

PRZEDMIOTOWY SYSTEM OCENIANIA Z INFORMATYKI

Przedmiotowy system oceniania jest zgodny z:

- Ustawą z dnia 7 września 1991 r. o systemie oświaty.
- Rozporządzeniem MEN z 10 czerwca 2015 r. w sprawie szczegółowych warunków i sposobu oceniania, klasyfikowania i promowania uczniów oraz słuchaczy w szkołach publicznych.
- Rozporządzeniem MEN z dnia 26 lutego 2019 r. zmieniającym rozporządzenie w sprawie szczegółowych warunków i sposobu oceniania, klasyfikowania i promowania uczniów i słuchaczy w szkołach publicznych.
- Szczegółowymi zasadami oceniania wewnątrzszkolnego.
- Programem nauczania matematyki w szkole podstawowej.

System uwzględnia wymagania zawarte w obowiązującej Podstawie Programowej.

Opracowanie obejmuje:

- I.** Cele wewnątrzszkolnego oceniania.
- II.** Cele ogólne oceniania w informatyce.
- III.** Program nauczania.
- IV.** Kryteria oceniania poszczególnych form aktywności.
- V.** Wymagania edukacyjne na poszczególne stopnie szkolne.
- VI.** Ocenianie semestralne i końcowe.
- VII.** Sposoby poprawiania oceny szkolnej.
- VIII.** Sposoby informowania uczniów i rodziców o pracy i postępach w nauce.
- IX.** Odwołanie od oceny śródrocznej lub rocznej
- X.** Tryb i warunki uzyskiwania przez uczniów wyższych niż przewidywane śródrocznych lub rocznych ocen klasyfikacyjnych
- XI.** Zakres wiedzy i umiejętności.

I. Cele wewnątrzszkolnego oceniania

Ocenianie wewnątrzszkolne osiągnięć edukacyjnych ucznia ma na celu:

- a) poinformowanie ucznia o poziomie jego osiągnięć edukacyjnych i postępach w tym zakresie,
- b) pomoc uczniowi w samodzielnym planowaniu swojego rozwoju, wspomaganie efektów uczenia się,
- c) motywowanie ucznia do dalszej pracy,
- d) dostarczanie rodzicom (prawnym opiekunom) i nauczycielom informacji o postępach, trudnościach i specjalnych uzdolnieniach ucznia
- e) umożliwienie nauczycielom doskonalenia organizacji i metod pracy dydaktyczno-wychowawczej.

II. Cele ogólne oceniania w informatyce

Przedmiotowy system oceniania powinien być:

- użyteczny – wskazywać co jest najważniejsze dla uczniów w procesie nauczania, uczenia się.
- wielowątkowy – skoncentrowany na umiejętnościach każdego ucznia z osobna,
- otwarty – daje możliwość porównywania osiągnięć uczniów ze standardami,

oraz powinien:

- wspomagać proces nauczania i uczenia się,
- zapewniać pewność wnioskowania i spójność wewnętrzną.

Cele systemu to:

- rozpoznanie przez nauczyciela poziomu i postępów w opanowaniu przez ucznia wiadomości i umiejętności w stosunku do wymagań programowych,
- poinformowanie ucznia o poziomie jego osiągnięć w dziedzinie informatyki i postępach w tym zakresie,
- pomoc uczniowi poprzez przekazywanie informacji o tym, co zrobił dobrze i jak powinien się dalej uczyć,
- pomoc uczniowi w samodzielnym kształceniu,
- motywowanie ucznia do dalszej pracy,
- dostarczanie rodzicom (prawnym opiekunom) informacji o postępach, trudnościach i specjalnych uzdolnieniach ucznia.

III. Program nauczania

Program „Lubię to!” został zatwierdzony przez MEN i wpisany do wykazu programów. Jest zgodny z podstawą programową kształcenia ogólnego dla szkoły podstawowej.

Program umożliwia wypełnienie wszystkich celów i zadań określonych w podstawie programowej.

W procesie nauczania z tym programem, uczeń nabywa wiedzę i umiejętności umożliwiające edukację informatyczną na kolejnych etapach kształcenia.

W procesie dydaktycznym uczeń jest traktowany podmiotowo.

Nauczanie informatyki odbywa się poprzez dostarczanie uczniom przykładów sytuacji bliskich ich doświadczeniu, a ukazujących potrzebę zastosowań informatyki, wykorzystując wiadomości z innych dziedzin.

Sprzyja kształceniu umiejętności określonych w standardach wymagań egzaminacyjnych.

IV. Kryteria oceniania poszczególnych form aktywności

Ocenie podlegają:

- ćwiczenia praktyczne,
- prace klasowe, sprawdziany,
- kartkówki,
- aktywność i praca ucznia na lekcji.

Ad. Ćwiczenia praktyczne:

Ćwiczenia praktyczne obejmują zadania praktyczne, które uczeń wykonuje podczas lekcji.

Oceniając je, nauczyciel bierze pod uwagę wartość merytoryczną, stopień zaangażowania w wykonanie ćwiczenia, dokładność wykonania polecenia, indywidualne rozwiązania zastosowane przez ucznia, staranność i estetykę.

Ad. Prace klasowe oraz sprawdziany:

Prace klasowe oraz sprawdziany są przeprowadzane po zakończeniu każdego działu lub po zrealizowaniu części dłuższego działu. Są przeprowadzane w formie pisemnej i/lub praktycznej. Praca klasowa jest zapowiadana co najmniej tydzień wcześniej. Prace klasowe są obowiązkowe. Jeżeli uczeń opuścił pracę klasową z przyczyn losowych, powinien ją napisać w terminie uzgodnionym z nauczycielem. Prace klasowe przechowuje nauczyciel i są do wglądu dla uczniów i ich rodziców.

Ad. Kartkówki:

Pisemne kartkówki sprawdzają wiadomości i umiejętności z trzech ostatnich lekcji. Są przeprowadzane w formie pisemnej. Mogą nie być zapowiedziane przez nauczyciela.

Ad. Aktywność i praca ucznia na lekcji

Uczeń może uzyskać pozytywną ocenę m.in. za: samodzielne wykonanie pracy na lekcji, aktywną pracę w grupie, pomoc koleżeńską na lekcji przy rozwiązywaniu problemu, inicjatywę przy rozwiązywaniu problemów, znalezienie nieszablonowych rozwiązań.

Zadania z prac klasowych, sprawdzianów oceniane są punktowo, o ogólnej ocenie decyduje suma zdobytych punktów.

Obowiązująca skala ocen:

- **0% - 29%** możliwych do zdobycia punktów = **ocena niedostateczna**
- **30% – 49%** możliwych do zdobycia punktów = **ocena dopuszczająca**
- **50% – 74%** możliwych do zdobycia punktów = **ocena dostateczna**
- **75% – 89%** możliwych do zdobycia punktów = **ocena dobra**
- **90% – 99%** możliwych do zdobycia punktów = **ocena bardzo dobra**
- **100%** możliwych pkt = **ocena celująca**

V. Wymagania edukacyjne na poszczególne stopnie szkolne.

Stopień **celujący** otrzymuje uczeń, który:

- wyróżnia się wiedzą i umiejętnościami określonymi w programie nauczania przedmiotu obowiązującymi w danej klasie,
- samodzielnie i twórczo rozwija własne uzdolnienia,
- zdobytą wiedzę stosuje w rozwiązywaniu problemów teoretycznych i praktycznych,
- samodzielnie i twórczo dobiera stosowne rozwiązanie w nowych, nietypowych sytuacjach problemowych,
- samodzielnie stosuje właściwe algorytmy dla rozwiązania danych problemów i przewiduje ich następstwa,
- wie, jak poprawić ewentualne błędy,
- sprawnie posługuje się poznanymi programami użytkowymi,
- chętnie podejmuje prace dodatkowe, służy pomocą innym, pomaga w pracach związanych z prawidłowym funkcjonowaniem pracowni.

Stopień **bardzo dobry** otrzymuje uczeń, który:

- opanował wiedzę i umiejętności określone w programie nauczania przedmiotu obowiązującego w danej klasie,
- potrafi zastosować zdobytą wiedzę w praktyce,
- samodzielnie stosuje właściwe algorytmy dla rozwiązania danych problemów i przewiduje ich następstwa,
- wie, jak poprawić ewentualne błędy,
- sprawnie posługuje się poznanymi programami użytkowymi.

Stopień **dobry** otrzymuje uczeń, który:

- dobrze opanował wiadomości określone programem nauczania,
- korzystając ze wskazówek nauczyciela rozwiązuje zadania i problemy,
- potrafi samodzielnie projektować algorytmy rozwiązań,
- zna podstawowe pojęcia i właściwą terminologię z przedmiotu,
- czasem popełnia błędy, ale potrafi je wskazać i poprawić.

Stopień **dostateczny** otrzymuje uczeń, który:

- opanował podstawowe treści programowe określone programem nauczania danej klasy,
- posiadał umiejętności typowe i wykonuje zadania o średnim stopniu trudności,
- umie opisać przebieg wykonania zadania i rozumie sens jego rozwiązania,
- potrafi posługiwać się podstawowymi programami użytkowymi i wykonywać zadania o niewielkim stopniu trudności,

Stopień **dopuszczający** otrzymuje uczeń, który:

- niewystarczająco opanował wiadomości określone programem nauczania w danej klasie,
- rozumie pojęcia informatyczne,
- ma trudności z obsługą systemu operacyjnego i podstawowych programów użytkowych,
- stosuje posiadane wiadomości tylko z pomocą nauczyciela,
- ma trudności z zastosowaniem swojej wiedzy w praktyce.

Stopień **niedostateczny** otrzymuje uczeń, który:

- nie opanował wiadomości i umiejętności określonych programem nauczania danej klasy,
- ma braki w wiadomościach i umiejętnościach uniemożliwiające kontynuację dalszej nauki z przedmiotu,
- nie potrafi wykonać zadań o podstawowym stopniu trudności, nawet z pomocą nauczyciela.

VI. Ocenianie semestralne i końcowe.

Zasady klasyfikowania śródrocznego i rocznego zawarte są w szczegółowych zasadach oceniania wewnątrzszkolnego w Szkole Podstawowej

Przy wystawianiu oceny śródrocznej i rocznej brane są pod uwagę oceny cząstkowe uzyskane przez ucznia w danym okresie czasu a w przypadku tej ostatniej uwzględnia się również ocenę śródroczną.

W skład oceny wchodzi wiedza, umiejętności i postawy ucznia.

Ocena śródroczna i roczna nie jest średnią arytmetyczną ocen cząstkowych.

VII. Sposoby poprawiania oceny szkolnej

Uczeń ma prawo poprawić każdą ocenę z pracy klasowej lub sprawdzianu. Uczeń pisze poprawę tylko jeden raz.

Uczeń, który otrzymał niedostateczną ocenę z pracy klasowej lub sprawdzianu jest zobowiązany do podjęcia próby poprawy oceny.

Termin poprawy należy uzgodnić wcześniej z nauczycielem.

Do dziennika obok oceny z pracy klasowej lub sprawdzianu wpisuje się ocenę, którą uczeń uzyskał w wyniku poprawy.

Pozostałe oceny nie podlegają poprawie.

Uczeń, który w wyznaczonym dniu nie przekazał do sprawdzenia nauczycielowi pracy wykonywanej podczas lekcji (ćwiczenia praktycznego) otrzymuje w dzienniku zapis „0” i ma obowiązek przesłać daną pracę nauczycielowi w jak najszybszym możliwym terminie (maksymalnie w ciągu miesiąca).

VIII. Sposoby informowania uczniów i rodziców o pracy i postępach w nauce

Na początku roku szkolnego informowanie uczniów i rodziców o wymaganiach edukacyjnych wynikających z realizowanego programu nauczania oraz sposobach sprawdzania osiągnięć edukacyjnych uczniów.

Sprawdzone i ocenione prace pisemne, oceny bieżące zainteresowani rodzice mogą otrzymać do wglądu na zebraniu rodziców lub indywidualnym spotkaniu z nauczycielem.

Uczeń jest na bieżąco informowany o otrzymanych ocenach.

Każda ocena jest jawna.

Informację o przewidywanej ocenie klasyfikacyjnej podaje się uczniowi na tydzień przed klasyfikacyjnym posiedzeniem rady pedagogicznej.

Informację o przewidywanej ocenie niedostatecznej otrzymuje uczeń i jego rodzice miesiąc przed klasyfikacyjnym posiedzeniem rady pedagogicznej.

Dopuszcza się w dzienniku lekcyjnym stosowania zapisu „0”, który ma informować o nieprzekazaniu nauczycielowi pracy ucznia w wyznaczonym terminie.

IX. Odwołanie od oceny śródrocznej lub rocznej

Uczniowi lub jego rodzicom (prawnym opiekunom) przysługuje prawo do odwołania się od oceny śródrocznej lub rocznej zgodnie z zasadami zamieszczonymi w Statucie Szkoły.

X. Tryb i warunki uzyskiwania przez uczniów wyższych niż przewidywane śródrocznych lub rocznych ocen klasyfikacyjnych

Uczeń ma prawo starać się o uzyskanie wyższej, niż przewidywana przez nauczyciela, śródrocznej lub rocznej oceny klasyfikacyjnej z zajęć edukacyjnych w wyjątkowych sytuacjach np.:

- 1) w przypadku dłuższej nieobecności nauczyciela, która uniemożliwiła uczniowi wcześniejszą poprawę oceny;
- 2) w przypadku dłuższej usprawiedliwionej nieobecności ucznia w szkole;
- 3) w innych losowych sytuacjach.

Uczeń może ubiegać się o podwyższenie przewidywanej oceny tylko o jeden stopień.

Uczeń może uzyskać wyższą niż proponowana ocenę śródroczną lub roczną na podstawie sprawdzianu składającego się z zadań o poziomie wymagań na ocenę, o którą uczeń się ubiega.

Sprawdzian ten uczeń pisze na pisemny wniosek rodziców (prawnych opiekunów), który powinien wpłynąć do nauczyciela uczącego w terminie nie późniejszym niż 2 dni robocze po powiadomieniu ucznia oraz jego rodziców o przewidywanej ocenie śródrocznej lub rocznej.

Zdobycie powyżej 80% punktów z takiego sprawdzianu skutkuje podwyższeniem oceny śródrocznej lub rocznej.

Termin sprawdzianu ustala nauczyciel.

XI. Zakres wiedzy i umiejętności.

Zakres wiedzy i umiejętności znajduje się w tabeli poniżej.

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.

Temat lekcji	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					
Formuły i adresowanie względne w arkuszu kalkulacyjnym	<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
Funkcje oraz adresowanie bezwzględne i mieszane w arkuszu kalkulacyjnym	<ul style="list-style-type: none"> rozumie różnice między adresowaniem względnym, bezwzględnym i mieszanym 	<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
Przedstawianie danych na wykresie	<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
Zastosowania arkusza kalkulacyjnego	<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					
Wprowadzenie do programowania w języku Python	<ul style="list-style-type: none"> definiuje pojęcia: algorytm, program, programowanie 	<ul style="list-style-type: none"> wymienia różne sposoby przedstawienia algorytmu: opis 	<ul style="list-style-type: none"> wymienia przykładowe środowiska programistyczne 	<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"> • podaje kilka sposobów przedstawienia algorytmu 	<p>słowny, schemat blokowy, lista kroków</p> <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"> • wyjaśnia, czym jest specyfikacja problemu • opisuje etapy rozwiązywania problemów • opisuje etapy powstawania programu komputerowego • zapisuje proste polecenia języka Python • 		
Piszemy programy w języku 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 if oraz if else w programach • wykorzystuje iterację w konstruowanych algorytmach • wykorzystuje w programach instrukcję iteracyjną for • definiuje funkcje w języku Python i omawia różnice między funkcjami zwracającymi wartość a funkcjami niezwracającymi wartości 	<ul style="list-style-type: none"> • buduje złożone schematy blokowe służące do przedstawiania skomplikowanych algorytmów • 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
Algorytmy na liczbach naturalnych	<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 while 	<ul style="list-style-type: none"> • omawia algorytm Euklidesa w wersji z odejmowaniem i z dzieleniem – 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 while a pętlą for • pisze programy obliczające NWD, stosując algorytm Euklidesa, oraz wypisujące cyfry danej liczby • wyjaśnia różnice między algorytmem Euklidesa w wersjach z odejmowaniem i z dzieleniem 	<ul style="list-style-type: none"> • pisze programy wykorzystujące algorytmy Euklidesa (np. obliczający NWW) oraz wyodrębniania cyfr danej liczby
Algorytmy wyszukiwania	<ul style="list-style-type: none"> • wyjaśnia potrzebę wyszukiwania informacji w zbiorze 	<ul style="list-style-type: none"> • zapisuje algorytm wyszukiwania elementu w zbiorze nieuporządkowanym, w tym 	<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 	<ul style="list-style-type: none"> • samodzielnie modyfikuje i optymalizuje algorytmy wyszukiwania

	<ul style="list-style-type: none"> określa różnice między wyszukiwaniem w zbiorach uporządkowanym i nieuporządkowanym sprawdza działanie programów wyszukiwających element w zbiorze 	<p>elementu największego i najmniejszego</p> <ul style="list-style-type: none"> zapisuje algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia implementuje grę w zgadywanie liczby 	<ul style="list-style-type: none"> omawia funkcje zastosowane w realizacji algorytmu wyszukiwania metodą połowienia implementuje algorytm wyszukiwania największej wartości w zbiorze 	<p>zbiorze metodą połowienia, w tym elementu największego i najmniejszego</p> <ul style="list-style-type: none"> implementuje algorytm wyszukiwania elementu w zbiorze metodą połowienia 	
Algorytmy porządkowania	<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 metodami przez wybieranie oraz przez zliczanie omawia implementację algorytmu sortowania przez wybieranie stosuje pętle zagnieżdżone i wyjaśnia, jak działają 	<ul style="list-style-type: none"> omawia implementację algorytmu sortowania przez zliczanie omawia funkcje zastosowane w kodzie źródłowym algorytmów sortowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> implementuje algorytmy porządkowania metodami przez wybieranie oraz przez zliczanie wprowadza modyfikacje w implementacji algorytmów porządkowania przez wybieranie oraz przez zliczanie 	<ul style="list-style-type: none"> samodzielnie modyfikuje i optymalizuje programy sortujące metodą przez wybieranie, metodą przez zliczanie
DZIAŁ 4. Projekty					
Dokumentacja szkolnej imprezy sportowej	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, wykonując powierzone mu zadania o niewielkim stopniu trudności 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej wprowadza dane do zaprojektowanych tabel 	<ul style="list-style-type: none"> przygotowuje dokumentację imprezy, wykonuje obliczenia, projektuje tabele oraz wykresy współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, przygotowuje zestawienia, drukuje wyniki współpracuje w grupie podczas pracy nad projektem 	<ul style="list-style-type: none"> bierze udział w przygotowaniu dokumentacji szkolnej imprezy sportowej, tworzy zestawienia zawierające zaawansowane formuły, wykresy oraz elementy graficzne współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera
Sterowanie obiektem na ekranie	<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
Historia i rozwój informatyki	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności – znalezienie informacji w internecie, umieszczenie ich w chmurze 	<ul style="list-style-type: none"> współpracuje z innymi podczas pracy nad projektem analizuje zebrane dane tworzy projekt prezentacji multimedialnej 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania tworzy prezentację wg projektu zaakceptowanego przez zespół 	<ul style="list-style-type: none"> aktywnie uczestniczy w pracach zespołu analizuje i weryfikuje pod względem merytorycznym i technicznym przygotowaną prezentację 	<ul style="list-style-type: none"> współpracuje w grupie podczas pracy nad projektem, przyjmuje funkcję lidera wzbogaca prezentację o elementy podnoszące jej walory estetyczne i merytoryczne

<p>Informatyka w moim przyszłym życiu</p>	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, realizuje powierzone zadania o niewielkim stopniu trudności • bierze aktywny udział w dyskusji nad wyborem atrakcyjnego zawodu wymagającego kompetencji informatycznych 	<ul style="list-style-type: none"> • gromadzi informacje dotyczące wybranych zawodów, umieszcza je w zaprojektowanych tabelach i dokumentach tekstowych 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu • projektuje tabele do zapisywania informacji o zawodach • weryfikuje i formatuje przygotowane dokumenty tekstowe 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, weryfikuje opracowane treści i łączy wszystkie dokumenty w całość 	<ul style="list-style-type: none"> • aktywnie uczestniczy w pracach zespołu, przyjmuje rolę lidera • podczas dyskusji przyjmuje funkcję moderatora
---	--	--	--	---	--