

Przedmiotowe Zasady Oceniania – Informatyka. Klasa 2 TI zakres rozszerzony szkoła ponadgimnazjalna.

I Zasady oceniania

1. Ocenianie pracy uczniów odbywa się na podstawie przeprowadzonych sprawdzianów praktycznych, kartkówek, odpowiedzi ustnych, prac domowych oraz aktywności uczniów na lekcji.
2. Sprawdziany są obowiązkowe.
3. Oceny ze sprawdzianów stanowią najważniejszą część składową oceny semestralnej (rocznej).
4. Sprawdziany są zapowiadane z co najmniej tygodniowym wyprzedzeniem.
5. Każdy uczeń ma prawo do poprawy jednej oceny w semestrze.
6. Uczeń przyłapany na ściąganiu otrzymuje ocenę niedostateczną i traci prawo do poprawy tej oceny.
7. W przypadku usprawiedliwionej nieobecności ucznia na zapowiedzianym sprawdzianie wiadomości, jest on zobowiązany do zaliczenia odpowiedniej partii materiału w terminie i formie ustalonej z nauczycielem.
8. Planowane powtórzenia materiału oraz pisemne sprawdziany wiadomości są przeprowadzane zgodnie z zasadami określonymi w wewnątrzszkolnym systemie oceniania.
9. Nauczyciel w terminie ustalonym w Wewnątrzszkolnym Systemie Oceniania informuje uczniów i rodziców o przewidywanych rocznych (semestralnych) ocenach. W przypadku, gdy uczeń, lub jego rodzice nie zgadzają się z przewidywaną oceną, a sprzeciw ma uzasadnienie w ocenach częściowych ucznia, uczeń ma prawo w terminie i formie ustalonej przez nauczyciela przystąpić do zaliczenia partii materiału objętego okresem klasyfikacji.
10. Uczeń może otrzymać ocenę dodatkową za udział w konkursach i projektach
11. W innych sprawach zastosowanie ma WSO.

Stopień dopuszczający (poziom wymagań K - konieczny)

Uczeń naprowadzony przez nauczyciela potrafi rozwiązywać proste (jednoetapowe) problemy (z zakresu określonego w planie wynikowym), które już wcześniej rozwiązywał, wykorzystując do tego celu poznane oprogramowanie. Potrafi zapisać wyniki swojej pracy na wskazanym nośniku.

Ocena dopuszczająca, to ocena dla ucznia słabego, który źle radzi sobie z pracą przy komputerze, nie potrafi samodzielnie wykonać ćwiczenia i nie w pełni rozumie zadanie przed nim postawione, który przy pomocy nauczyciela umie jednak zrealizować minimum ustalone dla danego ćwiczenia. W jego poczynaniach widać duże braki w zakresie wiedzy i umiejętności, ale podejmuje on próby zmierzenia się z zadaniem.

Stopień dostateczny (poziom wymagań P - podstawowy)

Uczeń potrafi dobrać podstawowe narzędzia programowe do rozwiązania prostego problemu z zakresu określonego w planie wynikowym. Samodzielnie rozwiązuje problemy realizowane w trakcie nauki, bazując na opanowanych podstawowych wiadomościach z technologii informacyjnej. W wykonywaniu zadania wkłada dużo pracy i wysiłku.

Ocenę dostateczną otrzymuje uczeń wykazujący braki w umiejętnościach i wiedzy, nadrabiający jednak pracowitością i chęcią wykonania ćwiczenia. Uczeń wykonuje swoją pracę poprawnie pod względem użycia funkcji programu, ale mało estetycznie i z błędami. Projekt pozostanie niewykończony. Uczeń stosuje jedynie podstawowe funkcje oprogramowania.

Stopień dobry (poziom wymagań R - rozszerzający)

Uczeń potrafi samodzielnie dobrać podstawowe narzędzia programowe do rozwiązania kilkietapowego problemu (wymagającego wykorzystania kilku narzędzi). Rozwiązuje problemy (określone w planie wynikowym) o tym samym stopniu trudności, co realizowane w ramach programu nauczania.

Ocena dobra jest oceną dla ucznia samodzielnie wykonującego ćwiczenia, którego prace zawierają drobne błędy, lecz są wykonane estetycznie. Uczeń wykazuje znajomość programu i jego średnio zaawansowanych funkcji. Stosuje klasyczne rozwiązania, wzorowane na istniejących projektach.

Stopień bardzo dobry (poziom wymagań D - dopełniający)

Uczeń wykorzystując znane narzędzia programowe, potrafi rozwiązać złożone problemy z zakresu określonego w planie wynikowym, z którymi do tej pory nie zetknął się na lekcji. Przygotowuje i prowadzi prelekcje oraz referaty związane z tematyką zajęć.

Ocenę bardzo dobrą stawiamy uczniowi biegle posługującemu się oprogramowaniem i urządzeniami peryferyjnymi, dobrze dobierającemu materiał do projektów, umiejącemu zaproponować kilka alternatywnych rozwiązań, wykonującemu projekt bezbłędnie i estetycznie.

Stopień celujący (poziom wymagań W - wykraczający)

Uczeń wykorzystując dostępne narzędzia programowe, potrafi samodzielnie rozwiązać złożone (nietypowe) problemy z zakresu określonego w planie wynikowym. Formułuje problemy w kategoriach informatyki. Bierze czynny udział w konkursach i olimpiadach informatycznych, zdobywając punktowane miejsca. Przygotowuje i prowadzi prelekcje oraz referaty poszerzające tematykę zajęć, wykazując się wiadomościami wykraczającymi poza program nauczania. Potrafi rozwijać swoje uzdolnienia.

Na **ocenę celującą** zasługuje uczeń, który w czasie ćwiczenia stosuje zaawansowane funkcje programu i sprzętu nieomawiane na zajęciach (wykraczające ponad wymagania programowe), który wykona projekt na dobrym poziomie, estetyczny, dobrze skomponowany. Uczeń umie także zaproponować własne, oryginalne pomysły, a jego projekty są funkcjonalne i wykończone.

Umiejętności nabyta przez ucznia	Wymagania na ocenę:				
	2	3	4	5	6
zna pozycyjne systemy liczbowe i potrafi przeliczać liczby zapisane w jednym systemie na inny	•	•	•	•	•
zna pojęcie algorytmu, umie podać przykłady zadań algorytmicznych	•	•	•	•	•
potrafi zapisać prosty algorytm w pseudojęzyku lub za pomocą listy kroków	•	•	•	•	•
zapisuje prosty algorytm za pomocą schematu blokowego		•	•	•	•
potrafi wymienić cechy algorytmu		•	•	•	•
umie wyjaśnić, co to jest algorytm liniowy	•	•	•	•	•
wyjaśnia, co to jest algorytm rozgałęziony	•	•	•	•	•
wyodrębnia części składowe algorytmu	•	•	•	•	•
omawia klasyczne, proste algorytmy		•	•	•	•
umie dokonać analizy algorytmu			•	•	•
potrafi określić złożoność obliczeniową algorytmu				•	•
umie obsługiwać narzędzie służące do implementacji programów w wybranym języku programowania	•	•	•	•	•
pisze kod źródłowy w sposób przejrzysty, stosuje komentarze i wcięcia	•	•	•	•	•
zna podstawowe elementy języka programowania	•	•	•	•	•
rozumie pojęcia: stałe, zmienne i umie podać przykłady ich zastosowania	•	•	•	•	•
potrafi zadeklarować zmienne prostych typów w pisanim kodzie	•	•	•	•	•
umie zaimplementować, skompilować i uruchomić prosty program liniowy	•	•	•	•	•
rozumie pojęcia: funkcja, procedura i umie je wykorzystać przy implementacji programu			•	•	•
zna operatory matematyczne i umie je zastosować w programie	•	•	•	•	•
zna operatory logiczne i relacji, potrafi je zastosować w programie		•	•	•	•
umie zastosować typ tablicowy w implementacji programów (tablica jednowymiarowa o elementach typu prostego)		•	•	•	•
rozumie pojęcie struktury i potrafi podać przykłady zmiennych strukturalnych	•	•	•	•	•
definiuje strukturę składającą się z kilku pól		•	•	•	•
umie zadeklarować tablicę jednowymiarową o elementach typu zdefiniowanej przez siebie struktury			•	•	•
potrafi przekazać do funkcji jako parametr formalny tablicę jednowymiarową o elementach typu prostego			•	•	•
umie przekazać do funkcji jako parametr formalny tablicę jednowymiarową o elementach typu strukturalnego				•	•
stosuje typ tablicowy w implementacji programów (tablica mająca dwa wymiary lub więcej)				•	•

potrafi zaimplementować program przeszukujący ciąg znaków w celu odnalezienia wyróżnionego elementu	•	•	•	•	•
potrafi zaimplementować program porządkujący ciąg elementów metodą bąbelkową		•	•	•	•
potrafi zaimplementować program porządkujący ciąg elementów metodą sortowania przez wstawianie			•	•	•
potrafi zaimplementować program porządkujący ciąg elementów metodą sortowania przez selekcję			•	•	•
potrafi zaimplementować iteracyjnie programy realizujące proste metody numeryczne i metodę Monte Carlo			•	•	•
zna pojęcie rekurencji, umie podać proste przykłady jej zastosowania (ciąg Fibonacciego, silnia, potęga)	•	•	•	•	•
umie samodzielnie zaimplementować proste programy rekurencyjne (jak powyżej)			•	•	•
potrafi napisać program rozwiązujący klasyczny problem wież Hanoi				•	•
umie przeanalizować i zaimplementować program klasycznego problemu skoczka szachowego					•
potrafi przeanalizować i rozwiązać implementacyjnie problem ośmiu hetmanów					•
umie wyjaśnić, na czym polega metoda sortowania szybkiego			•	•	•
potrafi samodzielnie zaimplementować program sortujący metodą szybką				•	•
zna schemat Hornera i umie go zaimplementować			•	•	•
zna algorytm Euklidesa i umie go zaimplementować			•	•	•
zna metody szyfrowania, implementuje przynajmniej jedną z nich			•	•	•
implementuje program, na przykład kolejnych przybliżeń miejsca zerowego funkcji wielomianowej			•	•	•
wykorzystuje w programach funkcje zdefiniowane w plikach bibliotecznych napisanych przez siebie				•	•
zna pojęcie wskaźnika, umie zastosować wskaźnik do obsługi zmiennych typów prostych				•	•
stosuje wskaźniki przy pracy z tablicami				•	•
tworzy tablice dynamiczne i obsługuje je za pomocą wskaźnika					•
wykonuje operacje wejście i wyjścia na plikach	•	•	•	•	•

Dostosowanie wymagań dla ucznia z dysleksją rozwojową.

- docenić chęć pokonywania trudności, wysiłek i wytrwałość w działaniu, samodzielność ład w miejscu pracy i porządek w działaniu
- pozostawić więcej czasu na wykonanie pracy
- stosować polecenia krótkie i nieskomplikowane
- upewnić się czy uczeń zrozumiał polecenie
- w pracach klasowych zadaniach przeznaczonych do samodzielnego wykonania upewnić się czy uczeń zrozumiał polecenie. Zadania powinny być zapisane na kartce-nie dyktować.
- posadzić ucznia blisko nauczyciela by nauczyciel mógł kontrolować pracę ucznia

W przypadku innych dysfunkcji - szczegółowe dostosowanie wymagań – zgodnie z orzeczeniem poradni psychologiczno – pedagogicznej.