawdzić poprawność konwersji w obie strony.	Uczeń pisze w Pythonie program, który dokonuje konwersji liczb między systemem dziesiętnym a binarnym.
6. Alfabet komputera..	Uczeń potrafi podać, że komputer zapisuje litery jako liczby.	Uczeń potrafi wyjaśnić, czym jest kod ASCII.	Uczeń potrafi wskazać różnicę między ASCII a Unicode.	Uczeń potrafi sprawdzić, jak polskie znaki diakrytyczne są reprezentowane w Unicode.	Uczeń tworzy program, który szyfruje i deszyfruje tekst, operując na kodach ASCII znaków (szyfr Cezara).
7. Hello, World!...	Uczeń potrafi uruchomić środowisko programistyczne Pythona.	Uczeń potrafi wpisać i uruchomić prosty program wyświetlający napis.	Uczeń potrafi porównać programowanie blokowe z pisanem kodu.	Uczeń samodzielnie pisze i uruchamia program "Hello, World!" oraz modyfikuje jego treść.	Uczeń eksperymentuje z funkcją print(), wyświetlając wieloliniowy tekst lub proste grafiki

					ASCII.
8. Z kodu na ekran...	Uczeń potrafi użyć funkcji print() do wyświetlenia napisu.	Uczeń potrafi wyświetlić na ekranie własną wiadomość.	Uczeń potrafi połączyć tekst i liczby w jednej instrukcji wyjścia.	Uczeń potrafi uruchomić kilka instrukcji print() w określonej kolejności, aby stworzyć spójny komunikat.	Uczeń tworzy program, który generuje wizytówkę lub prosty rachunek, formatując dane wyjściowe za pomocą kilku instrukcji print().
9. Pudełka na dane...	Uczeń potrafi podać, że zmienne służą do przechowywania danych.	Uczeń potrafi zadeklarować zmienną i nadać jej wartość.	Uczeń potrafi zmieniać wartość zmiennej w trakcie działania programu.	Uczeń potrafi używać zmiennych w obliczeniach i instrukcjach print().	Uczeń tworzy program, który wykonuje serię operacji na zmiennych, a na końcu prezentuje wynik końcowy.
10. Komputer pyta...	Uczeń potrafi użyć funkcji input() do zadania pytania.	Uczeń potrafi zapisać odpowiedź użytkownika w zmiennej.	Uczeń potrafi wykorzystać dane od użytkownika w dalszej części programu.	Uczeń tworzy prosty, interaktywny program, który prowadzi dialog z użytkownikami.	Uczeń tworzy program typu "Mad Libs", w którym użytkownik podaje różne słowa, a program

					wstawia je do gotowej historii.
11. Kalkulator w programie...	Uczeń potrafi zapisać w kodzie podstawowe działania matematyczne.	Uczeń potrafi wczytać liczby od użytkownika i zamienić je na typ liczbowy.	Uczeń potrafi wykonać obliczenia i zapisać wynik w zmiennej.	Uczeń potrafi wyświetlić wynik w czytelny sposób, z opisem.	Uczeń tworzy program-kalkulator, który pozwala użytkownikowi wybrać rodzaj operacji (dodawanie, odejmowanie, mnożenie, dzielenie).
12. Prawda czy fałsz?...	Uczeń potrafi podać, czym są wartości True/False.	Uczeń potrafi używać operatorów porównania (np. ==, >, <).	Uczeń potrafi łączyć warunki operatorami i and, or, not.	Uczeń potrafi napisać złożone wyrażenie logiczne i sprawdzić jego wynik.	Uczeń tworzy program, który symuluje bramki logiczne (AND, OR, NOT), pobierając dane od użytkownika.
13. Gdy warunek jest...	Uczeń potrafi wyjaśnić, jak działa instrukcja if.	Uczeń potrafi napisać prostą instrukcję if w programie.	Uczeń potrafi poprawnie zastosować wcięcia w bloku if.	Uczeń potrafi użyć zmiennych i porównań w warunku if.	Uczeń tworzy program, który na podstawie kilku warunków if

					podejmuje decyzję (np. prosty system rekomendacji).
14. Dwie drogi...	Uczeń potrafi wyjaśnić, jak działa instrukcja if-else.	Uczeń potrafi zapisać prosty warunek w programie.	Uczeń potrafi stworzyć program, który wykonuje jedną czynność, gdy warunek jest spełniony, a inną - gdy nie jest.	Uczeń potrafi testować różne wartości i sprawdzać, jak działa instrukcja if-else.	Uczeń tworzy program, który na podstawie złożonego warunku i instrukcji if-else podejmuje decyzję.
15. Powtarzaj, dopóki...	Uczeń potrafi wyjaśnić, jak działa pętla while.	Uczeń potrafi napisać prostą pętlę while z licznikiem.	Uczeń potrafi aktualizować zmienne, aby uniknąć pętli nieskończonej.	Uczeń potrafi użyć break/continue do kontroli przebiegu pętli.	Uczeń tworzy grę w zgadywanie liczby, w której pętla while działa, dopóki użytkownik nie odgadnie prawidłowej wartości.
16. Policzmy to razem...	Uczeń potrafi wyjaśnić, jak działa pętla for z	Uczeń potrafi napisać prostą pętlę, która	Uczeń potrafi użyć pętli for do wypisania elementów	Uczeń potrafi łączyć pętlę for z innymi instrukcjami	Uczeń tworzy program, który za pomocą

	range()).	powtarza czynność określoną liczbę razy.	listy.	(warunki, obliczenia).	pętli for przetwarza dane w liście (np. oblicza sumę, znajduje największy element).
17. Matematyczny detektyw...	Uczeń potrafi użyć operatora % do sprawdzenia podzielności.	Uczeń potrafi napisać program, który sprawdza, czy liczba jest podzielna przez inną.	Uczeń potrafi połączyć warunek z pętlą, by sprawdzać wiele liczb.	Uczeń potrafi wyświetlić odpowiedni komunikat w zależności od wyniku.	Uczeń pisze program, który znajduje wszystkie dzielniki podanej liczby.
18. Największy wspólny dzielnik...	Uczeń potrafi podać, czym jest NWD.	Uczeń potrafi opisać algorytm Euklidesa krok po kroku.	Uczeń potrafi zapisać algorytm Euklidesa w Pythonie, używając pętli.	Uczeń potrafi przetestować program na różnych przykładach.	Uczeń implementuje dwie wersje algorytmu Euklidesa (z odejmowaniem i z resztą z dzielenia) i porównuje ich działanie.
19. Jak naprawić program?...	Uczeń potrafi podać, że debugowanie to poprawianie błędów.	Uczeń potrafi rozpoznać błąd składniowy.	Uczeń potrafi uruchamiać program krok po kroku i obserwować	Uczeń potrafi samodzielnie zidentyfikować i poprawić	Uczeń świadomie używa narzędzi debuggera (np. punkty przerwania,

			ć jego działanie.	prosty błąd logiczny.	śledzenie zmiennych) do analizy złożonego programu.
20. Robot sterowany kodem...	Uczeń potrafi uruchomić środowisko do pisania programu dla robota.	Uczeń potrafi napisać prosty program sterujący ruchem lub światłami robota.	Uczeń potrafi przetestować kod i sprawdzić reakcję robota.	Uczeń potrafi porównać programowanie tekstowe z blokowym, wskazując na wady i zalety obu.	Uczeń pisze program, który realizuje złożone zadanie z użyciem czujników robota, np. jazda po linii do momentu wykrycia przeszkody.
Blok: Arkusz kalkulacyjny					
21. Magia w arkuszu...	Uczeń potrafi podać, że arkusz obsługuje teksty i daty.	Uczeń potrafi połączyć teksty z dwóch komórek w jedną.	Uczeń potrafi użyć funkcji do pracy z datami (np. obliczyć różnicę dni).	Uczeń potrafi użyć funkcji do dzielenia tekstu na części (np. imię i nazwisko).	Uczeń tworzy w arkuszu bazę danych (np. lista kontaktów) i wykorzystuje funkcje tekstowe oraz daty do jej przetwarzania i analizy.

22. Nie tylko liczby...	Uczeń potrafi utworzyć wykres w arkuszu.	Uczeń potrafi sformatować wykres (zmienić tytuł, kolory, opisy).	Uczeń potrafi dopasować typ wykresu do rodzaju danych.	Uczeń tworzy estetyczny i czytelny wykres, który poprawnie reprezentuje dane.	Uczeń tworzy dynamiczny wykres, który aktualizuje się automatycznie po zmianie danych w tabeli.
23. Szkolny superpomocnik...	Uczeń potrafi stworzyć prostą tabelę i policzyć sumę.	Uczeń potrafi używać arkusza do rozwiązywania prostych zadań z innych przedmiotów.	Uczeń potrafi przedstawić dane w formie tabeli i wykresu w jednym projekcie.	Uczeń samodzielnie znajduje zastosowanie dla arkusza kalkulacyjnego w rozwiązywaniu problemu szkolnego.	Uczeń tworzy złożony projekt interdyscyplinarny (np. analiza danych historycznych, symulacja eksperymentu fizycznego) w arkuszu kalkulacyjnym.
Blok: Projektowanie i publikacja					
24. Szkolny reporter...	Uczeń potrafi zaplanować prosty tekst (wstęp, rozwinięcie,	Uczeń potrafi napisać krótki artykuł lub relację,	Uczeń potrafi dodać do tekstu zdjęcie lub grafikę.	Uczeń tworzy estetyczny i gotowy do publikacji artykuł.	Uczeń przeprowadza wywiad, redaguje go i przygotowuje

	zakończeni e).	dbając o poprawność. ć.			je do publikacji w formie artykułu z ilustracjami.
25. Projektujemy w Google Docs...	Uczeń potrafi podać, jakie elementy powinien zawierać plakat.	Uczeń potrafi ustawić odpowiedni układ strony i wstawić obrazy.	Uczeń potrafi wstawić i sformatowa ć ozdobne czcionki (np. WordArt).	Uczeń dba o czytelność i estetykę plakatu (kontrast, układ).	Uczeń tworzy serię plakatów na kampanię społeczną, zachowując spójną identyfikacj ę wizualną.
26. Porządkujemy dokument...	Uczeń potrafi zastosować style nagłówków.	Uczeń potrafi automatycz nie utworzyć spis treści.	Uczeń potrafi wstawić przypis dolny.	Uczeń potrafi zaktualizow ać spis treści po zmianach w dokumencie.	Uczeń tworzy szablon dokumentu z predefiniow anymi stylami, nagłówkiem , stopką i automatycz nym spisem treści.
27. Moja pierwsza strona...	Uczeń potrafi podać, że HTML to język do tworzenia stron.	Uczeń potrafi napisać prosty dokument HTML z podstawow ą strukturą.	Uczeń potrafi używać znaczników do tworzenia nagłówków, akapitów i list.	Uczeń potrafi wstawić na stronę link i obraz.	Uczeń tworzy prostą, wielostronic ową witrynę WWW, połączoną hiperłączami.
28. Od	Uczeń	Uczeń	Uczeń	Uczeń	Uczeń

pomysłu do strony...	potrafi zaplanować prostą stronę.	potrafi stworzyć plik HTML i zapisać w nim strukturę strony.	potrafi połączyć różne elementy (nagłówki, akapity, listy, linki, obrazy) w jednym projekcie.	potrafi sprawdzić stronę w przeglądarce i poprawić błędy w kodzie.	publikuje swoją prostą stronę internetową w sieci (np. za pomocą GitHub Pages).
29. Kolory i style na stronie...	Uczeń potrafi podać, że CSS służy do zmiany wyglądu strony.	Uczeń potrafi zmienić kolor i rozmiar tekstu za pomocą stylów.	Uczeń potrafi ustawić kolor tła strony lub elementu.	Uczeń potrafi połączyć plik HTML z zewnętrznym arkuszem stylów CSS.	Uczeń tworzy spójny wizualnie projekt strony, definiując style dla różnych elementów (nagłówki, linki, akapity) w pliku CSS.
30. Od prostego kodu do...	Uczeń potrafi zastosować style, aby wyróżnić nagłówki i akapity.	Uczeń potrafi zadbać o spójne kolory i czcionki na stronie.	Uczeń potrafi poprawić kod, jeśli strona wygląda inaczej, niż planował.	Uczeń tworzy estetyczną i czytelną stronę internetową, świadomie stosując zasady HTML i CSS.	Uczeń tworzy responsywną stronę internetową, która poprawnie wyświetla się na urządzeniach o różnej wielkości ekranu.
Blok: Sztuczna					

inteligencja					
31. Czym jest sztuczna inteligencja. ..	Uczeń potrafi podać, czym jest AI i gdzie ją spotyka na co dzień.	Uczeń potrafi wskazać korzyści płynące z korzystania z AI.	Uczeń potrafi wymienić zagrożenia i ograniczenia związane z AI.	Uczeń rozumie, że AI to narzędzie, które trzeba stosować odpowiedzialnie.	Uczeń przeprowadza debatę na temat etycznych aspektów wykorzystania sztucznej inteligencji w wybranej dziedzinie.
32. AI jako twórcze narzędzie...	Uczeń potrafi wykorzystać proste narzędzie AI do stworzenia grafiki lub tekstu.	Uczeń potrafi poprawić i dostosować efekty pracy AI do własnych potrzeb.	Uczeń potrafi odróżnić własną twórczość od treści wygenerowanej przez AI.	Uczeń świadomie i krytycznie korzysta z narzędzi AI, rozumiejąc ich możliwości i ograniczenia.	Uczeń realizuje projekt, w którym kreatywnie łączy własną pracę z treściami wygenerowanymi przez różne narzędzia AI, dokumentując proces.
Blok: Bezpieczeństwo cyfrowe					
33. Bezpieczny, uczciwy...	Uczeń potrafi podać podstawowe	Uczeń potrafi stosować zasady	Uczeń potrafi wyjaśnić, czym jest	Uczeń potrafi rozpoznać zachowania	Uczeń tworzy poradnik lub

	e zagrożenia w internecie.	bezpieczne go korzystania z sieci (hasła, ostrożność) .	prawo autorskie.	nieetyczne w internecie i wskazać właściwe sposoby reagowania .	prezentację na temat cyberbezpieczeństwa i etyki online, opierając się na analizie rzeczywistych przypadków.
--	-------------------------------------	---	---------------------	--	--