Ośrodek Rozwoju Edukacji

Bernardeta Wójtowicz

Informatyka — moja pasja!

Program nauczania informatyki w zakresie rozszerzonym dla IV etapu edukacyjnego

Spis treści

Wstęp

Informatyka jest wszechobecna w otaczającym nas świecie tylko niekoniecznie zdajemy sobie z tego sprawę. Absolwent szkoły ponadgimnazjalnej, który poważnie myśli o wyborze studiów związanych z informatyką w takim czy w innym zakresie powinien być wyposażony w zestaw wiadomości i umiejętności, które pozwolą mu na w miarę bezproblemowe podjęcie studiów.

Opracowany przeze mnie program nauczania jest oparty na podstawie programowej kształcenia ogólnego dla szkół ponadgimnazjalnych, których ukończenie umożliwia uzyskanie świadectwa dojrzałości po zdaniu egzaminu maturalnego, zawartej w załączniku nr 4 ROZPORZĄDZENIA MINISTRA EDUKACJI NARODOWEJ z dnia 23 grudnia 2008 r. w sprawie podstawy programowej wychowania przedszkolnego oraz kształcenia ogólnego w poszczególnych typach szkół (Dz. U. z dnia 15 stycznia 2009 r.)

Pisząc informatyka mam na myśli informatykę na poziomie rozszerzonym wybieraną jako przedmiot dodatkowy. Ta terminologia będzie konsekwentnie stosowana w całym tym dokumencie, o ile nie zostanie wyraźnie zaznaczone inaczej.

Realizacja tego programu zapewnia zrealizowanie wszystkich treści zawartych w postawie programowej. Zgodnie z wymaganiami zakładam, że nauczyciel zrealizuje co najmniej 180 godzin informatyki łącznie w 2 i 3 klasie szkoły ponadgimnazjalnej.

Uczniowie, którzy będą realizowali program przedmiotu informatyka wybierają ten przedmiot jako dodatkowy, więc możemy założyć, że ten przedmiot leży w sferze ich zainteresowań. Są to uczniowie przeważnie drugiej i trzeciej klasy, bo zgodnie z założeniami reformy uczniowie wybierają przedmioty dodatkowe w pierwszej klasie aby zacząć ich naukę w klasie drugiej.

Uwagi na temat rozkładu nauczania

Proponowany przeze mnie program nauczania informatyki na poziomie rozszerzonym opiera się na dużej liczbie ćwiczeń, zadań wykonywanych przez uczniów.

Drugie założenie to brak powiązania z konkretnym podręcznikiem przygotowanym do nowej podstawy programowej. Jest to celowe działanie, ponieważ z jednej strony każdy podręcznik, który widziałam był przeładowany treścią, a z drugiej strony pragnąc uniknąć konieczności kupowania podręcznika, chcę oszczędzić wydatków rodzicom uczniów. Uczniowie mogą korzystać z dużo tańszych, być może odkupionych od starszych kolegów podręczników do starej podstawy programowej – oczywiście z treści wybranych z konkretnych rozdziałów. Inną metodą – często występującą - jest używanie podręczników, które być może znajdują się na wyposażeniu pracowni komputerowej w szkole.

Kolejna cecha mojego programu to korzystanie z ogólnodostępnych materiałów zamieszczonych w sieci Internet, a chodzi tutaj nie tylko o wprowadzane treści bardziej teoretyczne, ale także o dużą bazę ćwiczeń, filmów itp. Nauczenie uczniów właściwego, krytycznego korzystania z wiedzy zawartej w Internecie jest bardzo dużym wyzwaniem, które jednak warto podjąć.

Uczniowie rozpoczynając naukę informatyki na poziomie rozszerzonym są bardzo pewni swojej wiedzy i umiejętności – poruszają się w tym świecie od wielu lat, aczkolwiek niekoniecznie zdają sobie sprawę z faktu, że informatyka swoje początki ma w algorytmice i że bez dobrego algorytmu żaden program nie zadziała. Dlatego proponuję naukę informatyki zacząć właśnie od podstaw algorytmiki tzn. od specyfikacji problemu, schematów blokowych, a co także istotne od przyzwyczajania uczniów do zapisu programów w pseudojęzyku. W momencie, gdy rozpoczynamy naukę programowania w języku wyższego poziomu np. w C++, albo w Pascalu ważną rzeczą jest stopniowe rozbudowywanie programów. Uczniów cieszy każdy nawet najdrobniejszy sukces, dlatego rozpoczynając od szablonu programu, do którego stopniowo dopisujemy polecenia, w taki sposób, aby nie dodać zbyt dużo, ale aby efekt działania się zmienił jest bardzo istotny.

W podobny sposób wprowadzamy ułatwienia edytora takie jak np. podświetlanie dopasowanych klamer albo numerowanie wierszy.

W tym momencie jest dostępnych kilka podręczników do nauczania informatyki w zakresie rozszerzonym wg nowej podstawy programowej, aczkolwiek fragmenty obecnie obowiązującego podręcznika wydawnictwa OPERON – Informatyka, kształcenie w zakresie rozszerzonym byłby także odpowiedni, poza tym warto korzystać na lekcjach z zasobów portalu *dla nauczycieli* www.scholaris.pl gdzie znajduje się także narzędzie do tworzenia własnych materiałów. Oprócz tego bardzo dobre do wykorzystania są zasoby portali spoj.pl, informatyka.wroc.pl, main.edu.pl

Cele ogólne kształcenia informatyki na poziomie rozszerzonym w szkole ponadgimnazjalnej

1. Przyswojenie sobie przez uczniów określonego zakresu wiadomości z informatyki na poziomie rozszerzonym.
2. Rozumienie przez uczniów zasad działania poszczególnych algorytmów.
3. Zdobycie przez uczniów umiejętności stosowania własnych wiadomości do rozwiązywania zadań, wykonywania ćwiczeń, do planowania własnej pracy.
4. Rozwijanie myślenia abstrakcyjnego, umiejętności podziału zadań między członków grupy.
5. Rozwijanie krytycznego myślenia.
6. Rozwijanie umiejętności analizowania błędów swoich i innych uczniów, kształcenie umiejętności właściwego wyrażania opinii tak krytyki, jak i pochwały.
7. Kształcenie umiejętności:
   - czytania ze zrozumieniem, w tym analizowania treści zadania, formułowania specyfikacji problemu, wyodrębniania części składowych zadania,
   - stosowania narzędzi matematycznych w informatyce,
   - stosowanie narzędzi informatycznych w życiu codziennym,
   - krytycznego podejmowania decyzji w oparciu o informacje pochodzące z różnych źródeł,
   - wysnuwania wniosków na podstawie zaobserwowanych regularności w danych empirycznych,
   - wyszukiwania, wybierania i krytycznej analizy informacji,
   - używania poprawnego języka polskiego i ew. języków obcych we wszelkich pracach pisemnych i programach komputerowych,
   - sprawnego posługiwania się technologiami informacyjno – komunikacyjnymi,
   - sprawnej i sprawiedliwie podzielonej pracy w grupie.

Cele wychowania

1. Planowanie własnej przyszłości w tym kariery zawodowej – także w kontekście rozwoju informatyki.
2. Podkreślanie znaczenia uczciwości w każdym momencie procesu edukacyjnego, tępienie plagiatów i używania cytatów bez podania źródła.
3. Zachęcanie do nauki języków obcych i motywowanie do osiągania w nich biegłości.
4. Zwrócenie uwagi uczniów na problem przestępstw związanych z Internetem i innymi mediami, a także piractwa komputerowego. // kwestia legalności oprogramowania.
5. Kształtowanie właściwych postaw prozdrowotnych, propagowanie odpowiedniej diety i aktywności fizycznej, aby przeciwdziałać nasilającemu się problemowi otyłości wśród dzieci i młodzieży.

6. Zachęcanie uczniów do pomocy innym uczniom, a także innym ludziom będącym w potrzebie – próba zapobiegania różnego typu dyskryminacji.

Cele kształcenia – wymagania ogólne

zawarte w podstawie programowej dla IV etapu edukacyjnego do przedmiotu informatyka

I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.

Uczeń sprawnie posługuje się komputerem i programami w nimi zainstalowanymi. Potrafi zmodyfikować ustawienia komputera, doinstalować potrzebne programy, z dbałością o prawa autorskie i istniejącą licencję. Uczeń stosuje technologie informacyjno-komunikacyjne do przygotowywania różnego typu prac dotyczących przedmiotów szkolnych, sprawnie wymienia informacje i wiadomości z poszanowaniem Netykiety oraz używając poprawnego języka.

II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.

Uczeń sprawnie i efektywnie wyszukuje potrzebne mu informacje korzystając z różnorodnych źródeł. Potrafi je przeanalizować, wybrać najbardziej adekwatne, właściwie je przedstawić i zilustrować diagramami czy też wykresami. Potrafi przekonująco zaprezentować wyniki swojej pracy.

III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.

Uczeń potrafi właściwie zaplanować rozwiązanie problemu algorytmicznego, od specyfikacji problemu, poprzez stworzenie algorytmu zapisanie go w odpowiednim języku programowania, sprawdzenie poprawności oraz przeanalizowanie efektywności działania.

IV. Wykorzystanie komputera oraz programów i gier edukacyjnych do poszerzania wiedzy i umiejętności z różnych dziedzin oraz do rozwijania zainteresowań.

Uczeń potrafi wyszukać i dobrać odpowiedni program edukacyjny, aby poszerzyć swoją wiedzę czy też rozwinąć zainteresowania – np. służący do nauki języków obcych czy też innego języka programowania, tutorial dotyczący programu graficznego, itp.

V. Ocena zagrożeń i ograniczeń, docenianie społecznych aspektów rozwoju i zastosowań informatyki.

Uczeń właściwie ocenia zagrożenia płynące z pobierania plików z niesprawdzonych źródeł, rozumie kwestię legalności oprogramowania, wie o potrzebie używania aktualnych wersji programów, programu antywirusowego, zapory ogniowej.

Propozycja ramowego rozkładu materiału

Łączna liczba godzin do wykorzystania w klasie drugiej i trzeciej szkoły ponadgimnazjalnej wynosi co najmniej 180 i na tyle godzin jest zaplanowany poniższy program. Można założyć, że w klasie drugiej będzie to 120 godzin, a w klasie trzeciej 60 godzin. Obok nazwy działu podano po pauzie liczbę godzin przeznaczoną na zrealizowanie danej partii materiału:

Algorytmika - 14 godzin
Programowanie strukturalne i obiektowe - 67 godzin
Wprowadzenie do baz danych - 14 godzin
Relacyjne bazy danych – 21 godzin

Sieci komputerowe i protokoły sieciowe - 19 godzin
Kryptografia, kryptoanaliza, bezpieczeństwo danych – 24 godziny
Elementy grafiki komputerowej - 21 godzin

Treści nauczania – wymagania szczegółowe, propozycja szczegółowego rozkładu materiału.

Algorytmika – 14 godzin

Specyfikacja problemu algorytmicznego – 2 godz.
Zapisywanie prostych algorytmów liniowych za pomocą listy kroków, pseudojęzyka i schematu blokowego. - 2 godz.
Schematy rozgałęzione i algorytmy złożone – ćwiczenia.- 2 godz.
Algorytmy iteracyjne. - 2 godz.
Ocena złożoności obliczeniowej algorytmu oraz zgodności ze specyfikacją. Formułowanie własnych problemów algorytmicznych. - 2 godz.
Powtórzenie i uzupełnienie wiadomości z algorytmiki, ćwiczenia. - 2 godz.
Sprawdzian wiadomości i jego omówienie – 2 godz.

Programowanie strukturalne i obiektowe – 67 godzin

Pierwszy program w C++. Szkielet programu. - 1 godz.
Podstawowe typy zmiennych. - 2 godz.
Instrukcja warunkowa i jej zastosowanie w programach. - 2 godz.
Instrukcja warunkowa – ćwiczenia (np. sprawdzanie warunku trójkąta, obliczanie wartości bezwzględnej) - 2 godz.
Jednoczesne znajdowanie elementu największego i najmniejszego w zbiorze.- 1 godz.
Instrukcja switch i jej zastosowanie np. w programie do obliczania pól figur z menu. - 2 godz.
Wprowadzenie do iteracji, algorytm iteracyjny – schemat blokowy. - 2 godz.
Wprowadzenie pętli for – ćwiczenia. - 2 godz.
Wprowadzenie pętli do…while – ćwiczenia. - 2 godz.
Wprowadzenie pętli while – ćwiczenia. - 2 godz.
Ćwiczenia w stosowaniu różnych pętli, różnice między nimi. - 2 godz.
Kartkówka/sprawdzian z pętli i jej omówienie. - 2 godz.
Wprowadzenie do tablic – tablice dwuwymiarowe. - 2 godz.
Wprowadzanie danych do tablicy, wypisywanie zawartości tablicy. - 2 godz.
Tablice dwuwymiarowe – ćwiczenia. - 2 godz.
Tablice znaków - 1 godz.
Wprowadzenie do funkcji - 1 godz.
Zastosowanie funkcji w prostych programach - 2 godz.
Funkcje i ich zastosowanie – ćwiczenia. - 2 godz.
Wprowadzenie do rekurencji. Algorytm rekurencyjny. - 1 godz.
Obliczanie silni za pomocą rekurencji i iteracji. - 2 godz.
Obliczanie wartości wielomianu za pomocą Schematu Hornera iteracyjnie i rekurencyjnie. - 2 godz.
Obliczanie potęgi liczby całkowitej za pomocą rekurencji i iteracji. - 2 godz.
Wprowadzenie do sortowania, różne algorytmy sortujące porównanie efektywności algorytmów na podstawie narzędzi zamieszczonych w Internecie. - 2 godz.
Sortowanie przez wybór - algorytm i jego realizacja. Sortowanie tablicy przez wybór. - 2 godz.
Sortowanie przez wstawianie algorytm i jego realizacja. - 2 godz.
Sortowanie przez scalanie algorytm i jego realizacja. - 2 godz.
Porównanie złożoności obliczeniowej różnych algorytmów poznanych na lekcji, a także inne sposoby sortowania np. szybki, kubełkowy, kopcowy – 2 godz.
Struktury – wprowadzenie. - 1 godz.
Stosowanie struktur w prostych zadaniach algorytmicznych. - 1 godz.
Zapis do pliku i odczyt z pliku. - 2 godz.

Złożone zadania algorytmiczne, pobieranie danych z pliku tekstowego, rozwiązanie problemu, zapis wyników do pliku. - 2 godz.
Rozwiązywanie układu 2 równań z 2 niewiadomymi za pomocą metody wyznaczników. - 2 godz.
Algorytm obliczania wartości pierwiastka kwadratowego, realizacja tego algorytmu w arkuszu kalkulacyjnym i poprzez program. – 2 godz.
Elementy programowania obiektowego – pojęcie klasy, dziedziczenia, rola konstruktora i destruktora, struktury dynamiczne na przykładzie obiektu wektor i opis działań na wektorach. – 4 godz.
Sprawdzian wiadomości i jego omówienie – 2 godz.

Wprowadzenie do baz danych – 14 godzin

Tabela w edytorze tekstowym lub arkuszu kalkulacyjnym jako najprostsza baza danych. - 1 godz.
Funkcje w arkuszu kalkulacyjnym – rysowanie wykresów. - 1 godz.
Formatowanie warunkowe, formuły w arkuszu kalkulacyjnym. - 1 godz.
Import danych z pliku tekstowego do arkusza. - 1 godz.
Baza danych w arkuszu kalkulacyjnym – operacje na danych: sortowanie, filtr, autofiltr. - 1 godz.
Baza danych w arkuszu kalkulacyjnym – sumy pośrednie. - 1 godz.
Opracowywanie danych i ich graficzna reprezentacja. - 1 godz.
Tworzenie formularza w edytorze tekstowym i arkuszu kalkulacyjnym, zabezpieczenie przed edycją i przed niepowołanym dostępem. - 2 godz.
Formularze na stronach www. - 1 godz.
Przygotowywanie gry – bazy danych - typu „Zgadnij Kto" – praca zespołowa – 2 h
Proste bazy danych – sprawdzian wiadomości i jego omówienie. – 2 godz.

Relacyjne bazy danych – 21 godzin

Pierwsza baza danych w programie Access. - 1 godz.
Import danych do bazy z pliku tekstowego, użycie separatorów, przygotowanie arkusza do eksportu danych. - 1 godz.
Tworzenie tabeli za pomocą widoku projektu, różne typy pól, dobór odpowiedniego typu pola, klucz podstawowy. - 1 godz.
Relacje między tabelami, typy relacji, klucz obcy. - 2 godz.
Wprowadzanie danych do tabeli, maska wprowadzania danych.- 1 godz.
Projektowanie formularza. - 1 godz.
Filtrowanie danych - kartkówka, tworzenie tabel i relacji. - 1 godz.
Wprowadzenie do kwerend. Kwerendy wybierające. - 1 godz.
Kwerendy parametryczne. - 1 godz.
Kwerendy modyfikujące. - 1 godz.
Kwerendy krzyżowe. - 1 godz.
Kwerendy – widok SQL. - 1 godz.
Tworzenie raportów. - 1 godz.
Bazy danych – ćwiczenia. - 2 godz.
Tworzenie własnej bazy danych -projekt – np. nt. wyższych uczelni w Polsce, albo tanich hoteli w Polsce, itp. – 2 godz.
Bazy danych – powtórzenie. - 1 godz.
Sprawdzian z baz danych i jego omówienie.- 2 godz.

Sieci komputerowe i protokoły sieciowe – 19 godzin

Budowa komputera – przypomnienie i uzupełnienie wiadomości. - 1 godz.
Sposoby reprezentowania różnych form informacji w komputerze – np liczb, dźwięków. - 2 godz.
Podstawowe systemy operacyjne. BIOS. Funkcje systemu operacyjnego. - 1 godz.
Najważniejsze dystrybucje systemu Linux. - 1 godz.
Podstawowe polecenia systemu Linux. - 1 godz.
Najprostsza sieć komputerowa, podstawowe urządzenia sieciowe. - 1 godz.
Połączenia sieciowe, topologie sieci. - 1 godz.
Adres IP, maska sieci, adres rozgłoszeniowy. - 1 godz.
Warstwowy model sieci komputerowej. - 1 godz.

Protokoły sieciowe. - 1 godz.
Wymiana informacji w sieci komputerowej. - 1 godz.
Tablice routingu. - 1 godz.
Opis sieci komputerowej w pracowni szkolnej. - 1 godz.
Zasady administrowania siecią komputerową. - 1 godz.
Projektowanie własnej sieci komputerowej – ćwiczenie. - 2 godz.
Sieci komputerowe – kartkówka i jej omówienie. - 1 godz.

Kryptografia, kryptoanaliza, bezpieczeństwo danych – 24 godziny

Wstęp do kryptografii – rys historyczny. - 1 godz.
Szyfr Cezara i jego komputerowa realizacja. - 1 godz.
Uogólniony szyfr Cezara, alfabet jawny i szyfrowy. - 1 godz.
Szyfr płotkowy. - 1 godz.
Szyfr polialfabetyczny, np. Viegnere'a- 1 godz.
Szyfr Playfair. - 1 godz.
Realizacja wybranego szyfru za pomocą programu. - 1 godz.
Klucz publiczny- 1 godz.
Szyfrowanie danych podczas przesyłania. - 1 godz.
Szyfrowanie i deszyfrowanie, przygotowanie własnego programu szyfrującego i deszyfrującego z odpowiednim menu i interfejsem. – 4h
Programy szyfrujące i deszyfrujące. - 1 godz.
Kompresja danych stratna i bezstratna. - 1 godz.
Wirusy komputerowe i inne złośliwe oprogramowanie. - 1 godz.
Algorytm Huffmana- 1 godz.
Bezpieczeństwo danych, certyfikat, zapora ogniowa. - 1 godz.
Bezpieczeństwo komputera - program antywirusowy. - 1 godz.
Rozwój informatyki i perspektywy nauki. - 1 godz.
Konkurs „Łamacze szyfrów". 2 godz.
Sprawdzian i jego omówienie. - 2 godz.

Elementy grafiki komputerowej – 21 godzin

Podstawowe modele barw. - 1 godz.
Grafika rastrowa a wektorowa - porównanie. - 1 godz.
Podstawowe formaty plików graficznych, kompresja - 1 godz.
Zmiana wymiarów obrazu, zmiana rozdzielczości. - 1 godz.
Wprowadzenie do programu GIMP – podstawowe części okna programu, pokazywanie i ukrywanie poszczególnych okien, dostosowanie okna programu do własnych potrzeb i upodobań. - 1 godz.
Edycja pierwszego obrazu, desaturacja, wstawienie prostego napisu, uzyskanie efektu cienia. - 1 godz.
Rozjaśnienie zdjęcia, modyfikacja kolorów, progowanie. - 1 godz.
Wstawianie napisów na poszczególnych warstwach, operacje na warstwach, ukrywanie, duplikowanie, zmiana kolejności warstw. - 1 godz.
Wyrównywanie warstw, różne efekty uzyskiwane za pomocą wyrównywania warstw. - 1 godz.
Zmiana wymiarów płótna, skalowanie warstw, komponowanie obrazu złożonego z kilku innych obrazów. - 1 godz.
Projekt pocztówki okolicy albo najbliższego dużego miasta. - 1 godz.
Stosowanie rozmycia i różne efekty uzyskiwane w ten sposób. - 1 godz.
Zastosowanie filtrów. - 1 godz.
Obramowanie warstw. - 1 godz.
Selekcja obrazu. - 1 godz.
Korzystanie z kanałów podczas edycji obrazu. - 1 godz.
Zastosowanie skryptów do modyfikacji obrazu. - 1 godz.
Komponowanie własnych zdjęć - fotomontaż - 1 godz.
Przygotowanie projektu folderu szkoły. - 1 godz.
Przygotowanie własnego filmu za pomocą telefonu komórkowego i przetworzenie go za pomocą bezpłatnego programu komputerowego. - 2 godz.

Propozycje oczekiwanych osiągnięć uczniów po realizacji poszczególnych działów

Algorytmika

Uczeń potrafi:

- analizować, modelować i rozwiązać sytuacje problemowe z różnych dziedzin;
- stosować podejście algorytmiczne do rozwiązywania problemu;
- sformułować przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;
- opisać podstawowe algorytmy i zastosować:
    - algorytmy na liczbach całkowitych, takich jak np. reprezentacja liczb w dowolnym systemie pozycyjnym, np. w dwójkowym i szesnastkowym, obliczanie pola obszarów zamkniętych,
    - algorytmy badające własności geometryczne, np.: sprawdzanie warunku trójkąta, badanie położenia punktów względem prostej;
- opisać własności algorytmów na podstawie ich analizy;
- ocenić zgodność algorytmu ze specyfikacją problemu;
- obliczyć liczbę operacji wykonywanych przez algorytm.

Programowanie strukturalne i obiektowe

Uczeń potrafi:

- zastosować algorytmy wyszukiwania i porządkowania (sortowania), np.: jednoczesne znajdowanie największego i najmniejszego elementu w zbiorze; algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie;
- zastosować algorytmy na liczbach całkowitych;
- sprawdzić, czy liczba jest liczbą pierwszą, doskonałą;
- przedstawić iteracyjną i rekurencyjną realizację algorytmu Euklidesa;
- obliczyć iteracyjnie i rekurencyjnie wartości liczb Fibonacciego;
- stosować algorytmy numeryczne, np.: do obliczanie wartości pierwiastka kwadratowego;
- obliczać wartości wielomianu za pomocą schematu Hornera;
- zastosować algorytmy na tekstach, np.: do sprawdzania, czy dany ciąg znaków tworzy palindrom, anagram, do porządkowania alfabetycznego;
- wyszukiwać wzorzec w tekście;
- projektować rozwiązanie problemu (realizację algorytmu) i dobierać odpowiednią strukturę danych;
- stosować metodę zstępującą i wstępującą przy rozwiązywaniu problemu;
- dobierać odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne;
- stosować zasady programowania strukturalnego i modularnego do rozwiązywania problemu;
- dobierać efektywny algorytm do rozwiązania sytuacji problemowej i zapisać go w wybranej notacji;
- posługiwać się podstawowymi technikami algorytmicznymi;
- oceniać własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;
- stosować rekurencję w prostych sytuacjach problemowych;
- opracować i przeprowadzać wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;
- posługiwać się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;

- stosować podejście zachłanne w rozwiązywaniu problemów;
- przeprowadzać komputerową realizację algorytmu i rozwiązania problemu;
- sprawnie posługiwać się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów;
- stosować podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzyć strukturę programu;
- dobierać najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu;
- dobierać właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;
- oceniać poprawność komputerowego rozwiązania problemu na podstawie jego testowania;
- szacować wielkość pamięci potrzebnej do komputerowej realizacji algorytmu;
- badać efektywność komputerowych rozwiązań problemów;
- wyjaśniać źródło błędów w obliczeniach komputerowych (błąd względny, błąd bezwzględny);
- realizować indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu;
- korzystać z zasobów edukacyjnych udostępnianych na portalach przeznaczonych do kształcenia na odległość.

Wprowadzenie do baz danych

Uczeń potrafi:

- wykorzystać arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów;
- wykorzystywać zasoby i usługi sieci komputerowych w komunikacji z innymi użytkownikami, w tym do przesyłania i udostępniania danych;
- wziąć udział w dyskusjach w sieci.

Relacyjne bazy danych

Uczeń potrafi:

- zaprojektować relacyjną bazę danych z zapewnieniem integralności danych;
- stosować metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
- stworzyć aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnić integralność danych na poziomie pól, tabel, relacji;
- znaleźć odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin.

Sieci komputerowe i protokoły sieciowe

Uczeń potrafi:

- przedstawić sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;
- wyjaśnić funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne;
- przedstawić warstwowy model sieci komputerowych, określić ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisać zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługiwać się terminologią sieciową, korzystać z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;

- opisać możliwości nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi;
- zdobyć informacje dotyczące nowych programów i systemów oprogramowania.

Kryptografia, kryptoanaliza, bezpieczeństwo danych.
Uczeń potrafi:
- zastosować algorytmy kompresji i szyfrowania, np.: kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana, szyfr Cezara, szyfr przestawieniowy, szyfr z kluczem jawnym;
- podpis elektroniczny;
- opisać mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora ogniowa;
- omówić zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci;
- opracować indywidualne i zespołowe projekty przedmiotowe i między przedmiotowe z wykorzystaniem metod i narzędzi informatyki;
- opisać najważniejsze elementy procesu rozwoju informatyki i technologii informacyjno-komunikacyjnych;
- wyjaśnić szanse i zagrożenia dla rozwoju społecznego i gospodarczego oraz dla obywateli, związane z rozwojem informatyki i technologii informacyjno-komunikacyjnych;
- zastosować normy etyczne i prawne związane z rozpowszechnianiem programów komputerowych, bezpieczeństwem i ochroną danych oraz informacji w komputerze i w sieciach komputerowych;
- przygotować się do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego.

Elementy grafiki komputerowej
Uczeń potrafi:
- opisać podstawowe modele barw i ich zastosowanie;
- określić własności grafiki rastrowej i wektorowej oraz scharakteryzować podstawowe formaty plików graficznych;
- tworzyć i edytować obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;
- przetwarzać obrazy i filmy, np.: zmieniać rozdzielczość, rozmiar, model barw, stosować filtry.

Procedury osiągania celów

Zaczynając nauczanie informatyki wg nowej podstawy programowej należy pamiętać, że ten przedmiot wybrali sami uczniowie jako swój dodatkowy przedmiot, dlatego można założyć, że spora ich część jest juz na wstępie zainteresowana przedmiotem. To zainteresowanie należy podtrzymać i rozwinąć, aby korzyści ze współpracy były jak największe dla obu stron: i dla uczniów, i dla nauczyciela.

Informatyka jest specyficznym, bardzo szybko rozwijającym się przedmiotem i nic w tym dziwnego, że uczniowie są z reguły bardziej na bieżąco z nowikami informatycznymi niż nauczyciel. Dlatego też nauczyciel informatyki powinien stale się dokształcać i poszerzać swoją wiedzę. Nie chodzi tutaj bynajmniej o kursy doskonalące, ale o wiedzę, którą czerpią uczniowie, którzy mnóstwo czasu wolnego spędzają przed komputerem przeszukując Internet.

Warto wg mnie rozważyć poświęcanie określonego czasu na lekcji – cyklicznie – raz czy dwa razy w tygodniu na tzw. nowinki informatyczne – czasami coś ciekawego przeczytają uczniowie,

czasami nauczyciel. Taka wymiana informacji pokazuje uczniom, że nauczyciel też jest zaangażowany w przedmiot, którego uczy.

Kolejna rzecz to nieomylność nauczyciela – wg mnie w przypadku informatyki nie ma takiego pojęcia. Zawsze może znaleźć się uczeń, którego wiedza przewyższa wiedzę nauczyciela w pewnym obszarze. Należy wtedy taką sytuację wykorzystać proponując uczniowi przygotowanie referatu, kółka, specjalnych zajęć poświęconych jego pasji.

Na lekcjach warto walczyć ze stereotypami – że na informatyce można sobie pograć i odpocząć a uczyć już się nie trzeba. Należy zacząć już od pierwszej lekcji jasno ustalając zasady pracy. Jeśli np. wymagamy prowadzenia zeszytu – egzekwujmy to, można też zasugerować uczniom prowadzenie zeszytu w wersji elektronicznej, albo w edytorze tekstu, albo np. bloga czy forum dyskusyjnego. Daje nam to okazję do indywidualizowania procesu nauczania, bo jednemu uczniowi wystarczą notatki w zeszycie, a inny uczeń wykaże się tworząc rozbudowaną i pięknie ilustrowaną stronę www.

Na lekcjach pozwalajmy uczniom, w miarę możliwości, na wymianę doświadczeń, dyskusje na temat zadania oczywiście.

Weźmy pod uwagę fakt, że niektórzy uczniowie najlepiej pracują gdy się ruszają, więc jeśli to nie przeszkadza innym, a uczeń chce przejść się po sali myśląc – można na to wyrazić zgodę zwracając oczywiście uwagę na bezpieczeństwo i higienę pracy w pracowni komputerowej – np. informując, że bieganie nie jest dopuszczalne.

Pozwólmy uczniom na kreatywność i na szukanie własnych metod rozwiązania problemu.

Należy tak prowadzić lekcje, aby trafiały do uczniów z różnymi stylami uczenia się – możemy tworzyć rysunki, układać piosenki, itp.

Ostatnie, ale nie najmniej ważne jest nastawienie nauczyciela – jeśli nauczyciel lubi to co robi i robi to z pasją, wtedy może nią zarazić uczniów i to jest klucz do sukcesu!

Uwagi nt. realizacji poszczególnych działów

Naukę informatyki możemy rozpocząć od zapytania uczniów: z czym kojarzy im się ten przedmiot, od sprawdzenia jakie są wyobrażenia uczniów, czego wg nich będą się uczyć. Koniecznie należy zaplanować lekcję organizacyjną, na której zapoznamy uczniów z wymaganiami edukacyjnymi, sposobami oceniania i sprawdzania osiągnięć edukacyjnych. Należy także koniecznie zapoznać uczniów z przepisami BPH w pracowni komputerowej.

Algorytmika

W tej części wprowadzającej do przedmiotu informatyka proponujemy omówienie tylko podstawowych zagadnień związanych z algorytmiką, a kolejne bardziej skomplikowane wprowadzać równocześnie, albo naprzemiennie z nauką programowania. Ćwiczenia w tym dziale mogą dotyczyć obliczania pierwiastka kwadratowego z dowolnej liczby rzeczywistej, wraz z podkreśleniem faktu, że algorytm powinien dla wprowadzonej liczby ujemnej zwrócić komunikat o błędnych danych wejściowych.

Inne przykłady to obliczanie wartości funkcji danej dwu lub trzy normowo, obliczanie pola kwadratu, czy też pól innych prostych figur geometrycznych. Warto przyzwyczaić uczniów do zapisywania programów w pseudojęzyku, aby później łatwiej było uczniom programować w języku wyższego poziomu np. w C++ albo w Pascalu, czy też w Javie.

Przygotowując lekcje warto skorzystać z bezpłatnych zasobów portalu scholaris.pl - możemy tam znaleźć np. gotowe ilustracje zawierające schematy blokowe – tylko należy je wcześniej dokładnie sprawdzić przed pokazaniem uczniom, bo niektóre zawierają błędy rzeczowe. Oprócz schematów we wspomnianym portalu można znaleźć kartę pracy dotyczącą algorytmów liniowych.

Programowanie strukturalne i obiektowe

Proponuję przy wprowadzaniu do programowania wykorzystać metodę gąbki i tablicy albo odpowiednio przygotowanej prezentacji. Umieszczamy zapisany w pseudojęzyku prosty program i wymazujemy poszczególne części programu zastępując je poleceniami z języka programowania w podobny sposób można przygotować prezentację multimedialną. Ten fakt wymazywania albo zastępowania jest dla uczniów o tyle znaczący, że zauważają związek z instrukcjami z pseudojęzyka a instrukcjami z języka C++.

Przy programowaniu warto zaczynać od jak najprostszych programów np. takich, które tylko wypisują tekst. Potem rozbudowujemy program dopisując pytanie o wiek, imię, miejsce urodzenia i stopniowy wprowadzamy różne typy zmiennych. Już w tym momencie możemy zindywidualizować proces nauczania, bo niektórzy uczniowie zaczną samodzielnie rozbudowywać program łącząc krótkie programy w większą całość, można zaproponować im temat takiego większego programu opartego na najprostszych poleceniach. Jednak niektórzy uczniowie zaczną zadawać pytania „bardzo do przodu". Nie hamujmy tej naturalnej ciekawości – proponuję odpowiedzieć na jedno-dwa pytania, a potem wskazać źródło odpowiedzi, pożyczyć podręcznik czy inną książkę o programowaniu, wskazać tutorial w Internecie, podpowiedzieć gdzie znajdziemy ciekawe zadania od najprostszych po naprawdę trudne. Warto zachęcić uczniów do samodzielnej pracy na stronach www.spoj.pl, czy też main.edu.pl gdzie automatyczna sprawdzarka będzie czuwać nad poprawnością wyników.

Należy trzymać się zasady, że na lekcji o konkretnym temacie bezwzględnie staramy się, aby każdy uczeń zrozumiał treści z bieżącej lekcji – czyli najpierw troszczymy się o najsłabszych, czy też o uczniów, którzy mają trudności w zrozumieniu danego pojęcia. Dużo ambitniejszym uczniom można zaproponować pomoc słabszym uczniom w opanowaniu danej partii materiału.

Warto też stosować się do zasady, że myszka i klawiatura jest tego, kto przy danym stanowisku siedzi – tzn. mogę podpowiedzieć gdzie, co napisać, czy też gdzie należy kliknąć, ale te działania uczeń musi wykonać samodzielnie.

Do lekcji programowania można wykorzystać także zasoby portalu scholaris.pl tam znajduje się m. in. karta pracy o przekazywaniu parametrów przez referencję czy też dotycząca rekurencji, albo algorytmów zachłannych. Na wspomnianym portalu znajdziemy scenariusze lekcji dotyczące algorytmiki, liczb Fibonacciego, ale także animację dotyczącą programowania obiektowego. Uczniowie zainteresowani tematem, a posługujący się biegle językiem angielskim mogą oglądnąć wykład z Harvardu w języku angielskim a dotyczący np. wyszukiwania binarnego i trójkąta Sierpińskiego http://www.naharvard.pl/dziedzina/informatyka.html.

Wprowadzenie do baz danych

Biorąc pod uwagę fakt, że w otaczającym nas świecie występuje coraz więcej baz danych, którymi uczniowie powinni się biegle posługiwać warto podać klika przykładów baz danych, które uczniowie na pewno znają, ale nie mają tej świadomości. Najprostszą bazą danych jest np. lista uczniów w dzienniku. Podobna tabelka utworzona w arkuszu kalkulacyjnym pozwala na sortowanie danych, ich filtrowanie i tworzenie ich ilustracji graficznej. Taki wstęp pozwoli uczniom łatwiej i łagodniej przejść do relacyjnych baz danych. Pracując z uczniami w tym dziale warto wykorzystać kartę pracy dotyczącą korespondencji seryjnej, a dostępną na stronie scholaris.pl

Ponadto uczniowie mogą przygotować własną bazę danych zawierającą np. spis stron do tanich linii lotniczych, albo mogą pracować metodą projektu i wyszukać najtańszą wycieczkę z miejsca zamieszkania do Warszawy, Krakowa czy Gdańska razem wyszukaniem najtańszego transportu, mieszkania i opłat za wstępy do zabytków. Porównując prace poszczególnych grup nauczyciel powinien docenić wkład pracy poświecony na wyszukanie informacji i ich weryfikację.

W arkuszu kalkulacyjnym można przygotować z uczniami komputerową wersję gdy „Zgadnij kto" – uczniowie w kolejnych wierszach umieszczają imiona 24 osób, a w kolejnych kolumnach ich cechy opisujące wygląd, np kolor włosów, kolor oczu, posiadanie nakrycia głowy, okularów, kolczyków, płeć itp. Jeśli baza jest odpowiednio przygotowana to jedna osoba wybiera losowo numer albo zdjęcie czy tez rysunek osoby a potem w jak najmniejszej liczbie prób ma zgadnąć kto to jest.

Przykładowe pytanie „czy to jest kobieta" pozwoli ustawić autofiltr w kolumnie płeć odpowiednio i zawęzić obszar wyszukiwania. Uczniów wciąga rywalizacja – a gdy mogą się jeszcze wykazać dodatkowo zdolnościami graficznymi i tworzą wizerunki poszczególnych osób – jest jeszcze ciekawiej.

Relacyjne bazy danych

Pracę nad relacyjnymi bazami danych warto rozpocząć od zaprezentowania uczniom gotowej bazy i pokazać jej poszczególne elementy. Można skorzystać z karty pracy dostępnej na stronie scholaris.pl a dotyczącej projektowania baz danych w Accessie. Na kolejnych zajęciach dotyczących importu danych z arkusza kalkulacyjnego warto pozwolić uczniom na popełnianie błędów. Gdy raz przekonają się, że arkusz należy przygotować do eksportu usuwając puste początkowe wiersze i kolumny – taką wiedzę zdobytą samodzielnie zapamiętają na dłużej. Pracując nad kolejnymi elementami baz danych można skorzystać z filmu video dotyczącego kwerend aktualizujących dostępnego na scholaris.pl.

Uczniowie chętnie poznają widok SQL kwerend wcześniej już utworzonych i zauważają z reguły, że czasami łatwiej zmodyfikować kwerendę właśnie korzystać z tego widoku. Szczególnie zainteresowani tematem są uczniowie, którzy prowadzą własne strony www oparte na php wtedy odkrywają, że wiedzą jak zmodyfikować w prostszy sposób kwerendy.

Sieci komputerowe i protokoły sieciowe

Sieci komputerowe to dział bardziej teoretyczny niż wcześniejszy – uczniowie tego nie lubią, ale jeśli właściwie wprowadzimy ten temat okaże się, że zainteresowanie uczniów wzrośnie. Można zacząć od pytania jakie urządzenia sieciowe zauważają w pracowni czy te w budynku szkoły; ile komputerów wystarczy aby powstała sieć, jaka jest topologia sieci w pracowni. Takie zaintrygowanie tematem to dobry punkt wyjścia do nauki.

Można rozdzielić wśród uczniów referaty dotyczące poszczególnych tematów, a potem sprawdzić zrozumienie treści przez krótkie kartkówki czy też konkurs.

Żmudne obliczanie adresu broadcast szybko sprawdzimy korzystając z kalkulatora on-line http://42.pl/ipcalc/.

Uczniowie mogą rozwiązywać krzyżówkę na temat systemów operacyjnych, wypełnić kartę pracy dotyczącą Linuksa. Czasami nie możemy pokazać któregoś z gniazd komputera albo innej jego składowej części – wtedy warto skorzystać z gotowych zdjęć. Uczniowie mogą też obejrzeć film video o samodzielnym wykonaniu kabla sieciowego, zasadzie działania dysku twardego, itp – a wszystkie te materiały są dostępne na scholaris.pl

Rywalizacja jest dobrą zachętą do nauki, dlatego możemy zaproponować uczniom udział w konkursie Dialnetmasters – a jeśli okaże się, że w danym roku ten konkurs akurat nie jest dostępny warto zrobić jego szkolną wersję. Przykładowe pytania z poprzednich edycji można znaleźć na stronach

http://r.gierwialo.com/2008/03/16/dialnet-masters-etap-ii-pytania/

http://i52.tinypic.com/34t4m76.png

Sieci komputerowe mogą być fascynujące, jeśli tylko ten temat właściwie przedstawimy naszym uczniom.

Kryptografia, kryptoanaliza, bezpieczeństwo danych

Szyfrowanie z reguły podoba się uczniom. Nic dziwnego owiane mgiełką tajemnicy afery szpiegowskie i złamanie tajemnicy Enigmy to atrakcyjne tematy.
Możemy skorzystać z gotowych e-lekcji dotyczących kryptografii i kryptoanalizy
http://www.scholaris.pl/frontend,4,94599.html
podobnych kart pracy o z tego tematu, a także obejrzeć film video o kryptografii symetrycznej.
Za pomocą narzędzi on-line możemy sprawdzić nasze zaszyfrowane wiadomości
http://www.webmaster.net.pl/narzedzia_online/szyfrowanie_rot13.php
Wielbicieli Enigmy zachęćmy do udziału w konkursie Łamacze Szyfrów, a jeśli byłby niedostępny w danym roku szkolnym to na stronie http://www.przygoda.nazwa.pl/LS/userfiles/file/Wyklad_1.pdf jest dostępny bezpłatnie pierwszy wykład z tego konkursu. Gdy mowa o prawie autorskim i jego poszanowaniu możemy skorzystać ponownie z portalu scholaris i gotowego scenariusza lekcji.

Elementy grafiki komputerowej

Ten wyjątkowo przyjemny dział warto wzbogacić scenariuszami lekcji zaczerpniętymi z portalu scholaris a dotyczącego modeli barw cmyk, rgb. Jeśli nasi uczniowie są już wyjątkowo biegli np. w programie GIMP można ich zachęcić do zapoznania się z możliwościami programu do tworzenia obiektów w 3D – korzystając z bezpłatnych tutoriali i testowych wersji.

Metody kontroli i oceny

Ocenianie powinno mieć na celu głównie przekazanie informacji uczniowi w jakim stopniu opanował wymagane do tej pory wiadomości i umiejętności.

Każda aktywność ucznia powinna być oceniona i warto podkreślić, że pochwała to także ocena. Uczniowie lubią być doceniani i dobrze, aby nauczyciel był nastawiony na systematyczne ocenianie i to różnych form pracy uczniów. Przy tak dużej liczbie godzin w drugiej klasie każdy uczeń w semestrze powinien mieć minimum 3 oceny i to z różnych form aktywności. Oceny z informatyki uczeń może uzyskać ze sprawdzianów pisemnych tradycyjnych i tych wykonywanych na komputerze.

Dużą podpowiedzią dla nauczyciela podczas wystawiania oceny rocznej, czy też semestralnej jest stosowanie średniej ważonej ocen. Konieczne jest podkreślenie faktu, że średnia ważona ma być dla nauczyciela sugestią, a nie jedynym wyznacznikiem przypisującym konkretną ocenę uczniowi.

Przypomnijmy, że sprawdzian powinien obejmować większą partię materiału – zaleca się aby był zapowiedziany minimum tydzień przed planowanym sprawdzianem, a jeśli to możliwe nawet 2 tygodnie. Integralną częścią przygotowania ucznia do sprawdzianu powinna być lekcja powtórzeniowa poprzedzająca tę dużą pracę pisemną. Proponuję, aby każdy element składowy sprawdzianu był punktowany – dobrze, jeśli uczniowie znają punktację w trakcie pisania sprawdzianu. Podobnie ma się rzecz w sytuacji sprawdzianu umiejętności wykonywanego na komputerze. W tej sytuacji należy zadbać – tak jak zwykle - o samodzielność pracy ucznia, w szczególności uniemożliwić komunikowanie z innymi uczniami za pomocą Internetu.

W przypadku sprawdzianu proponuję następujący sposób oceny tego typu pracy – mowa o liczbie procent zaokrąglonej do najbliższej liczby całkowitej.

od 0% do 39% - niedostateczny
od 40% do 50% - dopuszczający
od 51% do 70% - dostateczny
od 71% do 84% - dobry
od 85% do 98% - bardzo dobry
od 99% do 100% - celujący

Oczywiście nauczyciel może dostosować tę punktację do własnych potrzeb. Sugeruję aby waga oceny ze sprawdzianu wynosiła 5.

Krótsze formy pisemne takie jak kartkówki, czy też „komputerówki" – obejmujące materiał z 2-3 ostatnich lekcji i trwające około 15 minut powinny mieć też niższą wagę niż sprawdziany – proponuję, aby wynosiła ona 3. W przypadku kartkówek liczba punktów najlepiej, aby wynosiła 7, albo była wielokrotnością liczby 7.

Wtedy proponuję następującą punktację

0-2 p. – ocena niedostateczna
3p. - ocena dopuszczająca
4p. - ocena dostateczna
5-6 - ocena dobra
7 - ocena bardzo dobra

Ocenę celującą można przyznać, gdy np. uczeń wykona pracę w nowatorski sposób i do tego bezbłędnie, albo w wyjątkowo krótkim czasie, ale bez usterek.

Oceniając aktywność uczniów na lekcji można wziąć pod uwagę liczbę ćwiczeń, którą uczniowie powinni wykonać na lekcji i sposób ich wykonania. Proponuję ocenę aktywności wpisywać z wagą 1.

Oceny można wtedy wystawić wg następującego klucza

Ocena **celująca** – uczeń wykonał wszystkie ćwiczenia bezbłędnie i samodzielnie, a ponadto przedłużył problem wymyślając i rozwiązując własne ćwiczenia.

Ocena **bardzo dobra** – uczeń wykonał wszystkie ćwiczenia bezbłędnie i samodzielnie, ew. z niewielką pomocą nauczyciela polegającą na dawaniu wskazówek uczniowi.

Ocena **dobra** – uczeń wykonał niemal wszystkie ćwiczenia bezbłędnie i samodzielnie, albo wykonał wszystkie ćwiczenia, ale jedno było wykonane niepoprawnie lub z istotną pomocą nauczyciela.

Ocena **dostateczna** – uczeń wykonał większość ćwiczeń poleconych przez nauczyciela

Ocena **dopuszczająca** – uczeń wykonał przynajmniej jedno zadanie polecone przez nauczyciela, być może nawet w tym jednym była potrzebna wydatna pomoc nauczyciela.

Ocena **niedostateczna** – uczeń nie wykonał żadnego zadania na lekcji.

W podobny sposób można ocenić zadania domowe dodając kryterium, aby prace różnych uczniów nie były identyczne. Problem niesamodzielności można rozwiązać przydzielając różne tematy zadań domowych, albo uprzedzić uczniów przed zadaniem domowym o konsekwencjach niesamodzielnej pracy – tzn. np. o wpisaniu oceny niedostatecznej bez możliwości poprawy.

Warto nagradzać uczniów za udział w konkursach i olimpiadach tak pochwałami, punktami lub ocenami z zachowania jak również ocenami bardzo dobrymi lub celującymi z wagą 5 – równą wadze sprawdzianu. Często stosowane na informatyce są projekty długoterminowe lub krótkoterminowe realizowane przez więcej niż 1 ucznia równocześnie. W przypadku projektów długoterminowych warto oceniać uczniów systematycznie w trakcie poszczególnych etapów. Można sobie pomóc podczas oceny tego typu pracy prosząc uczniów o wypełnienie tabelki z punktami przydzielanymi pozostałym uczniom jak i sobie za poszczególne aktywności. Ważne jest aby liczba punktów możliwych do przydzielenia nie była wielokrotnością liczby uczniów pracujących wspólnie. Ten sposób wymusza ocenę, bo uczniowie wskazują w ten sposób komu wg nich przysługuje wyższa ocena. Należy też pamiętać, że projekt kończy się prezentacją wyników na forum klasy, szkoły czy też przed szerszym gronem. Ta prezentacja powinna także być oceniona, bo każde wystąpienie musi być przecież przygotowane, uczniowie przeżywają z reguły duży stres chcąc wypaść jak najlepiej i dobrze, aby ten fakt był doceniony. Oceny z projektu proponuję wystawiać z wagą 2 lub 3 oczywiście informując o tym uczniów.

Coraz większą popularnością cieszy się metoda WebQuest, dlatego chciałam zwrócić uwagę na kilka faktów. Metoda ta jest bardzo atrakcyjna, nowoczesna i lubiana przez uczniów, ale jej użycie

wymaga bardzo starannego planowania, dużego nakładu czasu i konsekwencji w stosowaniu. Najpierw nauczyciel określa temat nad którym chciałby pracować tą metodą. Następnie wyodrębnia temat – zadania dla poszczególnych grup, które powstaną. Nauczyciel wyszukuje źródła – najczęściej strony www, w których uczniowie będą szukać wiadomości na określony temat. Kolejny etap to szczegółowe zapisanie kryteriów oceny – tzn. za wykonanie jakiej czynności uczeń dostaje ile punktów, należy także podać ile punktów odpowiada jakiej ocenie. W ten sposób uczeń niejako projektuje swoją ocenę – wie, że musi wykonać konkretne czynności, aby dostać podaną liczbę punktów i ocenę.

Następnie nauczyciel tworzy stronę internetową na której umieszcza podział tematów, źródła dla poszczególnych grup i kryteria oceniania. Na lekcji dzieli klasę lub grupę na zespoły, którym przydziela konkretne tematy. Ustalany zostaje harmonogram pracy i termin ukończenia zadania. Uczniowie pracują w grupach, na lekcjach i w domu a w określonym czasie następuje podsumowanie, prezentacja wyników i ocena. Sugeruję, aby ta ocena miała wagę 3, bo wymaga dużego nakładu pracy od uczniów.

Aby na podstawie średniej ważonej wystawić ocenę semestralną lub roczną można stosować następującą skalę:

Jeśli uczeń ma średnią ważoną poniżej 1,80 – otrzymuje ocenę niedostateczną.
Jeśli uczeń ma średnią ważoną z przedziału <1,8;2,7) – otrzymuje ocenę dopuszczającą.
Jeśli uczeń ma średnią ważoną z przedziału <2,7;3,6) – otrzymuje ocenę dostateczną.
Jeśli uczeń ma średnią ważoną z przedziału <3,6;4,5) – otrzymuje ocenę dobrą.
Jeśli uczeń ma średnią ważoną z przedziału <4,5;5,4) – otrzymuje ocenę bardzo dobrą.
Jeśli uczeń ma średnią ważoną z przedziału <5,4;6,9> – otrzymuje ocenę celującą.

Ogólne kryteria ocen z informatyki

Opisując sylwetkę ucznia, który otrzymał z informatyki konkretną ocenę roczną można przyjąć następujące założenia.

Ocenę **celującą** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny bardzo dobre i celujące,
- bierze udział w różnego typu konkursach informatycznych (także w Olimpiadzie Informatycznej),
- samodzielnie rozwiązuje złożone problemy algorytmiczne, programistyczne i inne związane z informatyką, np. stworzył własną stronę www opartą na skryptach, albo przygotował bazę danych sklepu internetowego, albo stworzył i administruje małą siecią komputerową,
- stosuje nietypowe sposoby rozwiązywania zadań, potrafi przedłużyć problem,
- jest zainteresowany poszerzaniem swojej wiedzy w zakresie informatyki, np. w zakresie grafiki, gdzie jego umiejętności wykraczają znacznie poza materiał przekazywany w szkole,
- widzi powiązania informatyki z innymi dziedzinami wiedzy.

Ocenę **bardzo dobrą** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny bardzo dobre,
- zna sposób działania algorytmów i potrafi wybrać odpowiedni do danego problemu algorytmicznego,
- potrafi stworzyć od podstaw relacyjną bazę danych wykorzystując także SQL,
- zna i samodzielnie stosuje strategię rozwiązywania problemu algorytmicznego,

- potrafi zastosować program graficzny do zaawansowanej obróbki zdjęcia przekształcając je zgodnie ze swoimi potrzebami,
- zna tendencje rozwoju informatyki,
- potrafi stosować algorytmy szyfrujące i deszyfrujące.

Ocenę **dobrą** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny dobre,
- zna i rozumie sposób działania algorytmów, programów, narzędzi potrafi je odpowiednio dobrać, zastosować, przeanalizować sposób działania i uzasadnić swój wybór,
- potrafi dostosować program napisany w języku programowania (np. w C++) do nowego problemu,
- potrafi zmienić gotową bazę danych tak, aby pasowała do nowych założeń,
- potrafi samodzielnie przeprowadzić proste rozumowanie prowadzące do rozwiązania problemu.

Ocenę **dostateczną** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny dostateczne,
- zna podstawowe pojęcia informatyczne zawarte w podstawie programowej – potrafi je wyjaśnić i zastosować,
- potrafi w prostych przypadkach zmodyfikować gotowy program, bazę danych itp. aby przystosować je do nowego zadania,
- samodzielnie wykonuje proste zadania związane z informatyką,
- stara się używać poprawnej terminologii.

Ocenę **dopuszczającą** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny dopuszczające,
- zna podstawowe pojęcia informatyczne zawarte w podstawie programowej – z pomocą nauczyciela potrafi je wyjaśnić i zastosować,
- samodzielnie lub z pomocą nauczyciela wykonuje proste zadania związane z informatyką – np. pisze proste programy, ale ma problemy z poprawieniem błędów,
- potrafi określić czy zadanie jest rozwiązane poprawnie, czy rozwiązanie odpowiada zapotrzebowaniu.

Ocenę **niedostateczną** dostanie uczeń, który:
- ze sprawdzianów, kartkówek, projektów i innych aktywności otrzymuje oceny niedostateczne,
- nie opanował nawet podstawowych pojęć informatycznych zawartych w podstawie programowej,
- nie potrafi samodzielnie ani z pomocą nauczyciela wykonać prostych zadań algorytmicznych, programistycznych, i innych związanych z materiałem informatyki na poziomie rozszerzonym,
- nie rozumie najprostszych nawet pojęć związanych z informatyką,
- nie przejawia ochoty w nadrobieniu zaległości, uzupełnieniu braków, nie chce przyjąć pomocy w tym zakresie.

Warunki realizacji przedmiotu informatyka

Zajęcia z informatyki powinny odbywać się w odpowiednio wyposażonej pracowni komputerowej. Każdy uczeń w miarę możliwości powinien mieć samodzielne stanowisko komputerowe wyposażone w edytor tekstu, arkusz kalkulacyjny, program do tworzenia prezentacji multimedialnych, środowisko programistyczne za pomocą, którego uczeń będzie pisał programy

w języku programowania wyższego poziomu. Stanowisko to powinno też zapewniać dostęp do Internetu.

Biorąc pod uwagę stan i wyposażenie pracowni komputerowych w Polsce można założyć, że większość szkół, w których będzie realizowany przedmiot informatyka na poziomie rozszerzonym jest wyposażona w system operacyjny Windows oraz pakiet biurowy Ofice. W takiej sytuacji wystarczy doinstalować tylko jedną z wersji C++ - np. Dev C++ 4.9.9.2. Jeśli w pracowni brakuje któregoś elementu wspomnianego przeze mnie oprogramowania biorąc pod uwagę stan finansów szkół warto rozważyć zainstalowanie jednej z dystrybucji systemu operacyjnego Linux oraz pakietu Open Office i odpowiedniej wersji C++.

Dobrze by było, aby uczniowie pracowali korzystając z imiennych kont i mieli możliwość przechowywania swoich plików w taki sposób, aby inni uczniowie nie mieli do nich dostępu. Jeśli nie ma takiej możliwości należy zachęcić uczniów, aby przechowywali swoje pliki na edysku, albo wysyłali do siebie załączniki na swoją pocztę e-mail. Inna możliwość to zapisywanie kopii plików na pendrive, ale należy wtedy zwrócić uwagę na każdorazowe przeskanowanie tego typu pamięci programem antywirusowym.

Uwagi końcowe

Napisany przeze mnie program oczywiście nie wyczerpuje możliwości nauczyciela pozostawiając mu spory margines niezależności. Jeśli okaże się, że zostało nam trochę czasu proponuję zapoznać uczniów z teorią grafów dobrze zaprezentowaną na stronie informatyka.wroc.pl.

Zachęćmy uczniów do wspólnej pracy nad jednym dokumentem korzystając np. z GoogleDocs. Jeśli w szkole jest platforma edukacyjna zachęcam do umieszczania zadań na niej. Można np. umieścić błędny program napisany w C++ a zadaniem uczniów będzie wskazanie linijek, w których są błędy.

Podobnie niektóre klasy tworzą zamknięte grupy na portalach społecznościowych jak np. Facebook- gdzie w najlepsze dyskutują o najlepszych algorytmach czy tez o szyfrach, których jakoś nie udało sie złamać.

Warto pokazać uczniom możliwości programu GeoGebra – dostępnemu bezpłatnie na stronie www.geogebra.org – ma sporo zastosowań w matematyce, fizyce i w informatyce oczywiście też. Można wprowadzić słownictwo z języka angielskiego na lekcjach informatyki, o ile oczywiście nauczyciel czuje się pewnie w tym temacie.

Dla zainteresowanych zaawansowanymi możliwościami HTML polecam tutoriale w języku angielskim dostępne na stronie http://www.w3schools.com/css/default.asp uczniowie modyfikując kod źródłowy on-line mogą od razu zobaczyć efekty swojej pracy. Możliwości jest naprawdę dużo – wymagają one z reguły dużego nakładu pracy nauczyciela, ale korzyści dla uczniów są bardzo duże.

ORE
Ośrodek Rozwoju Edukacji

Aleje Ujazdowskie 28
00-478 Warszawa
tel. 22 345 37 00
fax 22 345 37 70

www.ore.edu.pl

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Publikacja współfinansowana przez Unię Europejską w ramach Europejskiego Funduszu Społecznego