



OŚRODEK  
ROZWOJU  
EDUKACJI

Romualda Laskowska

## Informatyka

Program nauczania  
IV etap edukacyjny – zakres rozszerzony

# Informatyka

## IV etap edukacyjny – zakres rozszerzony

### Program nauczania

*„Nasza największa słabość tkwi w rezygnowaniu. Najbardziej pewną rzeczą, aby odnieść sukces jest spróbować jeszcze raz.”*

*- Thomas A. Edison*



## Spis treści

I.	Wstęp.....	3
II.	Podstawa programowa – informatyka – zakres rozszerzony .....	3
III.	Szczegółowe cele kształcenia i wychowania oraz treści nauczania w korelacji z treściami podstawy programowej .....	8
1.	Reprezentacja informacji w komputerze .....	8
2.	Wstęp do algorytmiki i programowania .....	9
3.	Programowanie – aktywniejsze wykorzystywanie możliwości języka .....	10
4.	Systemy operacyjne .....	12
5.	Sieci komputerowe .....	13
6.	Bezpieczeństwo danych .....	14
7.	Bazy danych .....	15
8.	Grafika i multimedia .....	16
IV.	Propozycja rozkładu materiału.....	18
V.	Sposoby osiągnięcia celów kształcenia i wychowania .....	31
VI.	Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne.....	33
	Kryteria na poszczególne oceny: .....	34
VII.	Propozycje metod sprawdzania osiągnięć ucznia.....	35
VIII.	Uwagi o realizacji .....	35
	Warunki realizacji programu.....	35
	Oprogramowanie .....	36

## I. Wstęp

Program nauczania informatyki powstał na bazie doświadczeń w doradztwie metodycznym oraz długoletniej pracy nauczyciela.

Nowa podstawa programowa zachęciła mnie do napisania programu nauczania wykorzystującego moje uwagi wynikające z realizacji wdrożonego własnego programu nauczania, uwag uczniów i absolwentów, przeanalizowałam także zapisy podstawy programowej. Celem mojej pracy było stworzenie narzędzia pozwalającego nauczać informatyki w sposób jak najbardziej efektywny i jednocześnie atrakcyjny dla ucznia.

Nauczanie informatyki jest niezwykle trudne, stawia przed nauczycielem cały szereg zadań i specyficznych problemów. Nauczyciel musi być wyposażony w ciągle aktualizowaną wiedzę z informatyki, jak również wykazać się znajomością zagadnień pedagogicznych. Musi być specjalistą w wielu dziedzinach informatyki i jednocześnie nie zawsze dysponuje wystarczającym warsztatem pracy.

Mam nadzieję, że moje propozycje znajdą akceptację wielu nauczycieli i pomogą im w realizacji własnych pomysłów na interesujące lekcje informatyki.

## II. Podstawa programowa – informatyka – zakres rozszerzony

W rozporządzeniu 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 2009 r. Nr 4, poz. 17) określono m.in. podstawę programową do informatyki w zakresie rozszerzonym dla IV etapu edukacyjnego:

### INFORMATYKA

#### IV etap edukacyjny – zakres rozszerzony

##### Cele kształcenia – wymagania ogólne

- I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.
- II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
- III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
- IV. Wykorzystanie komputera oraz programów i gier edukacyjnych do poszerzania wiedzy i umiejętności z różnych dziedzin oraz do rozwijania zainteresowań.
- V. Ocena zagrożeń i ograniczeń, docenianie społecznych aspektów rozwoju i zastosowań informatyki.

## Treści nauczania – wymagania szczegółowe

1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej.

### Uczeń:

- 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;
- 2) wyjaśnia funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne;
- 3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, opisuje zasady administrowania siecią komputerową w architekturze klient-serwer, prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;
- 4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania.

2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.

### Uczeń:

- 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych;
- 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);
- 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;
- 4) znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin;
- 5) opisuje mechanizmy związane z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora ogniowa.

3. Komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.

### Uczeń:

- 1) wykorzystuje zasoby i usługi sieci komputerowych w komunikacji z innymi użytkownikami, w tym do przesyłania i udostępniania danych;
- 2) bierze udział w dyskusjach w sieci (forum internetowe, czat).

4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów.

### Uczeń:

- 1) opisuje podstawowe modele barw i ich zastosowanie;
- 2) określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;
- 3) przetwarza obrazy i filmy, np.: zmienia rozdzielczość, rozmiar, model barw, stosuje filtry;
- 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.

5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.

### Uczeń:

- 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;

- 2) stosuje podejście algorytmiczne do rozwiązywania problemu;
- 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;
- 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;
- 5) posługuje się podstawowymi technikami algorytmicznymi;
- 6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;
- 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;
- 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;
- 9) stosuje rekurencję w prostych sytuacjach problemowych;
- 10) stosuje podejście zachłanne w rozwiązywaniu problemów;
- 11) opisuje podstawowe algorytmy i stosuje:
  - a) algorytmy na liczbach całkowitych, np.:
    - reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym,
    - sprawdzanie, czy liczba jest liczbą pierwszą, doskonałą,
    - rozkładanie liczby na czynniki pierwsze,
    - iteracyjna i rekurencyjna realizacja algorytmu Euklidesa,
    - iteracyjne i rekurencyjne obliczanie wartości liczb Fibonacciego,
    - wydawanie reszty metodą zachłanną,
  - b) algorytmy wyszukiwania i porządkowania (sortowania), np.:
    - jednoczesne znajdowanie największego i najmniejszego elementu w zbiorze: algorytm naiwny i optymalny,
    - algorytmy sortowania ciągu liczb: bąbelkowy, przez wybór, przez wstawianie liniowe lub binarne, przez scalanie, szybki, kubełkowy,
  - c) algorytmy numeryczne, np.:
    - obliczanie wartości pierwiastka kwadratowego,
    - obliczanie wartości wielomianu za pomocą schematu Hornera,
      - zastosowania schematu Hornera: reprezentacja liczb w różnych systemach liczbowych, szybkie podnoszenie do potęgi,
      - wyznaczanie miejsc zerowych funkcji metodą połowienia,
      - obliczanie pola obszarów zamkniętych,
  - d) algorytmy na tekstach, np.:
    - sprawdzanie, czy dany ciąg znaków tworzy palindrom, anagram,
    - porządkowanie alfabetyczne,
    - wyszukiwanie wzorca w tekście,
    - obliczanie wartości wyrażenia podanego w postaci odwrotnej notacji polskiej,
  - e) algorytmy kompresji i szyfrowania, np.:
    - kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana,
    - szyfr Cezara,
    - szyfr przestawieniowy,
    - szyfr z kluczem jawnym (RSA),
    - wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym,
  - f) algorytmy badające własności geometryczne, np.:
    - sprawdzanie warunku trójkąta,
    - badanie położenia punktów względem prostej,

- badanie przynależności punktu do odcinka,
  - przecinanie się odcinków,
  - przynależność punktu do obszaru,
  - konstrukcje rekurencyjne: drzewo binarne, dywan Sierpińskiego, płatek Kocha;
- 12) projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych;
  - 13) stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu;
  - 14) dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne;
  - 15) stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu;
  - 16) opisuje własności algorytmów na podstawie ich analizy;
  - 17) ocenia zgodność algorytmu ze specyfikacją problemu;
  - 18) oblicza liczbę operacji wykonywanych przez algorytm;
  - 19) szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu;
  - 20) bada efektywność komputerowych rozwiązań problemów;
  - 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;
  - 22) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów;
  - 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;
  - 24) dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu;
  - 25) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;
  - 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania;
  - 27) wyjaśnia źródło błędów w obliczeniach komputerowych (błąd względny, błąd bezwzględny);
  - 28) realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.

6. Wykorzystuje komputer oraz programy i gry edukacyjne do poszerzania wiedzy i umiejętności z różnych dziedzin:

**Uczeń :**

- 1) opracowuje indywidualne i zespołowe projekty przedmiotowe i międzyprzedmiotowe z wykorzystaniem metod i narzędzi informatyki;
- 2) korzysta z zasobów edukacyjnych udostępnianych na portalach przeznaczonych do kształcenia na odległość.

7. Wykorzystuje komputer i technologie informacyjno-komunikacyjne do rozwijania swoich zainteresowań, opisuje zastosowania informatyki, ocenia zagrożenia i ograniczenia, docenia aspekty społeczne rozwoju i zastosowań informatyki:

**Uczeń**

- 1) opisuje najważniejsze elementy procesu rozwoju informatyki i technologii informacyjno-komunikacyjnych;

- 2) wyjaśnia szanse i zagrożenia dla rozwoju społecznego i gospodarczego oraz dla obywateli, związane z rozwojem informatyki i technologii informacyjno-komunikacyjnych;
- 3) stosuje normy etyczne i prawne związane z rozpowszechnianiem programów komputerowych, bezpieczeństwem i ochroną danych oraz informacji w komputerze i w sieciach komputerowych;
- 4) omawia zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci;
- 5) przygotowuje się do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego.

W tym rozporządzeniu podany został dokładny zakres treści programowych i umiejętności, które powinien zdobyć uczeń. Wiadomo więc, jakie cele należy osiągnąć. Program nauczania stara się wskazać sposoby i metody, które prowadzą do ich osiągnięcia. Został zbudowany w oparciu o specyfikę szkoły, w której uczy autorka. Może być także wykorzystany w innych warunkach nauczania. Nauczyciel ma prawo do dostosowania programu do specyfiki oraz warunków szkoły, w której uczy.



### III. Szczegółowe cele kształcenia i wychowania oraz treści nauczania w korelacji z treściami podstawy programowej

1. Reprezentacja informacji w komputerze.		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;</li> </ul>	<ul style="list-style-type: none"> <li>Działania na liczbach w różnych systemach liczbowych, wartość liczby pozycyjnej, przeliczenia na inny zapis pozycyjny</li> <li>wartość liczby stałoprzecinkowej, przeliczanie na zapis stałoprzecinkowy;</li> <li>Systemy pozycyjne o podstawie większej od 10, zapis zmiennoprzecinkowy, kody binarne;</li> <li>Naturalny system dwójkowy, dwójkowy system stałoprzecinkowy, operacje arytmetyczne w systemie dwójkowym, operacje logiczne na bitach;</li> <li>Konwersje dwójkowo-ósemkowe i dwójkowo-szesnastkowe, kodowanie liczb ze znakiem, zapis w systemie znak-moduł – ZM, zapis w systemie uzupełnień do 1 - U1, zapis w systemie uzupełnień do 2 - U2, zapis w systemie z nadmiarem;</li> <li>Kodowanie informacji za pomocą bitów, algebra Boole'a, kod Gray'a, minimalizacja funkcji logicznych;</li> <li>Model działania komputera – maszyna RAM</li> </ul>	<p>1.1 5.11a 5.19</p>

2. Wstęp do algorytmiki i programowania		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.</li> <li>analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;</li> <li>stosuje podejście algorytmiczne do rozwiązywania problemu;</li> <li>formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;</li> <li>dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;</li> <li>posługuje się podstawowymi technikami algorytmicznymi;</li> <li>ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;</li> <li>opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;</li> </ul>	<ul style="list-style-type: none"> <li>Wstęp do algorytmów, pojęcie algorytmu, przykłady algorytmów nieinformatycznych, poszczególne etapy algorytmu, sposób zapisu algorytmu (lista kroków, schemat blokowy, pseudocode, język programowania), cechy poprawnego algorytmu</li> <li>Wprowadzenie do analizy złożoności algorytmu, pojęcie złożoności algorytmu, podział algorytmów z uwagi na złożoność, optymalizacja i jej sens na określonych etapach</li> <li>Wstęp do programowania, omówienie najpopularniejszych języków programowania i ich zastosowania</li> <li>Wprowadzenie do języków C/C++, krótka charakterystyka języka, omówienie sposobów kompilacji, pierwszy program, operacje wejścia/wyjścia (cout, cin), pojęcie zmiennej, podstawowe typy zmiennej (int, double, char), operacje przypisania</li> <li>Podstawowe działania matematyczne (+, -, *, /) instrukcja warunkowa C/C++, proste warunki i warunki złożone (&amp;&amp;,   ), konstrukcja switch</li> <li>Przykładowe programy, takie jak: badanie znaków liczby, implementacja programu obliczającego rozwiązanie równania liniowego, implementacja programu obliczającego rozwiązanie równania kwadratowego</li> <li>Pętle for i while, rodzaje pętli i sposób ich zapisu,</li> <li>algorytmy badające właściwości liczb całkowitych, badanie podzielności, algorytm Euklidesa (NWW, NWD) – omówienie i implementacja programu.</li> <li>Liczby pierwsze, pojęcie liczby pierwszej, sprawdzanie czy liczba jest pierwsza</li> <li>Zapoznanie z pojęciem metody Monte Carlo, implementacja programu: znajdowanie przybliżonej wartości liczby Pi, ruchy Browna – błędzenie przypadkowe.</li> <li>Przybliżanie wartości pierwiastka kwadratowego, przedstawienie metody i implementacja programu (metoda Newtona–Raphsona),</li> <li>Wyznaczanie przybliżonej wartości miejsca zerowego funkcji, metoda wyznaczania przez połowienie przedziałów (bisekcję)</li> </ul>	<p>5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11a 5.11c 5.20 5.21 5.12 5.13 5.14 5.15 5.22 5.23 5.24</p>

<b>3. Programowanie – aktywniejsze wykorzystywanie możliwości języka</b>		
<b>Cele kształcenia</b>	<b>Treści nauczania</b>	<b>Zapisy podstawy programowej</b>
<ul style="list-style-type: none"> <li>• posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;</li> <li>• stosuje rekurencję w prostych sytuacjach problemowych;</li> <li>• stosuje podejście zachłanne w rozwiązywaniu problemów;</li> <li>• opisuje podstawowe algorytmy i stosuje:</li> <li>• projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych;</li> <li>• stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu;</li> <li>• dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne;</li> <li>• stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu;</li> <li>• opisuje własności algorytmów na podstawie ich analizy;</li> <li>• ocenia zgodność algorytmu ze specyfikacją problemu;</li> <li>• oblicza liczbę operacji wykonywanych przez algorytm;</li> <li>• szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu;</li> <li>• bada efektywność komputerowych rozwiązań problemów;</li> <li>• przeprowadza</li> </ul>	<ul style="list-style-type: none"> <li>• Funkcje w C/C++, sposób przekazywania parametrów do funkcji: parametry formalne i aktualne, zmienne globalne i lokalne, prototypy funkcji</li> <li>• Tablica jednowymiarowa i dwuwymiarowa, deklaracja, inicjalizacja, przekazywanie tablic do funkcji, poszukiwanie maksimum/minimum/średniej, przeszukiwanie tablicy w celu znalezienia wyróżnionego elementu – wersja z wartownikiem i bez wartownika, porównanie obu metod pod kątem optymalności algorytmu</li> <li>• Implementacja obliczeń na systemach liczbowych, Sito Eratostenesa</li> <li>• Sortowanie tablicy jednowymiarowej metodą bąbelkową, omówienie metody i jej złożoności obliczeniowej</li> <li>• Sortowanie tablicy metodami przez wybór i selekcję</li> <li>• Przeszukiwanie binarne</li> <li>• Omówienie metody „dziel i zwyciężaj”</li> <li>• Operacje na macierzach, sumy na przekątnej, wyznacznik macierzy, rozwiązywanie układów równań</li> <li>• Implementacja programów z zastosowaniem tablic jedno- i dwuwymiarowych i zdefiniowanych przez programistę funkcji, w tym funkcji sortujących</li> <li>• Rekurencja, pojęcie rekurencji, przykłady rekurencji, takie jak: silnia, potęga, kolejne wyrazy ciągu Fibonacciego</li> <li>• Schemat Hornera, implementacja programu, porównanie metody z obliczaniem wartości wielomianu przy wykorzystaniu postaci definicyjnej wielomianu</li> <li>• Algorytmy rekurencyjne, wieża Hanoi, problem ośmiu hetmanów i skoczka szachowego</li> <li>• Konstrukcje rekurencyjne: drzewo binarne, dywan Sierpińskiego, płatek Kocha;</li> <li>• Sortowanie przez scalanie</li> <li>• Sortowanie szybkie (ang. quicksort)</li> <li>• Programowanie – struktury i wskaźniki, pojęcie wskaźnika, tekst jako wskaźnik, operacje na wskaźnikach,</li> <li>• Pojęcie struktury, prosta baza danych</li> <li>• Lista jednokierunkowa i dwukierunkowa, drzewo, tablice dynamiczne</li> <li>• Algorytmy na tekstach, sprawdzanie, czy dany</li> </ul>	<p>5.11b 5.11d 5.11e 5.11f 5.12 5.13 5.14 5.15 5.22 5.23 5.24 5.25 5.26 5.27 5.28 6.1 7.5</p>

<p>komputerową realizację algorytmu i rozwiązania problemu;</p> <ul style="list-style-type: none"> <li>• sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów;</li> <li>• stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;</li> <li>• dobiera najlepszy algorytm, odpowiednie struktury danych i oprogramowanie do rozwiązania postawionego problemu;</li> <li>• dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;</li> <li>• ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania;</li> <li>• wyjaśnia źródło błędów w obliczeniach komputerowych (błąd względny, błąd bezwzględny);</li> <li>• realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.</li> </ul>	<p>ciąg znaków tworzy palindrom, anagram, porządkowanie alfabetyczne, wyszukiwanie wzorca w tekście, obliczanie wartości wyrażenia podanego w postaci odwrotnej notacji polskiej,</p> <ul style="list-style-type: none"> <li>• Algorytmy kompresji i szyfrowania, kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana, szyfr Cezara, szyfr przestawieniowy, szyfr z kluczem jawnym (RSA), wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym,</li> <li>• Algorytmy badające własności geometryczne, sprawdzanie warunku trójkąta, badanie położenia punktów względem prostej, badanie przynależności punktu do odcinka, przecinanie się odcinków, przynależność punktu do obszaru,</li> <li>• Przygotowanie projektu programistycznego</li> </ul>	
--	---	--

4. Systemy operacyjne		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>wyjaśnia funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne;</li> <li>zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania</li> </ul>	<ul style="list-style-type: none"> <li>Podstawowe elementy systemu operacyjnego i ich funkcje;</li> <li>Budowa systemu plików – podstawowe polecenia operujące na plikach i katalogach w systemie NTFS</li> <li>Budowa systemu plików – podstawowe polecenia operujące na plikach i katalogach w systemie ext3</li> <li>Interpretowanie i ustalanie praw dostępu. Przeglądanie i usuwanie działających procesów.</li> <li>Montowanie urządzeń. Wymiana plików pomiędzy systemami operacyjnymi. Protokół SMB.</li> <li>Środowiska graficzne kontra tekstowe. Zarządzanie systemem z wykorzystaniem środowiska graficznego</li> <li>Pliki konfiguracyjne, ich lokalizacja, budowa</li> </ul>	<p>1.2 1.4 3.1</p>

5. Sieci komputerowe		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>• przedstawia warstwowy model sieci komputerowych,</li> <li>• określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci,</li> <li>• opisuje zasady administrowania siecią komputerową w architekturze klient-serwer,</li> <li>• prawidłowo posługuje się terminologią sieciową, korzysta z usług w sieci komputerowej, lokalnej i globalnej, związanych z dostępem do informacji, wymianą informacji i komunikacją;</li> </ul>	<ul style="list-style-type: none"> <li>• System operacyjny i programy narzędziowe – mechanizmy ochrony danych.</li> <li>• Sieci komputerowe – rodzaje sieci ze względu na topologię i zasięg, zalety, wady, urządzenia działające w sieci.</li> <li>• Pojęcia związane z funkcjonowaniem sieci komputerowych (protokół pakiet, DNS, ruting itp.), adresacja w sieci Internet, adresacja w sieci.</li> <li>• Warstwowy model budowy sieci.</li> <li>• Usługi sieciowe, udostępnianie zasobów komputera do sieci.</li> <li>• Bezpieczeństwo w sieci – programy chroniące zasoby komputera.</li> <li>• Śledzenie transmisji pakietów w Sieci, ich analiza</li> <li>• Zaciskanie kabli, gniazdek, podłączanie przełączników</li> <li>• Konfiguracja routera WiFi do udostępniania sieci domowej</li> </ul>	<p>1.2 1.3 3.1 3.2 6.2</p>

<b>6. Bezpieczeństwo danych</b>		
<b>Cele kształcenia</b>	<b>Treści nauczania</b>	<b>Zapisy podstawy programowej</b>
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>opisuje najważniejsze elementy procesu rozwoju informatyki i technologii informacyjno-komunikacyjnych;</li> <li>wyjaśnia szanse i zagrożenia dla rozwoju społecznego i gospodarczego oraz dla obywateli, związane z rozwojem informatyki i technologii informacyjno-komunikacyjnych;</li> <li>stosuje normy etyczne i prawne związane z rozpowszechnianiem programów komputerowych, bezpieczeństwem i ochroną danych oraz informacji w komputerze i w sieciach komputerowych;</li> <li>omawia zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci;</li> <li>przygotowuje się do świadomego wyboru kierunku i zakresu dalszego kształcenia informatycznego.</li> </ul>	<ul style="list-style-type: none"> <li>Zagadnienia bezpieczeństwa usług serwerów internetowych. Włamania do systemów, hacking. Profilaktyka i procedury bezpieczeństwa. Elementy polityki zabezpieczeń.</li> <li>Prezentacja zaawansowanej konfiguracji serwerów internetowych.</li> <li>Zabezpieczanie informacji (kopie bezpieczeństwa, ochrona antywirusowa);</li> <li>Bezpieczeństwo w sieciach komputerowych</li> <li>Stosowanie zasad zachowania w sieciach, normy prawne i etyczne</li> </ul>	<p>2.5 6.2 7.1 7.2 7.3 7.4</p>



7. Bazy danych		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>projektuje relacyjną bazę danych z zapewnieniem integralności danych;</li> <li>stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);</li> <li>tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;</li> <li>znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin;</li> <li>realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.</li> </ul>	<ul style="list-style-type: none"> <li>Baza danych na przykładzie MS-Access (lub alternatywnie dBase)</li> <li>Relacje w bazie danych</li> <li>Kwerendy wyszukujące, tworzenie kwerend</li> <li>Przetwarzanie wyników kwerend - tworzenie prostych raportów</li> <li>Tworzenie i projektowanie formularzy</li> <li>Język SQL i jego zastosowania</li> <li>Tworzenie przykładowych baz danych wykorzystujących kwerendy oparte na SQL</li> <li>Podstawy języka SQL, zasady składni i podstawowe instrukcje.</li> <li>Obsługa SQLowej bazy danych na podstawie MySQL i phpMyAdmin</li> <li>Import i eksport danych</li> <li>Mechanizmy ochrony bazy danych</li> <li>Przygotowanie projektu prostej bazy danych</li> </ul>	<p>2.1</p> <p>2.2</p> <p>2.3</p> <p>2.4</p> <p>5.1</p> <p>6.1</p> <p>6.2</p>



8. Grafika i multimedia		
Cele kształcenia	Treści nauczania	Zapisy podstawy programowej
<p><b>Uczeń:</b></p> <ul style="list-style-type: none"> <li>opisuje podstawowe modele barw i ich zastosowanie;</li> <li>określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;</li> <li>przetwarza obrazy i filmy, np.: zmienia rozdzielczość, rozmiar, model barw, stosuje filtry;</li> <li>realizuje indywidualnie lub zespołowo projekt programistyczny z wydzieleniem jego modułów, w ramach pracy zespołowej, dokumentuje pracę zespołu.</li> </ul>	<ul style="list-style-type: none"> <li>Obróbka grafiki rastrowej</li> <li>Pozyskiwanie grafiki z Internetu, skanera, aparatu cyfrowego, dopasowywanie wielkości obrazka, przycinanie, kompozycja</li> <li>Edytor grafiki wektorowej</li> <li>Objaśnienie pojęć grafika rastrowa i grafika wektorowa, rysowanie linii krzywych i odcinków, rysowanie łamanych, rysowanie figur zamkniętych</li> <li>Skalowanie figur, lustrzane odbicie w pionie i poziomie, kopiowanie figur, powtarzanie operacji kopiowania, obracanie i ścinanie figur</li> <li>Zastosowania figur wektorowych w prostych projektach graficznych</li> <li>Krzywe Beziera w grafice wektorowej</li> <li>Zastosowania edycji kształtu krzywej w prostych projektach graficznych</li> <li>Morphing</li> <li>Zastosowania efektów w prostych projektach graficznych</li> <li>Eksport rysunku wektorowego do formatu rastrowego</li> <li>Tworzenie prostych projektów graficznych</li> <li>Edytor grafiki wektorowej</li> <li>Rodzaje grafik - rastrowa (GIF, JPG) i wektorowa, różnice, zalety i wady</li> <li>Tworzenie prezentacji multimedialnej w Prezi</li> <li>Zasady tworzenia prezentacji, animacja slajdów oraz ich elementów, tworzenie ścieżki, osadzanie grafiki, muzyki i filmów w prezentacji</li> <li>Projekt prezentacji na zadany temat</li> <li>Elementy języka HTML 5; arkusze stylów CSS 3.0</li> <li>Idea scentralizowanego formatowania, tworzenie prostego arkusza stylów</li> <li>Przygotowanie prezentacji w oparciu o impress.js</li> <li>Projekt prezentacji z formatowaniem CSS na zadany temat</li> <li>Animacje komputerowe</li> <li>Budowa ekranu roboczego edytora Vectorian Giotto</li> <li>Tworzenie animacji poklatkowych</li> <li>Tworzenie animacji tween</li> </ul>	<p>4.1 4.2 4.3 4.4 6.1 2.3 2.4 3.1 5.1 5.28 6.1 6.2</p>

	<ul style="list-style-type: none"><li>• Animacje wielowarstwowe</li><li>• Język programowania Action Script</li><li>• Wstawianie kodu ActionScript do animacji Flash</li><li>• Podobieństwo Action Script do C++</li><li>• Tworzenie animacji wykorzystujących programowanie</li><li>• Język PHP</li><li>• Oprogramowanie po stronie serwera - zalety, bezpieczeństwo kodu</li><li>• Zasada działania preprocesora - przetwarzanie kodu PHP, osadzanie skryptu PHP w kodzie HTML strony - konfiguracja serwera HTTP do pracy ze stronami zawierającymi kod PHP</li><li>• Podobieństwo języka PHP do C++</li><li>• Proste projekty PHP - przetwarzanie formularzy, przekazywanie parametrów w URL strony, dostęp do plików, prosty licznik odwiedzin strony WWW, itp.</li><li>• Projekt strony zawierającej skrypty PHP</li></ul>	
--	--	--

#### IV. Propozycja rozkładu materiału

Treści nauczania i tematy lekcji	Liczba godzin
<b>Reprezentacja informacji w komputerze</b>	
Temat: Organizacja zajęć w pracowni komputerowej <ul style="list-style-type: none"> <li>• BHP w pracowni informatycznej.</li> <li>• Regulamin pracowni.</li> <li>• Wymagania programowe.</li> <li>• Organizacja zajęć.</li> <li>• Rozkład materiału.</li> <li>• Przydział stanowisk</li> </ul>	1
Temat: Działania na liczbach w różnych systemach liczbowych <ul style="list-style-type: none"> <li>• Działania na liczbach w różnych systemach liczbowych,</li> <li>• Wartość liczby pozycyjnej,</li> <li>• Przeliczenia na inny zapis pozycyjny</li> <li>• Proste operacje arytmetyczne w różnych systemach liczbowych</li> <li>• Wartość liczby stałoprzecinkowej</li> <li>• Przeliczanie na zapis stałoprzecinkowy</li> </ul>	1
Temat: Zapis zmiennoprzecinkowy, kody binarne <ul style="list-style-type: none"> <li>• Systemy pozycyjne o podstawie większej od 10,</li> <li>• Zapis zmiennoprzecinkowy,</li> <li>• Kody binarne;</li> </ul>	1
Temat: Naturalny system dwójkowy, <ul style="list-style-type: none"> <li>• dwójkowy system stałoprzecinkowy,</li> <li>• operacje arytmetyczne w systemie dwójkowym,</li> <li>• operacje logiczne na bitach;</li> </ul>	2
Temat: Konwersje dwójkowo-ósemkowe i dwójkowo-szesnastkowe <ul style="list-style-type: none"> <li>• kodowanie liczb ze znakiem,</li> <li>• zapis w systemie znak-moduł – ZM,</li> <li>• zapis w systemie uzupełnień do 1 - U1,</li> <li>• zapis w systemie uzupełnień do 2 - U2,</li> <li>• zapis w systemie z nadmiarem;</li> </ul>	2

<p>Temat: Kodowanie informacji za pomocą bitów</p> <ul style="list-style-type: none"> <li>• algebra Boole'a,</li> <li>• kod Gray'a,</li> <li>• minimalizacja funkcji logicznych;</li> </ul>	1
<p>Temat: Model działania komputera – maszyna RAM</p> <ul style="list-style-type: none"> <li>• Zasada działania komputera na przykładzie maszyny Ram</li> <li>• Proste operacje z użyciem maszyny RAM</li> <li>• Implementacja prostych algorytmów w maszynie RAM</li> </ul>	2
<b>Wstęp do algorytmiki i programowania</b>	
<p>Temat: Wstęp do algorytmiki</p> <ul style="list-style-type: none"> <li>• pojęcie algorytmu,</li> <li>• przykłady algorytmów nieinformatycznych,</li> <li>• poszczególne etapy algorytmu,</li> <li>• sposób zapisu algorytmu (lista kroków, schemat blokowy, pseudojęzyk, język programowania),</li> <li>• cechy poprawnego algorytmu</li> </ul>	2
<p>Temat: Ćwiczenia w zapisie prostych algorytmów</p>	3
<p>Temat: Wprowadzenie do analizy złożoności algorytmu,</p> <ul style="list-style-type: none"> <li>• pojęcie złożoności algorytmu,</li> <li>• podział algorytmów z uwagi na złożoność,</li> <li>• optymalizacja i jej sens na określonych etapach</li> <li>• wykorzystanie arkusza kalkulacyjnego do konstruowania prostych algorytmów</li> </ul>	2
<p>Temat: Wstęp do programowania,</p> <ul style="list-style-type: none"> <li>• omówienie najpopularniejszych języków programowania i ich zastosowania</li> </ul>	1
<p>Temat: Wprowadzenie do języków C/C++,</p> <ul style="list-style-type: none"> <li>• krótka charakterystyka języka,</li> <li>• omówienie sposobów kompilacji,</li> <li>• instalacja i konfiguracja środowiska programistycznego</li> <li>• omówienie środowiska programowania</li> <li>• pierwszy program,</li> <li>• operacje wejścia/wyjścia (cout, cin),</li> </ul>	2

<p>Temat: Pojęcie zmiennej,</p> <ul style="list-style-type: none"> <li>• podstawowe typy zmiennej (int, double, char),</li> <li>• operacje przypisania</li> <li>• Podstawowe działania matematyczne (+, -, *, /)</li> </ul>	2
<p>Temat: Podejmowanie decyzji w programie</p> <ul style="list-style-type: none"> <li>• instrukcja warunkowa C/C++,</li> <li>• proste warunki i warunki złożone (&amp;&amp;,   ),</li> </ul>	3
<p>Temat: Instrukcja wyboru</p> <ul style="list-style-type: none"> <li>• konstrukcja switch</li> <li>• zastosowanie konstrukcji w prostych programach</li> </ul>	2
<p>Temat: Ćwiczenia w programowaniu</p> <ul style="list-style-type: none"> <li>• badanie znaków liczby,</li> <li>• implementacja programu obliczającego rozwiązanie równania liniowego,</li> <li>• implementacja programu obliczającego rozwiązanie równania kwadratowego</li> <li>• badanie poprawności rozwiązania</li> <li>• zastosowanie narzędzi do badania poprawności rozwiązania programistycznego – szkolna testerka zadań</li> </ul>	4
<p>Temat: Iteracja</p> <ul style="list-style-type: none"> <li>• Pętle while i do...while</li> <li>• rodzaje pętli i sposób ich zapisu,</li> <li>• Pętla for – składnia i zastosowanie w programie</li> </ul>	3
<p>Temat: Ćwiczenia w stosowaniu iteracji w programach</p>	3
<p>Temat: Implementacja algorytmów badających właściwości liczb całkowitych</p> <ul style="list-style-type: none"> <li>• badanie podzielności,</li> <li>• algorytm Euklidesa (NWW, NWD) – omówienie (prezentacja: <a href="http://www.scholaris.pl/frontend,4,107425.html">http://www.scholaris.pl/frontend,4,107425.html</a>) i implementacja programu.</li> </ul>	2
<p>Temat: Co wiem o liczbach pierwszych</p> <ul style="list-style-type: none"> <li>• pojęcie liczby pierwszej,</li> <li>• sprawdzanie czy liczba jest pierwsza</li> </ul>	2

<p>Temat: Metody numeryczne</p> <ul style="list-style-type: none"> <li>• Zapoznanie z pojęciem metody Monte Carlo,</li> <li>• implementacja programu: znajdowanie przybliżonej wartości liczby Pi,</li> <li>• ruchy Browna – błędzenie przypadkowe.</li> </ul>	3
<p>Temat: Przybliżanie wartości pierwiastka kwadratowego</p> <ul style="list-style-type: none"> <li>• przedstawienie metody</li> <li>• implementacja programu (metoda Newtona–Raphsona),</li> </ul>	2
<p>Temat: Wyznaczanie przybliżonej wartości miejsca zerowego funkcji</p> <ul style="list-style-type: none"> <li>• metoda wyznaczania przez połowienie przedziałów (bisekcję)</li> <li>• implementacja algorytmu</li> </ul>	2
<p>Temat: Powtórzenie i utrwalenie wiadomości</p> <ul style="list-style-type: none"> <li>• Ćwiczenia w programowaniu</li> <li>• Sprawdzian wiadomości i umiejętności</li> </ul>	2
<b>Programowanie – aktywniejsze wykorzystywanie możliwości języka</b>	
<p>Temat: Funkcje w C/C++</p> <ul style="list-style-type: none"> <li>• sposób przekazywania parametrów do funkcji: parametry formalne i aktualne,</li> <li>• zmienne globalne i lokalne,</li> <li>• prototypy funkcji</li> </ul>	2
<p>Temat: Tablica jednowymiarowa i dwuwymiarowa</p> <ul style="list-style-type: none"> <li>• deklaracja,</li> <li>• inicjalizacja</li> </ul>	2
<p>Temat: Proste operacje na tablicach</p> <ul style="list-style-type: none"> <li>• poszukiwanie maksimum/minimum/średniej</li> <li>• generator liczb całkowitych</li> <li>• wypełnianie tablicy liczbami losowymi</li> </ul>	2
<p>Temat: Tablice w funkcji</p> <ul style="list-style-type: none"> <li>• przekazywanie tablic do funkcji</li> <li>• suma elementów tablicy</li> </ul>	2

<p>Temat: Przeszukiwanie tablic</p> <ul style="list-style-type: none"> <li>• przeszukiwanie tablicy w celu znalezienia wyróżnionego elementu – wersja z wartownikiem i bez wartownika,</li> <li>• porównanie obu metod pod kątem optymalności algorytmu</li> <li>• Zliczanie wg kryterium</li> <li>• Wyszukiwanie max lub min</li> <li>• Jednoczesne wyszukiwanie max i min</li> </ul>	4
<p>Temat: Implementacja obliczeń na systemach liczbowych</p> <ul style="list-style-type: none"> <li>• Sito Eratostenesa</li> <li>• Liczby względnie pierwsze</li> <li>• Odwrotność modulo - rozszerzony algorytm Euklidesa</li> <li>• Liczby pierwsze - generacja przez sprawdzanie podzielności</li> <li>• Liczby pierwsze - generacja sitem Eratostenesa</li> <li>• Liczby pierwsze - generacja sitem liniowym</li> <li>• Liczby pierwsze - generacja sitem Atkina-Bernsteina</li> <li>• Czynniki pierwsze - metoda próbnych dzielen</li> <li>• Czynniki pierwsze - metoda Fermata</li> <li>• Pierwszość liczby naturalnej - algorytmy naiwne</li> <li>• Pierwszość liczby naturalnej - Chiński Test Pierwszości</li> <li>• Pierwszość liczby naturalnej - Małe Twierdzenie Fermata</li> <li>• Pierwszość liczby naturalnej - test Millera-Rabina</li> </ul>	4
<p>Temat: Sortowanie tablicy jednowymiarowej metodą bąbelkową</p> <ul style="list-style-type: none"> <li>• omówienie metody i jej złożoności obliczeniowej</li> </ul>	1
<p>Temat: Sortowanie tablicy metodami przez wybór i selekcję</p> <ul style="list-style-type: none"> <li>• omówienie metod i ich złożoności obliczeniowej</li> </ul>	1
<p>Temat: Przeszukiwanie binarne</p> <ul style="list-style-type: none"> <li>• Wyszukiwanie najczęstszej wartości w zbiorze – dominanta</li> <li>• Wyszukiwanie binarne</li> <li>• Wyszukiwanie interpolacyjne</li> </ul>	2
<p>Temat: Omówienie metody „dziel i zwyciężaj”</p> <ul style="list-style-type: none"> <li>• Wyszukiwanie lidera</li> <li>• Wyszukiwanie k-tego największego elementu</li> <li>• Wyszukiwanie szybkie k-tego największego elementu</li> </ul>	2

<ul style="list-style-type: none"> <li>Wyszukiwanie mediany zbioru</li> </ul>	
<p>Temat: Implementacja programów z zastosowaniem tablic jedno- i dwuwymiarowych i zdefiniowanych przez programistę funkcji, w tym funkcji sortujących</p> <ul style="list-style-type: none"> <li>Ćwiczenia w programowaniu</li> <li>Sprawdzian wiadomości i umiejętności</li> </ul>	2
<p>Temat: Rekurencja</p> <ul style="list-style-type: none"> <li>pojęcie rekurencji,</li> <li>przykłady rekurencji, takie jak: silnia, potęga, kolejne wyrazy ciągu Fibonacciego, algorytm Euklidesa: <a href="http://www.scholaris.pl/frontend,4,107865.html">http://www.scholaris.pl/frontend,4,107865.html</a></li> <li>implementacja funkcji rekurencyjnych</li> </ul>	2
<p>Temat: Schemat Hornera,</p> <ul style="list-style-type: none"> <li>implementacja programu,</li> <li>porównanie metody z obliczaniem wartości wielomianu przy wykorzystaniu postaci definicyjnej wielomianu</li> </ul>	1
<p>Temat: Algorytmy rekurencyjne,</p> <ul style="list-style-type: none"> <li>wieża Hanoi,</li> <li>problem ośmiu hetmanów i skoczka szachowego</li> </ul>	2
<p>Temat: Zastosowanie trybu graficznego w konstruowaniu figur z użyciem rekurencji</p> <ul style="list-style-type: none"> <li>drzewo binarne,</li> <li>dywan Sierpińskiego,</li> <li>platek Kocha;</li> <li>fraktale</li> </ul>	2
<p>Temat: Sortowanie przez scalanie</p>	1
<p>Temat: Sortowanie szybkie (ang. quicksort)</p>	1
<p>Temat: Typ wskaźnikowy</p> <ul style="list-style-type: none"> <li>pojęcie wskaźnika,</li> <li>tekst jako wskaźnik,</li> </ul>	1



<ul style="list-style-type: none"> <li>operacje na wskaźnikach,</li> </ul>	
<p>Temat: Struktury danych</p> <ul style="list-style-type: none"> <li>pojęcie struktury,</li> <li>prosta baza danych</li> <li>operacje w bazie danych</li> </ul>	1
<p>Temat: Lista jednokierunkowa i dwukierunkowa,</p> <ul style="list-style-type: none"> <li>Podstawowe pojęcia dotyczące list</li> <li>Reprezentacja list w pamięci komputera</li> <li>Operacje na listach jednokierunkowych</li> <li>Operacje na listach dwukierunkowych</li> <li>Operacje na listach cyklicznych jednokierunkowych</li> <li>Liniowe przeszukiwanie listy</li> <li>Przeszukiwanie liniowe z wartownikiem listy dwukierunkowej</li> <li>Wyszukiwanie największego/najmniejszego elementu listy</li> <li>Zliczanie elementów listy</li> <li>Usuwanie z listy duplikatów</li> <li>Odwracanie listy jednokierunkowej</li> <li>Podział listy jednokierunkowej na dwie listy</li> </ul>	2
<p>Temat: Wstęp do teorii grafów</p> <ul style="list-style-type: none"> <li>Pojęcia podstawowe z teorii grafów</li> <li>Znajdowanie spójnych składowych grafu - implementacja w C++</li> <li>podsumowanie algorytmów przeszukiwania grafów: preorder, inorder, postorder, BFS, DFS.</li> <li>Numerowanie odwiedzanych węzłów, tworzenie drzew rozpinających.</li> <li>Pokaz e-lekcji: <a href="http://app.scholaris.pl/uruchom/algorytmy_grafowe.1">http://app.scholaris.pl/uruchom/algorytmy_grafowe.1</a></li> </ul>	3
<p>Temat: Algorytmy na tekstach,</p> <ul style="list-style-type: none"> <li>Podstawowe pojęcia dotyczące przetwarzania tekstów</li> <li>Podstawowe operacje na łańcuchach znakowych</li> <li>Naiwne wyszukiwanie wzorca w tekście</li> <li>Wyszukiwanie maksymalnego prefikso-sufiksu</li> <li>Szybkie wyszukiwanie wzorca algorytmem Morrisa-Pratta</li> <li>Szybkie wyszukiwanie wzorca algorytmem Knutha-Morrisa-Pratta</li> <li>Szybkie wyszukiwanie wzorca uproszczonym algorytmem Boyera-Moore'a</li> <li>Szybkie wyszukiwanie wzorca pełnym algorytmem Boyera-Moore'a</li> <li>Wyszukiwanie wzorca algorytmem Karpa-Rabina</li> <li>Zliczanie słów w łańcuchu</li> <li>Dzielenie łańcucha na słowa</li> <li>Wyszukiwanie najdłuższego słowa w łańcuchu</li> </ul>	4

<ul style="list-style-type: none"> <li>• Wyszukiwanie najdłuższego wspólnego podłańcucha</li> <li>• Wyszukiwanie najdłuższego wspólnego podciągu</li> <li>• Wyszukiwanie najkrótszego wspólnego nadłańcucha</li> <li>• Wyszukiwanie słów podwójnych</li> <li>• Wyszukiwanie palindromów</li> </ul>	
<p>Temat: Algorytmy kompresji i szyfrowania,</p> <ul style="list-style-type: none"> <li>• Szyfr Cezara</li> <li>• Szyfrowanie z pseudolosowym odstępem</li> <li>• Szyfry przedstawieniowe</li> <li>• Szyfr Enigmy</li> <li>• Szyfrowanie RSA</li> <li>• kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana,</li> <li>• wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym,</li> </ul>	3
<p>Temat: Algorytmy badające własności geometryczne,</p> <ul style="list-style-type: none"> <li>• sprawdzanie warunku trójkąta,</li> <li>• badanie położenia punktów względem prostej,</li> <li>• badanie przynależności punktu do odcinka,</li> <li>• przecinanie się odcinków,</li> <li>• przynależność punktu do obszaru,</li> </ul>	3
<p>Temat: Przygotowanie projektu programistycznego</p>	5

<b>Systemy operacyjne</b>	
<p>Temat: Podstawowe elementy systemu operacyjnego i ich funkcje</p> <ul style="list-style-type: none"> <li>System operacyjny i programy narzędziowe – mechanizmy ochrony danych.</li> </ul>	1
<p>Temat: Budowa systemu plików</p> <ul style="list-style-type: none"> <li>podstawowe polecenia operujące na plikach i katalogach w systemie NTFS</li> <li>podstawowe polecenia operujące na plikach i katalogach w systemie ext3</li> </ul>	2
<p>Temat: Interpretowanie i ustalanie praw dostępu</p> <ul style="list-style-type: none"> <li>Przeglądanie i usuwanie działających procesów</li> </ul>	1
<p>Temat: Wymiana plików pomiędzy systemami operacyjnymi</p> <ul style="list-style-type: none"> <li>Montowanie urządzeń.</li> <li>Protokół SMB.</li> </ul>	2
<p>Temat: Środowiska graficzne kontra tekstowe</p> <ul style="list-style-type: none"> <li>Zarządzanie systemem z wykorzystaniem środowiska graficznego</li> <li>Pliki konfiguracyjne, ich lokalizacja, budowa</li> </ul>	4
<b>Sieci komputerowe</b>	
<p>Temat: Podział sieci komputerowych</p> <ul style="list-style-type: none"> <li>rodzaje sieci ze względu na topologię i zasięg, zalety, wady, urządzenia działające w sieci.</li> </ul>	1
<p>Temat: Warstwowy model budowy sieci.</p> <ul style="list-style-type: none"> <li>Model OSI</li> <li>Model TCP</li> <li>Porównanie modeli</li> </ul>	2
<p>Temat: Pojęcia związane z funkcjonowaniem sieci komputerowych</p> <ul style="list-style-type: none"> <li>Protokół, pakiet, ramka, segment, routing, usługa</li> </ul>	1

<p>Temat: Adresacja w sieciach</p> <ul style="list-style-type: none"> <li>• Protokół IP, adres urządzenia,</li> <li>• Klasy adresów i adresacja bezklasowa</li> <li>• Maskowanie adresów, segmentacja sieci</li> <li>• Ćwiczenia w adresowaniu urządzeń w sieci</li> </ul>	2
<p>Temat: Usługi sieciowe, udostępnianie zasobów komputera do sieci</p> <ul style="list-style-type: none"> <li>• Śledzenie transmisji pakietów w sieci, ich analiza</li> <li>• Zaciskanie kabli, gniazdek, podłączanie przełączników</li> <li>• Konfiguracja routera WiFi do udostępniania sieci domowej</li> </ul>	4
<b>Bezpieczeństwo danych</b>	
<p>Temat: Zagadnienia bezpieczeństwa usług serwerów internetowych</p> <ul style="list-style-type: none"> <li>• Włamania do systemów, hacking.</li> <li>• Profilaktyka i procedury bezpieczeństwa.</li> <li>• Elementy polityki zabezpieczeń</li> </ul>	1
<p>Temat: Zabezpieczanie informacji</p> <ul style="list-style-type: none"> <li>• kopie bezpieczeństwa,</li> <li>• ochrona antywirusowa</li> <li>• rodzaje wirusów</li> </ul>	1
<p>Temat: Bezpieczeństwo w sieciach komputerowych</p> <ul style="list-style-type: none"> <li>• rodzaje ataków w sieciach</li> <li>• ochrona zasobów</li> </ul>	1
<p>Temat: Stosowanie zasad zachowania w sieciach, normy prawne i etyczne</p>	1
<b>Bazy danych</b>	
<p>Temat: Baza danych na przykładzie MS-Access (lub alternatywnie dBase)</p> <ul style="list-style-type: none"> <li>• Rodzaje baz danych,</li> <li>• Relacje w bazie danych</li> </ul>	1
<p>Temat: Tworzenie tabel w bazie</p> <ul style="list-style-type: none"> <li>• Typy danych w tabelach</li> <li>• Maski wprowadzania</li> </ul>	1

<p>Temat: Tworzenie kwerend</p> <ul style="list-style-type: none"> <li>• Kwerendy wyszukiwujące,</li> <li>• Przetwarzanie wyników kwerend - tworzenie prostych raportów</li> </ul>	1
<p>Temat: Tworzenie i projektowanie formularzy</p>	1
<p>Temat: Język SQL i jego zastosowania</p> <ul style="list-style-type: none"> <li>• Tworzenie przykładowych baz danych wykorzystujących kwerendy oparte na SQL</li> <li>• Podstawy języka SQL, zasady składni i podstawowe instrukcje.</li> </ul>	2
<p>Temat: Obsługa SQLowej bazy danych na podstawie MySQL i phpMyAdmin</p> <ul style="list-style-type: none"> <li>• Import i eksport danych</li> <li>• Mechanizmy ochrony bazy danych</li> </ul>	2
<p>Temat: Przygotowanie projektu prostej bazy danych</p>	2
<b>Grafika i multimedia</b>	
<p>Temat: Obróbka grafiki rastrowej</p> <ul style="list-style-type: none"> <li>• Pozyskiwanie grafiki z Internetu, skanera, aparatu cyfrowego, dopasowywanie wielkości obrazka, przycinanie, kompozycja</li> <li>• Edytor grafiki wektorowej</li> <li>• Objaśnienie pojęć grafika rastrowa i grafika wektorowa, rysowanie linii krzywych i odcinków, rysowanie łamanych, rysowanie figur zamkniętych</li> <li>• Skalowanie figur, lustrzane odbicie w pionie i poziomie, kopiowanie figur, powtarzanie operacji kopiowania, obracanie i ścinanie figur</li> </ul>	3
<p>Temat: Zastosowania figur wektorowych w prostych projektach graficznych</p> <ul style="list-style-type: none"> <li>• Edytor grafiki wektorowej</li> <li>• Zastosowania efektów w prostych projektach graficznych</li> <li>• Eksport rysunku wektorowego do formatu rastrowego</li> <li>• Tworzenie prostych projektów graficznych</li> </ul>	4

<p>Temat: Zastosowania edycji kształtu krzywej w prostych projektach graficznych</p> <ul style="list-style-type: none"> <li>• Krzywe Beziera w grafice wektorowej</li> <li>• Morphing</li> </ul>	2
<p>Temat: Rodzaje grafik - rastrowa (GIF, JPG) i wektorowa, różnice, zalety i wady</p> <ul style="list-style-type: none"> <li>• Wykorzystanie arkusza kalkulacyjnego oraz programu GeoGebra do wizualizacji prostych zależności funkcyjnych</li> </ul>	1
<p>Temat: Tworzenie prezentacji multimedialnej (np.: w Prezi)</p> <ul style="list-style-type: none"> <li>• Zasady tworzenia prezentacji, animacja slajdów oraz ich elementów, tworzenie ścieżki, osadzanie grafiki, muzyki i filmów w prezentacji</li> <li>• Projekt prezentacji na zadany temat</li> </ul>	3
<p>Temat: Elementy języka HTML 5; arkusze stylów CSS 3.0</p> <ul style="list-style-type: none"> <li>• Idea scentralizowanego formatowania, tworzenie prostego arkusza stylu</li> <li>• Przygotowanie prezentacji w oparciu o impress.js</li> <li>• Projekt prezentacji z formatowaniem CSS na zadany temat</li> </ul>	5
<p>Temat: Animacje komputerowe</p> <ul style="list-style-type: none"> <li>• Budowa ekranu roboczego edytora Vectorian Giotto</li> <li>• Tworzenie animacji poklatkowych</li> <li>• Tworzenie animacji tween</li> <li>• Animacje wielowarstwowe</li> </ul>	2
<p>Temat: Język programowania Action Script</p> <ul style="list-style-type: none"> <li>• Wstawianie kodu ActionScript do animacji Flash</li> <li>• Podobieństwo Action Script do C++</li> <li>• Tworzenie animacji wykorzystujących programowanie</li> </ul>	2
<p>Temat: Język PHP</p> <ul style="list-style-type: none"> <li>• Oprogramowanie po stronie serwera - zalety, bezpieczeństwo kodu</li> <li>• Zasada działania preprocesora - przetwarzanie kodu PHP, osadzanie skryptu PHP w kodzie HTML strony - konfiguracja serwera HTTP do pracy ze stronami zawierającymi kod PHP</li> <li>• Podobieństwo języka PHP do C++</li> </ul>	5
<p>Temat: Proste projekty PHP</p> <ul style="list-style-type: none"> <li>• przetwarzanie formularzy,</li> <li>• przekazywanie parametrów w URL strony,</li> <li>• dostęp do plików,</li> <li>• prosty licznik odwiedzin strony WWW, itp.</li> </ul>	4

<ul style="list-style-type: none"><li>Projekt strony zawierającej skrypty PHP</li></ul>	
<b>Powtórzenie, utrwalenie materiału</b>	
Lekcje do dyspozycji nauczyciela <ul style="list-style-type: none"><li>Godziny przeznaczone do dyspozycji nauczyciela powinny zostać wykorzystane na przygotowanie do matury - rozwiązywanie zadań maturalnych</li></ul>	10

## V. Sposoby osiągnięcia celów kształcenia i wychowania

Umiejętność uczenia się jest bardzo przydatna w zmieniającej się rzeczywistości. Świat zmienia się teraz znacznie szybciej niż w poprzednich wiekach. Codziennie otrzymujemy znacznie więcej informacji niż chcemy. Informatyka uczy takiego myślenia, które ułatwia przystosowanie się do dużego tempa zmian. Pozwala porządkować zdobyte informacje, dokonywać ich selekcji. Uczy precyzyjnego wysławiania się, a w rezultacie pomaga w komunikowaniu się z ludźmi. Z tych powodów należy starannie dobrać sposoby osiągnięcia sukcesu każdego ucznia na lekcjach.

Sposoby, metody i formy pracy zależą od potrzeb i możliwości uczniów. Pierwszym krokiem powinno być zatem wstępne zdiagnozowanie możliwości i ustalenie oczekiwań uczniów. Na tej podstawie można zaczerpnąć z szerokiej gamy form i metod kształcenia. Krokiem następnym jest odpowiednie motywowanie uczniów. Służą temu dobrze dobrane sposoby oceniania, ale również osobiste zaangażowanie nauczyciela i dobre relacje między uczniami i nauczycielem. Znakomitym sposobem jest ustalenie na samym początku kilku nienaruszalnych zasad współpracy. Nie może być ich za dużo, wzorem liczności może być dekalog. Zdecydowanie najlepiej zacząć od takich zasad, które zaproponują uczniowie. Później można dorzucić te, na których zależy nauczycielowi. Takie ustalenia pozwolą dbać o dobrą atmosferę pracy i korygować niewłaściwe zachowania uczniów.

Ważnym sposobem pozyskiwania wiedzy może być korzystanie z platformy e-learningowej. Jest to nowy sposób samodzielnego zdobywania wiedzy, dość intuicyjny dla uczniów, którzy przyzwyczaili się do poszukiwania wiadomości w internecie. Powinni więc nauczyć się korzystać z zasobów udostępnianych przez nauczyciela – skupić uwagę w dłuższym odcinku czasowym i pojąć cudzy tok rozumowania. Metoda ta nie może być nadużywana, może jedynie stanowić uzupełnienie lekcji, może to być praca domowa, wiadomości uzupełniające i wykraczające poza dość sztywne ramy lekcji. Uczymy w ten sposób korzystać samodzielnie z podręcznika, encyklopedii, słowników, czasopism popularnonaukowych, oraz informacji z sieci. Metoda ta ma na celu przyzwyczajanie uczniów do zbierania informacji z różnych źródeł, przetwarzania ich, analizowania definicji i twierdzeń, śledzenia cudzych rozwiązań, analizowania algorytmów. Dzięki zastosowaniu platformy e-learningowej możliwe jest skuteczne indywidualizowanie pracy z uczniami o różnych potrzebach edukacyjnych. Można im przygotować zadania odpowiadające ich możliwościom lub aspiracjom bez konieczności zabierania dodatkowego czasu na lekcji.

Zdecydowanie częściej należy używać dyskusji. Pozwala to na większą aktywność uczniów, możliwość zadawania pytań, spierania się, wyciągania wniosków. Ustalenia zasad dyskusji i kontrolowanie ich przestrzegania pozwala na realizację ważnych aspektów wychowawczych. Często dyskusja może być urozmaicona przedstawieniem problemu w formie animacji komputerowej, warto wykorzystać zasoby edukacyjne, szczególnie animacje oraz e-lekcje, zawarte w zasobach portalu <http://www.scholaris.pl>.

Metoda problemowa daje największy przyrost wiedzy. Uczniowie zmuszeni są do dużego wysiłku intelektualnego. Są postawieni wobec nieznanego drogi postępowania. Muszą samodzielnie zbudować model rozwiązania problemu, stawiają hipotezy i je weryfikują. To aktywizująca metoda kształcenia, która służy rozwijaniu umiejętności samodzielnego rozwiązywania zadań.



Szczególnie trudne jest uczenie algorytmiki i programowania. Są to te dziedziny informatyki, które wymagają myślenia abstrakcyjnego, uogólniania znalezionej odpowiedzi i twórczego podejścia do problemu. Doświadczenie wskazuje, że uczniowie mają najczęściej trudności z zadaniami z programowania. Dobrym sposobem przełamania tej niechęci jest zastosowanie zdrowej rywalizacji i wykorzystanie systemów konkursowych. Idealnym rozwiązaniem jest posiadanie narzędzia testującego na serwerze szkolnym. Niestety w takim narzędziu wyposażone są wyłącznie niektóre szkoły, samo zaś skonструowanie narzędzia nie jest proste. Stąd, szczególnie początkującym nauczycielom polecam wykorzystanie serwisu <http://spoj.pl>, gdzie można założyć darmowe konto, poprosić o uprawnienia administracyjne i przygotowywać zawody programistyczne dla uczniów w oparciu o gotowe zadania. Można również wykorzystać system <http://scarky.com/>, który jest łatwy w obsłudze, nawet dla początkujących „układaczy zadań”. Zresztą udostępniona jest dobrze opracowana pomoc: <http://contest.pl/2009/09/08/scarky-pomaga-uczyc-informatyki/>.

Polecam użycie tego systemu szczególnie wtedy, gdy uczniowie rozpoczynają swoją przygodę z programowaniem. Jest to sposób na uatrakcyjnienie lekcji oraz na zdrową rywalizację między nimi.

Formy pracy na lekcji też powinny być różnorodne. Praca z całą klasą pogłębia więzi między uczniami. Nauczyciel realizuje z całą klasą te same treści, jednocześnie stara się zaangażować wszystkich uczniów do rozwiązywania problemu. Praca w grupach uczy organizacji pracy, podziału obowiązków między członków grupy, odpowiedzialności za powierzone zadania. W takich grupach można również zrealizować projekt. To umożliwi korzystanie z różnych źródeł informacji, wykorzystanie komputera, przygotowanie prezentacji. Praca indywidualna ucznia może odbyć się z użyciem karty pracy na lekcji, systemu testującego, platformy e-learningowej. Jest równie ważna, pozwala samodzielnie szukać odpowiedzi na postawione pytania, zmusza do refleksji, kształcenia umiejętności uczenia się.

Przedstawiona w poprzednim punkcie kolejność materiału nauczania nie jest przypadkowa. Na początku określamy sposoby zapisu informacji w komputerze, model działania komputera w oparciu o maszynę RAM – tu pokazujemy uczniowi „jak myśli i działa komputer”. Zdecydowanie ułatwia to przejście do algorytmiki i programowania.

W kolejnym etapie uczeń poznaje podstawowe algorytmy oraz ich komputerowe realizacje zapisane w dowolnym języku programowania. Preferowany język programowania to C++, ponieważ jest on niezwykle przydatny w różnorodnych konkursach, olimpiadach oraz umożliwia przygotowanie się do egzaminu maturalnego.

Zagadnienia poświęcone systemom operacyjnym oraz komunikacji w sieci komputerowej umieszczono w dalszej części programu i potraktowano głównie jako rozszerzenie już posiadanej wiedzy z zakresu podstawowego.

Zagadnienia poruszane w kolejnych punktach programu (stosowanie podejścia algorytmicznego w rozwiązywaniu problemów, tworzenie projektów graficznych i prezentacji multimedialnych, „programowanie” stron WWW) można omawiać na końcu. Stanowią one swoistą klamrę spinającą wszystkie omawiane zagadnienia – pokazują możliwości i zastosowania myślenia algorytmicznego a także pomagają w podjęciu decyzji dotyczących ewentualnego dalszego kształcenia.

## VI. Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne

Ocenianie osiągnięć edukacyjnych uczniów polega na rozpoznawaniu przez nauczycieli poziomu i postępów w opanowaniu przez ucznia wiadomości i umiejętności w stosunku do wymagań edukacyjnych wynikających z podstawy programowej [...] i realizowanych w szkole programów nauczania uwzględniających tę podstawę<sup>1</sup>. Przedmiotem oceniania powinien być przyrost wiedzy ucznia, nigdy natomiast relacje między nauczycielem a uczniem. Ten aspekt oceny należy szczegółowo omówić z uczniami. Wymagania dotyczące poszczególnych ocen powinny być zgodne z wewnątrzszkolnym systemem oceniania, dostępne uczniom i ich rodzicom w trakcie całego roku szkolnego.

Uczniowie są szczególnie wrażliwi na sprawiedliwość oceniania. Na początku każdego roku szkolnego trzeba bardzo rzetelnie je omówić. Można zaproponować średnią ważoną. Ten model ma następującą zaletę: jest najczęściej postrzegany przez uczniów jako sprawiedliwy. Jest prosty, zrozumiały i klarowny. Nie należy oczywiście w sposób automatyczny korzystać ze średniej ważonej, która przypisuje wartość liczbową ocenie szkolnej. Ma ona raczej stanowić punkt wyjścia do ustalenia oceny końcowej. Dodatkowo trzeba uwzględnić zaangażowanie ucznia, aktywność w zdobywaniu wiedzy, udział w konkursach przedmiotowych itp. o kształcie oceny końcowej decyduje nauczyciel uwzględniając wszystkie powyżej wymienione aspekty.

Pracę w grupach trudniej tak klarownie ocenić. Może to być średnia ocen pisemnego opracowania efektów pracy grupy (przygotowana przez sekretarza grupy) i odpowiedzi wybranego przez nauczyciela przedstawiciela grupy. Spowoduje to większe zaangażowanie wszystkich uczniów, którzy będą się starali owocnie współpracować.

Wprowadzając obowiązujący system oceniania opiera się na sześciostopniowej skali ocen, trudno jednak uznać za tak samo ważną ocenę otrzymaną za zadanie trudne jak za zadanie proste. Proponuję zatem wprowadzenie stosownych wag do wystawionej oceny:

waga x 5 – za pracę klasową (to zadanie trudne lub bardzo pracochłonne obejmujące wiedzę i umiejętności dotyczące całego działu);

waga x 3 – za sprawdzian (to zadanie trudne lub bardzo pracochłonne obejmujące wiedzę i umiejętności dotyczące jednego zagadnienia);

waga x 2 – za kartkówkę (to zadanie niezbyt trudne lub niezbyt rozbudowane);

waga x 1 – za pracę na lekcji lub zadanie domowe (zadanie proste i niezbyt rozbudowane).

Elementem pomocniczym przy określaniu oceny śródsesemstralnej i za semestr II w klasach i jest średnia ważona, na podstawie której wyznaczana jest ocena śródsesemstralna i końcoworoczna. Jest ona obliczana zgodnie z następującym algorytmem:

<sup>1</sup> Rozporządzenie Ministra Edukacji Narodowej z 26 sierpnia 2011 roku zmieniające rozporządzenie w sprawie warunków i sposobu oceniania, klasyfikowania i promowania uczniów i słuchaczy oraz przeprowadzania sprawdzianów i egzaminów w szkołach publicznych. Dziennik Ustaw z 2011 roku, Nr 179, poz. 1063.

## Średnia ważona

Średnią ważoną n liczb  $a_1, a_2, \dots, a_n$ , z których każda ma przyporządkowaną pewną nieujemną wagę  $w_1, w_2, \dots, w_n$  nazywamy liczbę

$$\text{średnia}_w = \frac{\sum_{i=1}^n w_i * o_i}{\sum_{i=1}^n w_i} = \frac{w_1 * o_1 + w_2 * o_2 + \dots + w_i * o_i}{w_1 + w_2 + \dots + w_i}$$

gdzie:  $w_i$  – to waga oceny,  $o_i$  – liczbowy odpowiednik oceny (6 – celujący, 5 – bardzo dobry, 4 – dobry, 3 – dostateczny, 2 – dopuszczający, 1 – niedostateczny); dopuszcza się oceny z plusem (+) lub minusem (-); plus (+) podwyższa liczbowy odpowiednik oceny o 1/3, minus (-) obniża liczbowy odpowiednik oceny o 1/3.

Ocena wszystkie oceny końcowe (śródsesemestralne i końcoworoczne) wyznaczane są zgodnie z warunkami określonymi poniżej.

- < 1,5 niedostateczny
- < 2,5 dopuszczający
- < 3,5 dostateczny
- < 4,5 dobry
- < 5,5 bardzo dobry

Ocenę celujący może otrzymać uczeń uzyskujący miejsce 1 – 5 w konkursie rangi wojewódzkiej lub miejsce 1 – 10 w konkursie ogólnopolskim lub uzyska tytuł finalisty lub laureata Ogólnopolskiej Olimpiady Informatycznej.

### Kryteria na poszczególne oceny:

- ❖ - ocenę **dopuszczającą** otrzymuje uczeń, który wykonuje ćwiczenie z pomocą nauczyciela lub kolegi z zespołu, niedbale korzysta z narzędzi i nie stosuje wszystkich zaleceń wynikających z zadania, nie dba o estetykę dokumentów, nie potrafi współpracować z zespołem
- ❖ - ocenę **dostateczną** otrzymuje uczeń, który poprawnie użył określonych w zadaniu narzędzi, lecz nie wykorzystał ich wszystkich możliwości, w końcowym dokumencie lub projekcie występują niedoskonałości wynikające z niedbałości lub nieumiejętnego korzystania z narzędzi edytora, ma problemy ze współpracą z członkami zespołu
- ❖ - ocenę **dobrą** otrzymuje uczeń, który samodzielnie wykonuje zadanie, dobrze współpracuje z członkami zespołu, oddaje prace w określonym terminie, projekty mają drobne niedoskonałości, występują nieuzasadnione odstępstwa od założeń zadania
- ❖ - ocenę **bardzo dobrą** powinien uzyskać uczeń: samodzielnie wykonujący ćwiczenie lub swoje zadania w zespole, oddający prace w wyznaczonym terminie, używający wszystkich możliwych opcji narzędzi do poprawy jakości wykonanej pracy, oddający prace estetyczne bez wad, zawierające niewpływające na jakość pracy błędy lub niedociągnięcia
- ❖ - ocenę **celującą** powinien uzyskać uczeń: perfekcyjnie wykonujący założenia danego projektu lub zadania, oddający prace w wyznaczonym czasie, pracujący samodzielnie lub spełniający wzorowo swoje zadania w zespole, np. lidera grupy; w przypadku zadań graficznych powinno być spełnione kryterium estetyki i zgodności z zasadami tworzenia danego rodzaju grafiki oraz prawidłowe użycie narzędzi edytorów

Na ocenę powinna mieć też wpływ ogólna postawa ucznia i kultura informatyczna. Jest to subiektywny czynnik, dlatego nauczyciel powinien stosować go z rozwagą.

Bardzo dobrym narzędziem do oceniania, archiwizowania i opisywania prac oraz przeprowadzania sprawdzianów wiedzy i umiejętności jest testerka szkolna oraz platforma e-learningowa. Stosowanie tych narzędzi znacznie poprawia komfort pracy nauczyciela i pozwala uporządkować dokumentację zajęć. Sprawdziany – testy przeprowadzane za pośrednictwem platformy lub testerki sprawdzane są automatycznie i oceniane według kryteriów ustalonych przez prowadzącego zajęcia. Nauczyciel może kontrolować terminowość oddawania prac i uzasadniać wystawioną ocenę. Możliwe jest też ocenianie prac po upływie pewnego czasu, co dodatkowo zwiększa komfort pracy.

## **VII. Propozycje metod sprawdzania osiągnięć ucznia**

Należy tak organizować pracę na lekcji (przygotowywać odpowiednie ćwiczenia), aby również uczniowie mniej zaawansowani mieli szansę zdobycia najwyższej oceny i nie czuli się dyskryminowani. Wszystkim należy zapewnić sprawiedliwy sposób oceniania. Trzeba zwracać szczególną uwagę na postępy w zdobywaniu wiedzy. Dla zainteresowanych można przygotować ćwiczenia trudniejsze, aby mogli wykazać się swoimi umiejętnościami i wiedzą. Uczniom mającym trudności z wykonywaniem ćwiczeń i poleceń należy pomagać bezpośrednio podczas zajęć, motywować ich i zachęcać do dalszej pracy.

Uczniowie powinni otrzymywać dużo ocen cząstkowych. Należy oceniać np.:

- ćwiczenia wykonywane podczas lekcji,
- odpowiedzi na pytania,
- udział w projekcie grupowym,
- zadania domowe,
- aktywność na lekcji,
- ćwiczenia sprawdzające
- przykładowe rozwiązania zadań na szkolnej testerce

Dla szczególnie zainteresowanych informatyką polecamy zadania dostępne w serwisie Młodzieżowej Akademii Informatycznej: <http://main.edu.pl/pl>, a uzdolnionym podsuwamy „niebieskie książeczki” z Olimpiady Informatycznej i zachęcamy do udziału w zajęciach dodatkowych, pozalekcyjnych.

## **VIII. Uwagi o realizacji**

### **Warunki realizacji programu**

Aby poprawnie i w pełni zrealizować przedstawiony program niezbędna jest pracownia komputerowa średniej klasy – autorka korzysta z pracowni otrzymanej w ramach doposażenia szkół w 2007 roku. Dodatkowym wyposażeniem jest szkolna testerka programów oraz platformy e-learningowej. W przypadku problemów można skorzystać z gotowego i darmowego rozwiązania dostępnego pod adresem: <http://www.mdl2.com/>

## Oprogramowanie

Program może być zrealizowany przy użyciu programów komputerowych dostępnych w większości szkolnych pracowni komputerowych, w zasadzie opartych na dowolnym systemie operacyjnym. Nauczyciel powinien także sięgać do oprogramowania dostępnego na licencji OpenSource oraz wykorzystywać możliwości oprogramowania online

**Poniższa tabela podaje przykłady programów pozwalających na pełną realizację programu nauczania i dostępnych w wersji online lub darmowych.**

NAZWA PROGRAMU	OPIS
Maszyna RAM	Program będący symulacją działania komputera, do pobrania ze strony <a href="http://www.szakup.com">www.szakup.com</a>
<a href="http://www.eyeos.org">www.eyeos.org</a>	System operacyjny chmury internetowej dostępny wyłącznie on-line
Linux Ubuntu	System operacyjny, który należy pobrać w najnowszej wersji
<a href="http://prezi.com">http://prezi.com</a>	Serwis online do tworzenia prezentacji multimedialnych
<a href="http://edu.glogster.com/">http://edu.glogster.com/</a>	serwis online, w którym można tworzyć interaktywne plakaty (technologia Flash).
VisualRoute Lite Edition	Program do śledzenia połączeń w sieci
Packet Tracer	Aplikacja do obrazowania i symulacji sieci komputerowej
Easy Graphic Converter	Program do konwersji i zmiany parametrów grafiki komputerowej
Picasa	Przeglądarka do zdjęć i grafiki z opcjami edycji
InkScape	Edytor grafiki wektorowej
GIMP	Edytor grafiki rastrowej
<a href="http://pixlr.com/">http://pixlr.com/</a>	Edytor grafiki rastrowej i fotografii z chmury informatycznej
SketchUp	Edytor grafiki 3D
Blender	Edytor grafiki 3D
CadStd Lite	Edytor wspomagający projektowanie typu CAD
CodeBlocks	Edytor środowiska programistycznego
<a href="https://www.lucidchart.com/">https://www.lucidchart.com/</a>	Narzędzie online do wizualizacji, tworzenia map myśli, schematów
<a href="http://www.codebox.es/pdf-to-flash-page-flip">http://www.codebox.es/pdf-to-flash-page-flip</a>	Narzędzie do konwersji plików pdf do flash'a
<a href="http://www.mdl2.com/">http://www.mdl2.com/</a>	Darmowa platforma modle
<a href="http://search.creativecommons.org/">http://search.creativecommons.org/</a>	Wyszukiwarka udostępniana przez organizację Creative Commons do znajdowania plików na licencji Creative Commons.
<a href="http://drawisland.com/">http://drawisland.com/</a>	Program online do tworzenia animacji
<a href="http://spoj.pl/">http://spoj.pl/</a>	Testerka online

<a href="http://scarky.com/">http://scarky.com/</a>	System online do tworzenia własnych zadań programistycznych
<a href="http://www.scholaris.pl/">http://www.scholaris.pl/</a>	Zasoby dla nauczycieli: e-lekcje, prezentacje i animacje
<a href="http://main.edu.pl/pl">http://main.edu.pl/pl</a>	Zasoby Młodzieżowej Akademii Informatycznej – zadania dla zainteresowanych algorytmiką



OŚRODEK  
ROZWOJU  
EDUKACJI

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

[www.ore.edu.pl](http://www.ore.edu.pl)



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY

