

Przedmiotowy system oceniania- informatyka

Prowadzący zajęcia: Krzysztof Kadowski

Podstawa prawna:

- rozporządzenie z dn. 26 lutego 2019r poz. 373 z pz. w sprawie oceniania, klasyfikowania i promowania uczniów i słuchaczy szkół publicznych;
- WSO w Statucie Szkoły;
- Podstawa programowa z języka polskiego.

Cel oceniania:

Głównym celem oceniania jest monitorowanie rozwoju ucznia, czyli określanie jego postępów w stosunku do standardów osiągnięć, diagnozowanie jego rozwoju, rozpoznawanie indywidualnych uzdolnień, zainteresowań, predyspozycji a także trudności. Ocenianie wspiera i wzmacnia rozwój ucznia.

Ocenianie wewnątrzszkolne ma na celu:

- poinformowanie ucznia o poziomie jego osiągnięć edukacyjnych i postępach w tym zakresie, trudnościach i specjalnych uzdolnieniach,
- udzielanie uczniowi pomocy w nauce poprzez przekazywanie informacji o tym, co zrobił dobrze i jak powinien się dalej uczyć,
- pomoc uczniowi w samodzielnym planowaniu swojego rozwoju,
- motywowanie ucznia do systematycznej pracy, dalszych postępów w nauce i zachowaniu,
- rozwijanie poczucia samodzielności i odpowiedzialności za postępy w dziedzinie edukacji szkolnej,
- dostarczenie rodzicom (prawnym opiekunom) i nauczycielom informacji o postępach, osiągnięciach, trudnościach i specjalnych uzdolnieniach ucznia,
- dostarczenie nauczycielom informacji o stopniu opanowania przez uczniów wiadomości i umiejętności oraz poziomie realizacji założonych celów,
- umożliwienie nauczycielom doskonalenia organizacji i metod pracy dydaktyczno-wychowawczej.

Metody sprawdzania osiągnięć ucznia.

1. Ocenie podlegają różne formy aktywności ucznia:

- wypowiedzi ustne,
- pisemne prace klasowe (wypracowania, sprawdziany, testy, kartkówki),
- próbne egzaminy maturalne,

- badanie umiejętności rozumienia czytanego tekstu,
- praca w grupie i praca indywidualna na lekcji,
- zadania dodatkowe, np. projekty, prezentacje, itp.,
- aktywność na lekcji.

2. Dłuższą pracę pisemną nauczyciel zapowiada tydzień wcześniej, ze wskazaniem zakresu obowiązującego materiału.

3. Kartkówka obejmuje mniejszy zakres materiału (np. dotyczący trzech tematów, istotnego zagadnienia, problemu); może być niezapowiedziana, jeśli dotyczy trzech ostatnich tematów lub innych zagadnień wskazanych wcześniej przez nauczyciela;

Zasady ogólne

- Na pierwszej lekcji danego roku szkolnego lub semestru nauczyciel informuje uczniów o wymaganiach edukacyjnych odnośnie do realizowanego przez siebie programu nauczania oraz przedstawia uczniom zasady Przedmiotowego Systemu Oceniania.
- Ocenianie dokonywane jest według sześciostopniowej skali ocen. W ciągu semestru każdy uczeń otrzymuje co najmniej jedną ocenę w miesiącu.
- W przypadku, gdy uczeń z powodu nieobecności nie uzyska wymaganej liczby ocen, nauczyciel ma prawo zadać pracę dodatkową po ustaleniu zakresu, tematyki i formy pracy.
- Ocena śródroczna i roczna nie jest średnią arytmetyczną ocen częściowych. Ocena roczna stanowi podsumowanie rocznej pracy ucznia – wystawiana jest na podstawie ocen uzyskanych w ciągu całego roku szkolnego (nauczyciel bierze pod uwagę oceny z obu okresów).
- Ocena jest jawna, umotywowana na każdą prośbę ucznia lub jego rodziców. Ocena semestralna uwzględnia wszystkie oceny bieżące, przy czym największą wagę mają oceny z prac pisemnych, prac kontrolnych i odpowiedzi ustnych.
- Sprawdziany i inne prace pisemne przechowywane są w szkole do końca roku szkolnego. Sprawdzone i ocenione sprawdziany i kartkówki uczeń otrzymuje do wglądu podczas zajęć lekcyjnych na własną prośbę.
- Jeżeli uczeń nie uczestniczył w pracy pisemnej, to ma obowiązek zaliczenia jej w ciągu 2 tygodni, w uzgodnionym z nauczycielem terminie i formie. Po upływie tego terminu nauczyciel wpisuje ocenę niedostateczną.
- Nauczyciel ocenia prace pisemne w terminie do 14 dni (wyjątek stanowią przypadki losowe, np. choroba nauczyciela). Uczeń ma prawo zgłosić raz w semestrze nieprzygotowanie, z uzasadnionych przyczyn losowych, do zajęć, które sygnalizowane jest na początku lekcji raz w półroczu (nie dotyczy to lekcji, na której ma odbyć się zapowiedziany wcześniej sprawdzian lub praca klasowa).
- Uczeń ma prawo poprawić ocenę niedostateczną w ciągu 14 od jej otrzymania na zasadach ustalonych indywidualnie przez nauczyciela.
- Jeżeli uczeń nie poinformował nauczyciela o nie odrobieniu zadania domowego na początku lekcji, to otrzymuje ocenę niedostateczną, gdy ten fakt zostanie ujawniony przez nauczyciela w trakcie trwania zajęć.
- Uczeń otrzymuje ocenę niedostateczną, jeżeli podczas pracy pisemnej ściąga, odpisuje lub zmienia samodzielnie grupę.

Ocenie podlegają wszystkie formy aktywności ucznia:

- Sprawdziany, prace klasowe – po każdym dziale tematycznym lub module, zapowiedziane z tygodniowym wyprzedzeniem z jednoczesnym wpisaniem stosownej adnotacji do dziennika.
- Oceniane są według następującej skali punktowej:
 - 0% - 39% - ocena niedostateczna
 - 40% - 50% - ocena dopuszczająca
 - 51% - 68% - ocena dostateczna
 - 69% - 74% - ocena dostateczna +
 - 75% - 84% - ocena dobra
 - 85% - 89% - ocena dobra +
 - 90% - 98% - ocena bardzo dobra
 - 99% - 100% - ocena celująca
- Kartkówki – w ciągu semestru, mogą być niezapowiedziane. Obejmują materiał z trzech ostatnich tematów, mogą dotyczyć zadania domowego.
- Odpowiedzi ustne – sprawdzające bieżącą wiedzę ucznia, obejmują materiał z trzech ostatnich tematów.
- Zadania domowe – odrobione samodzielnie, złożone w formie i czasie uzgodnionym z nauczycielem.
- Praca w grupach, aktywność na lekcjach – wcześniej zapowiadana i przygotowana lub wynikająca z przebiegu lekcji.

Dla ocen bieżących ustalono następujące wagi:

sprawdzian – waga 5, poprawa sprawdzianu – waga 4,
ćwiczenia praktyczne – waga 4, poprawa ćwiczeń – waga 3,
kartkówka – waga 3, poprawa kartkówki – waga 2,
odpowiedź – waga 3
aktywność – waga 2
praca domowa – waga 1

Oceny klasyfikacyjne śródroczne i roczne ustala się w stopniach według skali:

6 – celujący
5 – bardzo dobry
4 – dobry
3 – dostateczny
2 – dopuszczający
1 – niedostateczny

O ocenie semestralnej decyduje średnia ważona ocen bieżących w danym semestrze. O ocenie na koniec roku decyduje średnia ważona ocen z pierwszego i drugiego semestru, nie jest to średnia arytmetyczna ocen w semestrze.

Ocena celująca (wymagania wykraczające) - Uczeń opanował pełny zakres wiedzy i umiejętności określony programem nauczania (99%÷100% powyższych kryteriów wymagań edukacyjnych):

- Uczeń nie tylko spełnia wymagania na ocenę bardzo dobrą, ale posiada wiedzę i umiejętności wykraczające poza ramy realizowanego programu nauczania.
- Wyróżnia się aktywnością w grupie, wykazuje szczególne zainteresowania przedmiotem i osiąga sukcesy w konkursach i olimpiadach przedmiotowych.

Ocena bardzo dobra (wymagania dopełniające) – Uczeń opanował pełny zakres wiedzy i umiejętności określony programem nauczania (ok. 90%÷98% powyższych kryteriów wymagań edukacyjnych).

Ocena dobra (wymagania rozszerzone) – Uczeń opanował pełny zakres wiedzy i umiejętności określony programem nauczania (ok. 75%÷84% powyższych kryteriów wymagań edukacyjnych).

Ocena dostateczna (wymagania podstawowe) – Uczeń opanował wiadomości i umiejętności umożliwiające zdobywanie dalszej wiedzy, dysponuje przeciętną wiedzą w zakresie materiału przewidzianego programem, w jego wiadomościach i umiejętnościach są braki (ok. 51%÷68% powyższych kryteriów wymagań edukacyjnych), czasami potrzebuje pomocy nauczyciela.

Ocena dopuszczająca (wymagania konieczne) – Uczeń opanował wiadomości i umiejętności niezbędne do kontynuowania dalszej nauki (ok. 40%÷50% powyższych kryteriów wymagań edukacyjnych), często potrzebuje pomocy nauczyciela.

Kryteria wymagań edukacyjnych

Uwaga!

W planie pominięto podstawowe umiejętności, które uczeń powinien już osiąść wcześniej, np. zachowywanie plików projektów, wczytywanie dokumentów do edycji i posługiwanie się systemem operacyjnym.

Kryteria danej oceny opracowano, zakładając, że zostały spełnione kryteria ocen niższych.

Temat	Ocena dopuszczająca. Uczeń:	Ocena dostateczna Uczeń:	Ocena dobra Uczeń:	Ocena bardzo dobra Uczeń:	Ocena celująca Uczeń:
I. Algorytmika i programowanie					
Więcej o liczbach, czyli logarytm, sito Eratostenesa i schemat Hornera.	– zna definicję logarytmu – zna ogólną ideę algorytmu sita Eratostenesa (wykreślanie wielokrotności kolejnych liczb pierwszych) – potrafi zapisać naiwny („pierwiastkowy”) algorytm sprawdzania pierwszości	– zna przykłady sytuacji, w których używa się logarytmu – potrafi prześledzić działanie algorytmu sita Eratostenesa na konkretnych danych	– potrafi rozpoznać złożoność logarytmiczną w algorytmach – potrafi zapisać w pseudojęzyku schemat Hornera	– potrafi zaimplementować sito Eratostenesa i schemat Hornera w wybranym języku programowania – zna złożoność algorytmu sita Eratostenesa i porównuje ją z „pierwiastkowym” algorytmem	–
Rekursja raz jeszcze, czyli dziel i zwyciężaj.	– potrafi zasymulować (prześledzić działanie) prostej funkcji rekurencyjnej	– zna warunki działania rekursji, potrafi przekształcać proste algorytmy (z jednym wywołaniem rekurencyjnym) z iteracyjnych na rekurencyjne i odwrotnie	– prawidłowo analizuje algorytmy rekurencyjne typu „dziel i zwyciężaj”, zna i umie stosować warunki działania rekursji w projektowaniu algorytmów	– potrafi samodzielnie zaprojektować algorytm opierający się na rekursji z dwoma wywołaniami typu „dziel i zwyciężaj”	– potrafi zanalizować nietypowe wywołania rekurencyjne (np. na więcej niż dwóch podzadaniach albo na różniących się od siebie podzadaniach)
Sortowanie przez	– zna ogólną ideę	– zna ogólną ideę	– potrafi zapisać cały	– potrafi zapisać	– potrafi

scalanie, czyli pierwszy efektywny algorytm sortowania.	algorytmu sortowania przez scalanie (podzielić na dwie połowy, rekursja, scalenie)	scalania, potrafi podać złożoność algorytmu ($n \log n$)	algorytm sortowania przez scalanie w pseudojęzyku	algorytm sortowania przez scalanie w wybranym języku programowania bez błędów i pilnując przypadków brzegowych	uargumentować złożoność algorytmu sortowania przez scalanie
Sortowanie szybkie, czyli jak się sortuje w praktyce.	– zna ogólną ideę algorytmu szybkiego sortowania (przestawić elementy, a potem wywołanie rekurencyjne)	– wie, czym jest klucz i zna ideę operacji przedstawiania, zna złożoność algorytmu w średnim wypadku	– potrafi zapisać cały algorytm sortowania przez scalanie w pseudojęzyku	– potrafi zapisać algorytm sortowania przez scalanie w wybranym języku programowania, rozumie korzyść z losowania klucza	– potrafi przedyskutować i uargumentować złożoność algorytmu przy różnych wyborach klucza (najlepszym/najgorszym)
Specjalne elementy w tablicy, czyli największy, najmniejszy i lider.	– potrafi znaleźć oddzielnie element najmniejszy i największy – wie, czym są lider i dominanta, potrafi podać przynajmniej jeden algorytm znajdowania dominanty w zbiorze	– potrafi prześledzić działanie (przeanalizować) algorytm jednoczesnego znajdowania najmniejszego i największego elementu oraz algorytm znajdowania lidera	– potrafi zapisać w pseudojęzyku algorytm jednoczesnego znajdowania najmniejszego i największego elementu oraz algorytm znajdowania lidera i podać ich złożoność	– potrafi uargumentować liczbę porównań w algorytmie jednoczesnego znajdowania największego/najmniejszego elementu oraz poprawność algorytmu znajdowania lidera	–
Tablice dynamiczne i listy wiązane, czyli kiedy tablica nie wystarcza	– zna ogólną ideę tablicy dynamicznej (tablica + zmienny rozmiar) oraz listy wiązanej	– opisuje operacje dostępne na tablicy dynamicznej i liście wiązanej, potrafi przedstawić na rysunku budowę i działanie listy wiązanej	– potrafi przeanalizować prosty algorytm, w którym występują operacje na tablicach dynamicznych/listach wiązanych, opisuje rozwiązanie problemu	– swobodnie analizuje i projektuje algorytmy zawierające tablice dynamiczne i listy wiązane	– stosuje tablice dynamiczne i listy wiązane (np. <code>std::vector</code> , <code>std::list</code> , listy w Pythonie) w swoich programach

			Flawiusza za pomocą listy wiązanej		
Stos i kolejka, czyli struktury proste i bardzo użyteczne.	– zna ogólną ideę stosu i kolejki	– opisuje operacje na stosie i kolejce	– analizuje proste algorytmy, w których występują struktury stosu i kolejki, aplikuje strukturę stosu do rozwiązania problemu ONP	– swobodnie projektuje analizuje algorytmy, w których występują struktury stosu i kolejki	– posługuje się typem stos/kolejka w wybranym języku programowania w swoich programach
Grafy, czyli co mają wspólnego sieć społecznościowa i paryskie metro	– potrafi rozpoznać graf w sytuacjach praktycznych	– podaje przykłady konkretnych sytuacji modelowanych za pomocą grafu, zna definicję grafu	– zna implementację grafu przez macierze i listy sąsiedztwa	– potrafi analizować i projektować algorytmy, w których występują grafy – potrafi porównać implementację macierzową i listową, podając ich zalety i wady	– potrafi samodzielnie zaimplementować graf w wybranym języku programowania
Wyszukiwanie idola, czyli o tym, jak spóźniliśmy się na turniej tenisowy	– wie, czym jest element-idol, potrafi znaleźć element-idol dowolnym algorytmem (np. pytając o wszystkie pary)	– zna ogólną ideę algorytmu wyszukiwania idola („pytaj jeden element do momentu przegranej”)	– potrafi zapisać w pseudojęzyku algorytm wyszukiwania idola	– potrafi uargumentować złożoność algorytmu wyszukiwania idola	–
Najkrótsze ścieżki w grafie, czyli jak komputery znajdują drogę	– potrafi podać, czym jest ścieżka w grafie i pokazać jej długość na przykładzie	– zna ogólną ideę algorytmu wyszukiwania ścieżki („najpierw elementy o odległości 1, potem 2, i tak dalej”)	– potrafi zapisać w pseudojęzyku szkic algorytmu BFS	– potrafi zapisać w pseudojęzyku pełny algorytm BFS z zastosowaniem kolejki i prosto uzasadnić jego złożoność	– implementuje działający algorytm BFS w wybranym języku programowania
Programowanie- strukturalne kontra-	–rozumie ideę- programowania-	–rozumie pojęcie- klasy/obiektu oraz-	–wyjaśnia różnicę- między-	–wyjaśnia bardziej- zaawansowane-	–potrafi stosować w- praktyce paradygmat-

obiektywne, czyli dane w centrum uwagi	obiektywne („funkcje przypisane do danych”)	pola i metody	programowaniem strukturalnym i obiektywnym, rozumie sytuacje, w których programowanie obiektywne jest koniecznością	pojęcia: <i>dziedziczenie</i> , <i>polimorfizm</i>	obiektywne (czyli samodzielnie pisze obiektywne programy)
Jak rozwiązywać zadania programistyczne, czyli podstawowe porady na proste kody	– rozumie treść zadań programistycznych i umie przełożyć proste zadania na implementację w wybranym języku programowania	– świadomie stosuje zasady czytelnego pisania kodu (wcięcia, nazwy zmiennych etc.)	– stosuje dobre praktyki programistyczne (unikanie kopiuj-wklej, wydzielanie do podprocedur) – umie przetestować swój program, w tym napisać dla niego proste dane testowe	– pisze programy w sposób czytelny i zrozumiały dla innych – potrafi swobodnie testować program, wyszukiwać i poprawiać błędy	– rozwiązuje zadania na poziomie olimpijskim, np. na serwerach typu „online judge”

II. Arkusz kalkulacyjny

Podstawowe typy danych, czyli co możemy wpisać w komórkę arkusza	– uzasadnia wybranie odpowiedniego typu formatowania do podanych danych – rozróżnia sposoby adresowania	– potrafi używać podstawowych funkcji arkusza i stosować w nich różne typy adresowania – potrafi dostosować typ formatowania do podanych danych	– potrafi samodzielnie używać różnych typów adresowania	– samodzielnie potrafi dobrać rodzaje adresowania, tak aby formuła była jak najbardziej uniwersalna i używać zagnieżdżonych formuł	–
Trójkąt Sierpińskiego, czyli do czego przydaje się adresowanie względne i adresowanie bezwzględne oraz	– wie, jak wykorzystać podane przez nauczyciela odwzorowania do wygenerowania trójkąta Sierpińskiego	– potrafi samodzielnie wypełnić komórki podobnymi wartościami (liczby, daty)	– potrafi samodzielnie otworzyć pokazane ćwiczenie, rozumie, jak użyć podanego układu iterowanych rozwiązań	– potrafi samodzielnie generować inne fraktale niż te pokazane na lekcji (po podaniu mu wzoru na iterowane	– rozumie i potrafi wyjaśnić, na czym polega generowanie fraktali za pomocą układu iterowanych rozwiązań

wypełnienie serią danych				przekształcenia)	
Jak korzystać z gotowych funkcji, podstawowe funkcje tekstowe, czyli jak łatwo modyfikować dane tekstowe	– zna i potrafi użyć funkcji pokazanych na lekcji	– potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach zagnieżdżonych	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach zagnieżdżonych, tak aby te formuły były uniwersalne
Jak korzystać z gotowych funkcji, podstawowe funkcje dotyczące daty i czasu, czyli jak łatwo pracować z datą.	– zna i potrafi użyć funkcji pokazanych na lekcji	– potrafi przeczytać opis nowej funkcji i użyć jej w zadaniu	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach zagnieżdżonych	– potrafi samodzielnie używać podanych funkcji i używać ich w formułach zagnieżdżonych, tak aby te formuły były uniwersalne
Jak szukać w Excelu, czyli gdzie to jest	– potrafi podać, jakie funkcje w arkuszu pozwalają wyszukać, potrafi używać funkcji WYSZUKAJ na prostych przykładach	– odtwarza przykłady użycia funkcji WYSZUKAJ podane w podręczniku	– potrafi użyć funkcji WYSZUKAJ.PIONO WO dla przykładów podanych w podręczniku	– potrafi samodzielnie stosować funkcję WYSZUKAJ dla dowolnych przykładów podanych przez nauczyciela i potrafi sam wskazać takie przykłady	– potrafi samodzielnie stosować funkcję WYSZUKAJ.PIONO WO dla dowolnych przykładów i potrafi sam wskazać takie przykłady
Co ukrywa numer PESEL, czyli do czego przydaje się adres mieszany	– potrafi z podanego numeru PESEL odczytać datę urodzin i płeć	– potrafi samodzielnie napisać formułę, która policzy z podanego numeru PESEL datę urodzin	– samodzielnie potrafi napisać formułę, która wyznaczy płeć z numeru PESEL	– potrafi napisać formułę, która sprawdzi, czy numer PESEL jest prawidłowy	– samodzielnie potrafi napisać uniwersalne formuły wyznaczające datę urodzenia, płeć i sprawdzające poprawność dla wielu

					danych
Tabela przestawna, czyli wygodne grupowanie danych	– wie, kiedy stosować tabele przestawne, potrafi odczytywać informacje z tabeli przestawnej	– potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji	– samodzielnie potrafi utworzyć tabelę przestawną i policzyć, sumę, średnią, minimum, maksimum dla wyznaczonych grup	– potrafi samodzielnie utworzyć tabelę przestawną, zastosować w niej filtr i posortować dane według podanego klucza	– potrafi samodzielnie utworzyć tabele przestawną i dla grup tworzyć podgrupy oraz użyć funkcji liczących dla grup i podgrup
Wykresy, czyli jak atrakcyjnie przedstawiać dane	– potrafi odczytywać dane z wykresu – wie, jakie są podstawowe typy wykresu – potrafi dobrać odpowiedni rodzaj wykresu do podanych danych	– potrafi odtworzyć przykłady użycia tabeli przestawnej pokazane na lekcji	– samodzielnie potrafi utworzyć wykres i sformatować go według podanych wytycznych	– potrafi samodzielnie dobrać wykres do podanych danych i sformatować go tak, aby był czytelny – potrafi przedstawić kilka serii danych na wykresie	– potrafi utworzyć wykres z tabeli przestawnej i sformatować go, aby dane były czytelne
Pobieranie danych z pliku, czyli z notatnika do Excela.	– potrafi pobrać dane z pliku tekstowego zawierające tekst i liczby całkowite	– potrafi pobrać dane z pliku zawierające długie liczby i numer PESEL	– potrafi pobrać dane zawierające liczby rzeczywiste	– potrafi pobrać dane z pliku zawierające datę i czas	– potrafi pobrać dane z dowolnego pliku zawierającego tekst ze znakami polskimi