

WEWNĄTRZSZKOLNE ZASADY OCENIANIA z INFORMATYKI (zakres ROZSZERZONY)

I Liceum Ogólnokształcące im. Marii Skłodowskiej – Curie w Pile

Klasa 2 i 3

Obowiązujące pomoce: podręcznik „Informatyka część 1 i2” G. Zawadzka, D. Korman wyd. Helion; zeszyt przedmiotowy

1. Ocenie podlegają:

a) praca na lekcji:

- ćwiczenia praktyczne,
- odpowiedzi ustne,
- jakość pracy,
- aktywność na lekcji,
- współpraca w grupie,

b) sprawdziany wiadomości i umiejętności po każdym zrealizowanym dziale: (mogą mieć formę sprawdzianu pisemnego lub testu komputerowego w formie zadań praktycznych i teoretycznych), zapowiedziane tydzień wcześniej (termin sprawdzianu nauczyciel wpisuje ołówkiem do dziennika)

c) kartkówki,

d) prace domowe

- bieżące – utrwalające lub przygotowujące do opracowania nowej lekcji,
- długoterminowe – stanowiące pracę nad projektem tematycznym

e) inne osiągnięcia ucznia (np. udział w olimpiadach, konkursach przedmiotowych, projektach edukacyjnych z zakresu informatyki, itp.)

2. Ogólne zasady oceniania:

- stopień niedostateczny otrzymuje uczeń, który nie spełnia wymagań podanych niżej kryteriów ocen pozytywnych,
- stopień dopuszczający otrzymuje uczeń, który ma braki w opanowaniu treści ustalonej w podstawie programowej kształcenia ogólnego danej dziedziny edukacyjnej, ale braki te nie przekreślają możliwości uzyskania przez ucznia podstawowej wiedzy i umiejętności z danej dziedziny w ciągu dalszej nauki; rozwiązuje typowe zadania teoretyczne i praktyczne o niewielkim stopniu trudności.
- stopień dostateczny otrzymuje uczeń, który opanował wiadomości i umiejętności określone programem nauczania danej dziedziny edukacyjnej na poziomie nie przekraczającym wy-

magań zawartych w podstawie programowej; rozwiązuje typowe zadania teoretyczne i praktyczne o średnim poziomie trudności.

- stopień dobry otrzymuje uczeń, który nie opanował w pełni wiadomości określonych programem nauczania danej dziedziny edukacyjnej, ale opanował je na poziomie przekraczającym wymagania zawarte w podstawie programowej kształcenia ogólnego; poprawnie stosuje wiadomości i umiejętności, rozwiązuje samodzielnie typowe zadania teoretyczne i praktyczne.
- stopień bardzo dobry otrzymuje uczeń, który opanował pełny zakres wiedzy i umiejętności określonej programem nauczania danej dziedziny edukacyjnej; sprawnie posługuje się zdobytymi wiadomościami, rozwiązuje samodzielnie problemy teoretyczne i praktyczne ujęte w programie nauczania, potrafi zastosować poznaną wiedzę do rozwiązywania zadań i problemów w nowych sytuacjach.
- stopień celujący otrzymuje uczeń, który samodzielnie opanował treści nauczania, biegle posługuje się zdobytymi wiadomościami i umiejętnościami w rozwiązywaniu problemów teoretycznych i praktycznych problemów programu nauczania danej dziedziny edukacyjnej, proponuje rozwiązania nietypowe; jest twórczy i samodzielnie rozwija własne uzdolnienia, uczestniczy w olimpiadach i konkursach przedmiotowych.

3. Ustala się następujące progi procentowe na poszczególne stopnie szkolne: 0% - 35% niedostateczny; 36% - 49% dopuszczający; 50% - 69% dostateczny; 70% - 85% dobry; 86% - 95% bardzo dobry; 96% - 100% celujący.

4. Szczegółowe wymagania edukacyjne (wymagania na poszczególne oceny)

Dopuszczający:

Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe. Wie, na czym polega powtarzanie tych samych operacji. Potrafi omówić, na przykładzie, algorytm znajdowania najmniejszego z trzech elementów. Potrafi zanalizować przebieg algorytmu dla przykładowych danych i ocenić w ten sposób jego poprawność. Zna klasyfikację języków programowania. Zna ogólną budowę programu i najważniejsze elementy języka – słowa kluczowe, instrukcje, wyrażenia, zasady składni. Potrafi zrealizować prosty algorytm liniowy w języku wysokiego poziomu; potrafi skompilować i uruchomić gotowy program. Wymienia przykłady prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach. Określa następujące pojęcia: bit, bajt. Zna pojęcie systemu pozycyjnego. Wymienia części składowe zestawu komputerowego, podaje ich parametry i przeznaczenie. Rozróżnia rodzaje pamięci komputera, określa ich własności i przeznaczenie. Wie, co to jest system operacyjny, i korzy-

sta z jego podstawowych funkcji. Wykonuje podstawowe operacje na plikach i folderach. Zna pojęcie sieci komputerowej, potrafi wymienić jej rodzaje. Zna pojęcie logowania. Potrafi wymienić kilka cech pracy w sieci, odróżniających ją od pracy na autonomicznym komputerze. Zna kilka sposobów połączenia z Internetem. Potrafi omówić historię komputerów. Umie wskazać ogólny kierunek zmian w technologiach komputerowych. Zna podstawowe zasady netykiety.

Wie, czym zajmuje się informatyka i jakie są jej metody, umie uruchomić komputer i programy komputerowe, umie komunikować się z komputerem za pomocą systemu operacyjnego w elementarnym zakresie, umie posługiwać się podstawowym oprogramowaniem użytkowym w elementarnym zakresie, umie stosować posiadane wiadomości do wykonywania elementarnych czynności w bardzo prostych sytuacjach, w wypowiedziach popełnia liczne błędy merytoryczne; zna podstawowe pojęcia dotyczące sieci komputerowej, wie, jakie urządzenia realizują przepływ informacji przez sieć, potrafi samodzielnie konfigurować komputery do pracy w sieciach, zna zastosowanie narzędzi administracyjnych systemu Windows; wie, jak się zapewnia bezpieczeństwo zasobów komputera, wie, jak wykorzystać konta w sieciach komputerowych z kontrolerami domeny, wie, jak zbudować małą sieć lokalną z możliwością wyjścia do Internetu, zna możliwości współczesnych serwerów sieciowych, zna istotę funkcjonowania komputerowej bazy danych, rozumie modele funkcjonowania systemu zarządzania bazą danych, zna systemy organizacji danych, potrafi zaprojektować prostą relacyjną bazę danych; potrafi opisywać podstawowe obiekty w relacyjnej bazie danych, wie, jak projektować tabele, potrafi ustalać odpowiednie relacje w bazie danych, zna podstawowe zasady tworzenia poleceń w języku SQL, potrafi budować w języku SQL proste zapytanie do bazy danych, zna techniki tworzenia statycznych serwisów www, potrafi klasyfikować multimedia ze względu na zastosowania, zna podstawowe wielkości opisujące dźwięk i obraz przechowywane w pamięci komputera, zna podstawowe formaty plików przechowujących dźwięki i obrazy potrafi przetwarzać; pliki graficzne i tworzyć proste animacje potrafi montować dźwięk i obraz; potrafi projektować schematy prezentacji i wykorzystywać w nich własne materiały multimedialne, zna podstawy prawa autorskiego

Dostateczny: Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm w postaci listy kroków. Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu. Korzysta (w stopniu podstawowym) z programu edukacyjnego do symulacji działania algorytmu skonstruowanego w postaci schematu blokowego. Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami). Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń. Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy. Oma-

wia i analizuje wybrane techniki sortowania w postaci gotowych schematów blokowych, skonstruowanych w programie edukacyjnym. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie. Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze. Rozumie pojęcia: implementacja, kompilacja, uruchomienie, testowanie. Rozumie znaczenie i działanie podstawowych instrukcji wybranego języka programowania wysokiego poziomu. Rozróżnia i poprawia błędy kompilacji i błędy wykonania. Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu. Zna podstawowe zasady poprawnego programowania; testuje tworzone programy; wie, jak uniknąć problemów, takich jak np. zapętlenie się programu. Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady. Deklaruje typy złożone. Wie, co to jest system binarny, i potrafi dokonać zamiany liczby z systemu dziesiętnego na binarny i odwrotnie. Potrafi sklasyfikować środki (urządzenia) i narzędzia (oprogramowanie) technologii informacyjnej. Wie, jak działa komputer. Wyjaśnia rolę procesora. Rozumie organizację pamięci komputerowych. Potrafi omówić funkcje systemu operacyjnego. Zna zasady ochrony plików. Potrafi nadać podstawowe atrybuty plikom, jak też wyszukać poszczególne pliki. Wymienia korzyści płynące z korzystania z sieci. Zna podstawowe klasy i topologie sieciowe. Potrafi wymienić urządzenia i elementy sieciowe oraz omówić ich ogólne przeznaczenie. Zna cechy systemu działającego w szkolnej pracowni. Orientuje się – w zakresie podstawowym – w działaniu Internetu. Potrafi określić nowoczesne trendy w zastosowaniu urządzeń komputerowych. Jest w stanie omówić prawne i społeczne aspekty zastosowania informatyki. Spełnia wszystkie wymagania na ocenę niższą. Wymienia algorytmy klasyczne zawarte w podstawie programowej.

Rozumie i potrafi opisywać problemy dostępu i aktualizacji danych w sieciach komputerowych, rozumie dynamiczne struktury danych, rozumie istotę deklaracji typu obiektowego i zmiennych obiektowych potrafi publikować serwisy www, potrafi wysyłać zapytanie do serwera www i odsyłać odpowiedź, zna elementy teorii zarządzania projektami, potrafi wskazać zagadnienia o szczególnym znaczeniu etycznym, zna zagrożenia wynikające z połączenia komputerów w sieci wie, jakie zagrożenia niesie wymiana informacji przez Internet zna zagrożenia wynikające z działania wirusów komputerowych

Dobry: Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów. Przedstawia dokładną specyfikację dowolnego zadania. Zna znaczenie i działanie instrukcji symbolicznego języka programowania (pseudojęzyka). Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci. Potrafi skonstruować algorytm z warunkami zagnieżdżonymi i pętlą za pomocą programu edukacyjnego. Analizuje algorytmy, w których występują powtórzenia (iteracje). Zna sposoby zakończenia iteracji. Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego. Zna iteracyjną postać algorytmu Euklidesa. Zna przynajmniej dwie techniki sortowania, np. bą-

belkowe i przez wybór. Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Zna rekurencyjną realizację wybranego algorytmu, np. silni. Rozumie, co to jest złożoność algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania. Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna rekurencyjne realizacje prostych algorytmów. Rozumie i stosuje zasady programowania strukturalnego. Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach. Wie, jakie znaczenie ma zasięg działania zmiennej. Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą. Zna zasady działania wybranych algorytmów sortowania. Zna podstawowe procedury graficzne, potrafi narysować na ekranie wykres funkcji i podstawowe figury geometryczne. Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach. Potrafi wykonać działania arytmetyczne na liczbach binarnych (dodawanie i odejmowanie). Zna system szesnastkowy i potrafi wykonać konwersję liczb binarnych na liczby w systemie szesnastkowym i odwrotnie. Analizuje model logiczny komputera. Wie, co to jest kod ASCII. Potrafi wymienić rodzaje aktualnie używanych komputerów. Zna metody wyszukiwania plików. Zna znaczenie protokołu w sieciach (w tym TCP/IP). Wie, na czym polega wymiana informacji w sieci. Zna zasady pracy w sieci, m.in. zasady udostępniania zasobów. Potrafi omówić zagrożenia płynące z sieci. Charakteryzuje różne połączenia z Internetem; potrafi omówić przesyłanie pakietów danych w Internecie. Potrafi wskazać nowości w zakresie usług internetowych oraz odszukać informacje na temat najnowszych pomysłów na komputery. Spełnia wszystkie wymagania na oceny niższe. Omawia algorytmy klasyczne zawarte w podstawie programowej.

Potrafi tworzyć formularze, potrafi stosować różne rodzaje kwerend, potrafi projektować i wykonywać raporty, potrafi budować kolejkę i stos, potrafi korzystać z obiektowych technik tworzenia programów rozumie wybrane algorytmy uwzględniające przybliżone wyniki obliczeń potrafi korzystać z dostępu do baz danych za pomocą ODBC.

Bardzo dobry: Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji), m.in. w pseudojęzyku. Zapisuje algorytmy z pętlą zagnieżdżoną. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Stosuje swobodnie oprogramowanie edukacyjne do graficznej prezentacji i analizy algorytmów. Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji. Zna inne algorytmy sortowania, np. kubełkowe, przez wstawianie. Zna przynajmniej jeden algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego. Wskazuje różnicę między rekurencją a iteracją. Zna rekurencyjną realizację wybranych algorytmów, np. silnię i algorytm Euklidesa. Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną. Potrafi porównać złożoność różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest uniwersalny. Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów. Potrafi realizować nawet bardzo złożone

algorytmy, stosować procedury graficzne w realizacji skomplikowanych zadań – np. tworzyć własne animacje. Potrafi prezentować algorytmy rekurencyjne w postaci programu; potrafi zamienić rozwiązanie iteracyjne algorytmu na rekurencyjne. Zapisuje w postaci programu wybrane algorytmy sortowania. Opracowuje złożony program – umie podzielić zadania, ustalić sposoby przekazywania danych pomiędzy procedurami. Zabezpiecza tworzone programy przed wprowadzeniem przez użytkownika błędnych danych. Rozumie, na czym polega dobór struktur danych do algorytmu. Potrafi zastosować rekordowy typ danych. Potrafi omówić dokładnie działanie procesora. Potrafi wykonać dowolną konwersję pomiędzy systemem dziesiętnym, dwójkowym i szesnastkowym. Zna sposób zapisu liczby całkowitej i rzeczywistej (zmiennoprzecinkowej). Umie wymienić przynajmniej dwa systemy operacyjne i podać ich najważniejsze funkcje. Zna zaawansowane metody wyszukiwania i odzyskiwania plików. Zna przynajmniej jeden algorytm szyfrowania danych. Potrafi zaszyfrować i odszyfrować prosty tekst. Zna schemat działania sieci komputerowych. Potrafi wymienić zalety i wady różnych topologii sieci. Charakteryzuje topologie gwiazdy, magistrali i pierścienia. Zna podstawowe cechy systemu Linux. Umie z pomocą nauczyciela zrealizować małą sieć komputerową – skonfigurować jej składniki, udostępnić pliki, dyski, drukarki, dodać użytkowników. Przygotowuje analizę porównawczą, pokazującą na przestrzeni wielu lat rozwój informatyki, w tym sieci komputerowych, oraz multimediiów. Spełnia wszystkie wymagania na oceny niższe. Potrafi zapisać algorytmy klasyczne zawarte w podstawie programowej w postaci schematu blokowego i języka programowania.

Wie, jak tworzyć formularz z podformularzami, potrafi tworzyć makropolecenia, potrafi pisać procedury obsługi zdarzeń, potrafi budować środowisko zarządzania bazą danych, potrafi zabezpieczać bazę danych, potrafi stosować kwerendę składającą i definiującą dane, zna struktury listowe, wie, jak implementowane są drzewa binarne i potrafi je przeszukiwać, zna wybrane algorytmy dotyczące tekstów, potrafi stosować wybrane techniki rozszerzające funkcjonalność języka HTML, potrafi przetwarzać pliki tekstowe umieszczone w zasobach serwera www, potrafi budować proste aplikacje multimedialne

Celujący: Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Potrafi trafnie dobrać do algorytmu sposób prezentacji. Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi zaproponować własny pseudojęzyk (postać instrukcji i zasady składni). Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję. Zna trudniejsze algorytmy, np. algorytm trwałego małżeństwa, problem ośmiu hetmanów. Zna inne techniki sortowania, np. sortowanie przez scalanie ciągów i metodę szybką. Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu). Zna inne algorytmy numeryczne, np. wyznaczanie miejsca zerowego funkcji. Korzysta samodzielnie z dodatkowej literatury. Ocenia złożoność czasową i pamięciową algorytmu. Zna odpowiednie wzory. Ocenia efektywność działania programu. Wie, na czym polega programowanie obiektowe i zdarzeniowe.

Potrafi stosować techniki programowania dynamicznego lub programowania obiektowego. Zna i rozumie podobieństwa i różnice w strukturze programu zapisanego w różnych językach programowania – w deklaracji zmiennych i procedur, w składni i zasadach działania poszczególnych procedur. Sprawnie korzysta z dodatkowej, fachowej literatury. Zna dynamiczne struktury danych. Potrafi zastosować zmienne typu wskaźnikowego w zadaniach. Zna struktury listowe, np. stos, kolejkę, listę. Rozumie i potrafi zastosować typ obiektowy. Zna operacje logiczne na liczbach binarnych i przesunięcia bitowe. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Wykonuje sprawnie operacje na liczbach zapisanych w różnych systemach pozycyjnych. Potrafi odzyskać utracony plik, stosując zaawansowane metody. Potrafi omówić różne systemy operacyjne, wskazać ich najważniejsze funkcje. Samodzielnie wyszukuje informacje na temat kompresji i szyfrowania danych. Zna kilka sposobów szyfrowania informacji. Potrafi zapisać algorytm szyfrowania w postaci programu. Zna działanie algorytmu kompresji. Omawia szczegółowo model warstwowy sieci. Omawia różne systemy sieciowe. Dokonuje ich analizy porównawczej. Charakteryzuje system Linux. Potrafi samodzielnie zbudować małą sieć domową. Wskazuje tendencje w rozwoju informatyki i jej zastosowania, dostrzegając przeobrażenia w tej dziedzinie w kraju i na świecie. Spełnia wszystkie wymagania na oceny niższe. Stosuje algorytmy klasyczne zawarte w podstawie programowej do rozwiązywania zadań maturalnych.

5. Uczeń, który opuścił więcej niż 50% zajęć lekcyjnych nie może być klasyfikowany z przedmiotu.

6. Przy ustalaniu ocen brany jest pod uwagę wysiłek wkładany przez ucznia w wykonywanie obowiązków wynikających ze specyfiki przedmiotu.