Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autorzy oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autorzy oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Joanna Zaręba

Projekt okładki: ULABUKA

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie?prpgw2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-2825-4

Copyright © Helion 2013

Wydanie II

Printed in Poland.

Spis treści

1. Informatyka na różnych poziomach nauczania **5**

2. Charakterystyka i założenia programu **7**

3. Podstawa programowa przedmiotu informatyka. IV etap edukacyjny — zakres rozszerzony **11**
3.1. Cele kształcenia — wymagania ogólne 11
3.2. Treści nauczania — wymagania szczegółowe 11

4. Cele kształcenia i wychowania **16**
4.1. Cele edukacyjne 17
4.2. Cele wychowawcze 22

5. Procedury osiągania celów **23**
5.1. Osiąganie celów edukacyjnych 23
5.2. Propozycje metod nauczania 24
5.3. Osiąganie celów wychowawczych 26

6. Zadania szkoły w zakresie edukacji informatycznej **27**
6.1. Baza sprzętowa — wyposażenie szkolnej pracowni komputerowej 28
6.2. Oprogramowanie 28

7. Ramowy plan nauczania **29**

8. Treści nauczania i przewidywane osiągnięcia uczniów **31**
8.1. Algorytmika 31
8.2. Programowanie 40
8.3. Bazy danych 44
8.4. Multimedia i grafika komputerowa 48

8.5. Systemy operacyjne i sieci komputerowe 50
8.6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki 53

9. Metody kontroli i oceny osiągnięć uczniów 54
9.1. Metody oceniania osiągnięć edukacyjnych 54
9.2. Kryteria oceniania 58

10. Zakres możliwych modyfikacji programu nauczania 61

1. Informatyka na różnych poziomach nauczania

Jednym z głównych celów edukacji informatycznej jest stworzenie sytuacji, w której każdy człowiek będzie dążył do zwiększenia swoich umiejętności w zakresie posługiwania się technologią informacyjno-komunikacyjną. Człowiek XXI wieku powinien opanować ją w takim samym stopniu jak podstawowe umiejętności niezbędne do życia w społeczeństwie, takie jak czytanie, pisanie czy liczenie. Technologia informacyjno-komunikacyjna staje się dziedziną wiedzy nieodzowną w społeczeństwie informacyjnym, a ponadto stanowi środek do przyspieszenia indywidualnego rozwoju człowieka. Aby osiągnąć wymienione tu cele na poziomie edukacji, należy stworzyć uczniom możliwości poznania podstaw i zastosowań tej technologii oraz przygotowania się do jej wykorzystania w praktyce.

Z technologią informacyjno-komunikacyjną mamy do czynienia w sytuacji, gdy korzystamy z narzędzi informatycznych. Informatyka pojawia się w momencie, gdy zaczynamy tworzyć coś nowego przy użyciu istniejących narzędzi. Granica jest nieostra i czasami trudno zauważyć, że już ją przekroczyliśmy. Wyróżniająca informatykę potrzeba twórczego i jednocześnie logicznego myślenia powoduje, że w tę dyscyplinę angażują się ludzie ze specyficznymi predyspozycjami i uzdolnieniami. Nie każdy może się więc skutecznie rozwijać w tym zakresie. Z kolei technologia informacyjno-komunikacyjna wymaga jedynie ćwiczeń i wiedzy, jest zatem przeznaczona dla wszystkich, którzy chcą z niej korzystać.

W ostatnich latach nastąpił niezwykle szybki rozwój nauk informatycznych i ich zastosowań. Technologia informacyjno-komunikacyjna i informatyka to dziedziny, których wykorzystanie i dostępność stale wzrastają, a tempo zachodzących zmian jest tu nieporównywalne z innymi dyscyplinami. Widać to również w dziedzinie edukacji informatycznej. Stale obniża się wiek ucznia rozpoczynającego rozwój w tym kierunku, w związku z czym zmieniają się warunki, w jakich odbywa się edukacja informatyczna.

Przedmioty informatyczne pojawiają się w procesie edukacji na coraz niższych poziomach: w szkole podstawowej już w okresie wczesnoszkolnym. Kształcenie w tym kierunku ma ogromny wpływ na rozwój ucznia, dlatego należy zwrócić szczególną uwagę, by było ono prawidłowo ukierunkowane. Młody człowiek musi mieć świadomość istniejących zagrożeń i samodzielnie dokonywać odpowiedniego wyboru narzędzi informatycznych. Rola nauczyciela prowadzącego zajęcia komputerowe w początkowym okresie jest niezwykle istotna, od niego bowiem zależy prawidłowy

rozwój ucznia w tym zakresie. Bardzo trudno naprawić błędy popełnione na samym początku nauki. Uczniowie na ogół nie wiedzą, z jak szeroką dziedziną mają do czynienia, nie zdają sobie sprawy z istnienia informatyki i technologii informacyjno-komunikacyjnej oraz różnic dzielących te dwa pojęcia.

Na wszystkich poziomach edukacji informatycznej niezwykle istotna jest ciągłość nauczania. Kolejne etapy powinny być kontynuacją i rozszerzeniem przygotowania informatycznego rozpoczętego we wcześniejszym okresie. Nauczyciele prowadzący zajęcia na poziomie szkoły podstawowej i gimnazjum powinni właściwie realizować programy nauczania, uwzględniając podstawę programową obowiązującą na danym etapie.

Edukacja informatyczna obecnie rozpoczyna się w szkole podstawowej, a następnie jest kontynuowana w gimnazjum. Na pierwszym i drugim etapie edukacyjnym, a więc na poziomie szkoły podstawowej, prowadzone są „zajęcia komputerowe". Na trzecim i czwartym etapie kształcenia informatycznego, czyli w gimnazjum i szkole ponadgimnazjalnej, przedmiot nosi nazwę „informatyka". Zajęcia komputerowe obejmują wyłącznie zagadnienia związane z technologiami informacyjno-komunikacyjnymi. W zakresie informatyki, zarówno w szkole gimnazjalnej, jak i ponadgimnazjalnej, oprócz kontynuacji zagadnień dotyczących technologii informacyjno-komunikacyjnych pojawiają się zagadnienia informatyczne związane z algorytmiką i programowaniem.

Warto tu zwrócić uwagę na odrębność przedmiotów informatycznych prowadzonych w szkole ponadgimnazjalnej: „informatyka na poziomie podstawowym" i „informatyka na poziomie rozszerzonym".

Informatyka realizowana na poziomie podstawowym:

- jest przedmiotem obowiązkowym prowadzonym we wszystkich klasach,
- nie jest przedmiotem maturalnym,
- stanowi poziom podstawowy dla przedmiotu „informatyka na poziomie rozszerzonym", a więc wchodzi w zakres wymagań maturalnych tego przedmiotu.

Informatyka prowadzona na poziomie rozszerzonym:

- **nie jest przedmiotem obowiązkowym**, a więc jest realizowana wyłącznie w klasach, w których zaplanowano rozszerzenie edukacji informatycznej,

- **jest przedmiotem maturalnym** znajdującym się w grupie przedmiotów dodatkowych, stąd **egzamin maturalny z informatyki można zdawać wyłącznie na poziomie rozszerzonym**.

2. Charakterystyka i założenia programu

Program nauczania realizowany w klasach przygotowujących ucznia szkoły ponadgimnazjalnej do egzaminu maturalnego z informatyki musi obejmować podstawę programową obowiązującą na takim egzaminie oraz pracę z uczniem w zakresie rozwoju jego samodzielnego i twórczego myślenia, rozwiązywania problemów, konstruowania algorytmów. W konsekwencji absolwenci tych klas powinni być dobrze przygotowani do studiów informatycznych i innych w dziedzinie nauk ścisłych.

Przedstawiony tu program zawiera materiał przeznaczony dla szkół ponadgimnazjalnych. Został on tak dobrany, aby przygotować ucznia do kontynuacji nauki na studiach wyższych. Realizacja programu powinna pomóc mu w podjęciu decyzji o kontynuowaniu nauki oraz o wyborze kierunku studiów. Duży nacisk położono więc na takie zagadnienia jak:

- stosowanie podejścia algorytmicznego przy rozwiązywaniu problemów oraz podejmowaniu decyzji z wykorzystaniem komputera,
- wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji,
- opracowywanie informacji za pomocą komputera, w tym: danych liczbowych, obrazów, dźwięków i filmów,
- współtworzenie zasobów w sieci,
- posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej.

Powyższe zagadnienia są w szczególny sposób związane z realizacją przedmiotu „informatyka" w zakresie rozszerzonym oraz pozwalają młodemu człowiekowi na dokonanie świadomego wyboru własnej drogi zawodowej. Wiele zagadnień wchodzących w zakres tego przedmiotu jest rozwinięciem tematów poruszanych już w ramach poziomu podstawowego i wcześniejszych etapów kształcenia (zwłaszcza etapu III, realizowanego w gimnazjum).

Proponowany program przewiduje właściwe przygotowanie ucznia do egzaminu maturalnego z informatyki, co wiąże się bezpośrednio z wyborem dalszego kierunku kształcenia. Ważna jest również odpowiednia konstrukcja prezentowanego tu programu — jego realizacja nie może wykraczać poza dostępne możliwości.

Ramowy układ materiału nauczania pokazuje, jaką wiedzę i umiejętności musi zdobyć uczeń, aby przystąpić do egzaminu maturalnego z informatyki na poziomie rozszerzonym. Wskazuje sposób realizacji podstawy programowej w tym zakresie. Wymaga od ucznia samodzielnej pracy, a od nauczyciela prawidłowego ukierunkowania rozwoju i działań podopiecznego. Jeśli nauczyciel chce przygotować ucznia do egzaminu maturalnego z informatyki, nie uniknie konieczności zlecania mu dodatkowej pracy w domu. Zaproponowany podział godzin umożliwia przeznaczenie większej liczby lekcji na ćwiczenia, zadania i sprawdziany tworzone na wzór zadań maturalnych, co w znacznym stopniu ułatwi pracę nauczyciela. Ponadto umożliwia realizację projektów indywidualnych i zespołowych, co również jest istotne w procesie prawidłowego rozwoju młodego człowieka.

Nie wszyscy uczniowie są zainteresowani egzaminem maturalnym z informatyki. Wybierając jednak ten przedmiot, mają nadzieję na intensywny rozwój w tym kierunku. To obejmuje między innymi rozwijanie szybkiego, logicznego myślenia oraz umiejętności samodzielnych działań w różnych dziedzinach z wykorzystaniem komputera i narzędzi informatycznych. Współczesny świat oczekuje od absolwenta szkoły ponadgimnazjalnej podstawowych wiadomości z przedmiotów ogólnokształcących, przede wszystkim jednak wymaga umiejętności łączenia wiedzy z różnych dziedzin oraz efektywnego jej wykorzystania. Do osiągnięcia tego celu niezbędny jest dziś komputer z dostępem do internetu. Zastosowanie narzędzi informatycznych pobudza ucznia do twórczego myślenia i proponowania nowych sposobów ich wykorzystania. Realizacja zagadnień informatycznych zwiększa więc rozwój ucznia w różnych dziedzinach, co ma bezpośredni wpływ na jego przyszłość.

Wiedza informatyczna powinna być skorelowana z innymi przedmiotami nauczania. Dzięki temu młody człowiek rozumie znaczenie komputera jako narzędzia usługowego w różnych dziedzinach nauki i życia, a w przyszłości będzie potrafił wybrać i zastosować bądź skonstruować potrzebne mu narzędzia służące do wspomagania pracy lub rozwiązania postawionego problemu. Korelacja z innymi przedmiotami ogólnokształcącymi ma również na celu uświadomienie uczniowi, iż wiedza z różnych przedmiotów wzajemnie się przenika, wspomaga i uzupełnia, a współczesny człowiek musi dysponować wiedzą ogólną — nie może ograniczać się do jednej wąskiej dziedziny nauki. Integracja międzyprzedmiotowa powinna się odbywać na dwóch poziomach, czyli na lekcjach informatyki (przez dobór zadań z różnych dziedzin i kształcenie z ich wykorzystaniem umiejętności informatycznych) oraz na lekcjach innych przedmiotów (przez wykorzystanie umiejętności z dziedziny informatyki do pracy nad wybranymi zagadnieniami).

Jednym z największych wyzwań w nauczaniu informatyki jest wykształcenie umiejętności algorytmicznego myślenia. Często spotykamy się z pytaniem: „W jaki sposób wprowadzać algorytmikę, aby uczniowie zaczęli właściwie wyobrażać sobie algorytmy i wykazywać świadomą inicjatywę w ich realizacji?". Przedstawiony program nauczania opracowano w taki sposób, aby wspomóc nauczyciela w tym zakresie. Algorytmika i programowanie obejmują ponad połowę czasu przeznaczonego na realizację przedmiotu „informatyka" w zakresie rozszerzonym. Pozytywny efekt można uzyskać jedynie dzięki dużej liczbie ćwiczeń praktycznych i właściwej kolejności wprowadzania poszczególnych zagadnień. Materiał dotyczący algorytmiki należałoby zacząć realizować już w klasie drugiej, ale to jest dodatkowo uzależnione od ustalonej siatki godzin. Należy też zwrócić uwagę na fakt, że podstawa programowa dla gimnazjum zakłada wprowadzanie zagadnień algorytmicznych już na tym poziomie kształcenia. Prawidłowa edukacja gimnazjalna w zakresie algorytmiki daje uczniowi szansę świadomego wyboru kontynuacji nauki w tej dziedzinie w szkole ponadgimnazjalnej. Przypadkowy wybór przedmiotu „informatyka" w zakresie rozszerzonym może być przyczyną późniejszych problemów i rozczarowania.

Przedstawiony tu program nauczania został skonstruowany z myślą o wszechstronnej pomocy dla nauczyciela. Jednym z zadań realizowanych przez nauczyciela jest wyrównywanie różnic w zakresie umiejętności uczniów, którzy mieli już kontakt z kształceniem informatycznym na poziomie szkoły podstawowej i gimnazjum. Kolejnym zadaniem jest wdrażanie ucznia do samodzielnej i twórczej pracy, pobudzanie do rozwiązywania problemów, inspirowanie do szybkiego i logicznego myślenia z wykorzystaniem algorytmicznej wyobraźni. Niezwykle istotne jest również przygotowanie ucznia do pracy zespołowej przy wykonywaniu niektórych projektów informatycznych. Umiejętność ta okazuje się przydatna na przykład podczas realizacji projektów programistycznych, w przypadku których mamy do czynienia z inżynierią oprogramowania.

Uczymy samodzielności w rozwiązywaniu problemów, w tym analizy zadania, prawidłowego doboru narzędzi służących do jego wykonania i dążenia do optymalnego rozwiązania, co sprawia, że uczeń, otrzymując odpowiednie narzędzia i materiały, jest pobudzany do rozwoju i kreatywnej pracy. W ten sposób — przy zaangażowaniu zarówno ucznia, jak i nauczyciela — realizacja programu nauczania daje najlepsze efekty.

Opracowany program nauczania zakłada aktywną postawę nauczyciela. Jeśli prowadzący zajęcia będzie chciał dokonać w nim pewnych zmian, może to zrobić zgodnie ze wskazówkami znajdującymi się w rozdziale 10., „Zakres możliwych modyfikacji programu nauczania". Szczególnie korzystna może się okazać korelacja z innymi przedmiotami ścisłymi, przede wszystkim z matematyką i fizyką.

Proponowany tu program wychodzi naprzeciw zmianom wprowadzanym w zakresie edukacji informatycznej na poziomie szkoły ponadgimnazjalnej. Dostosowany jest do wymogów stawianych osobom zdającym egzamin maturalny z informatyki na poziomie rozszerzonym. Obejmuje podstawę programową obowiązującą na egzaminie maturalnym z informatyki od roku szkolnego 2014/2015.

Program ten zawiera elementy niezbędne do jego pełnej realizacji, takie jak podstawa programowa obowiązująca uczniów, którzy wybrali naukę informatyki na poziomie rozszerzonym, ponadto cele kształcenia i wychowania oraz procedury osiągania tych celów, zadania szkoły w zakresie edukacji informatycznej, ramowy podział materiału nauczania i szczegółowe treści nauczania wraz z przewidywanymi osiągnięciami uczniów, analizę metod kontroli i oceny osiągnięć oraz zakres możliwych modyfikacji.

W treściach programowych znajdujących się w rozdziale 8., „Treści nauczania i przewidywane osiągnięcia uczniów", materiał dodatkowy został wyróżniony **pogrubieniem**. Uwzględnione tam zagadnienia nie są zawarte w podstawie programowej.

3. Podstawa programowa przedmiotu informatyka. IV etap edukacyjny — zakres rozszerzony

3.1. 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.

3.2. 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 sieciowa.
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 realizuje 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. Wykorzystywanie komputera oraz programów i gier edukacyjnych 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. Wykorzystywanie komputera i technologii informacyjno-komunikacyjnych do rozwijania własnych zainteresowań, opisywanie zastosowań informatyki, ocenianie zagrożeń i ograniczeń, docenianie aspektów społecznych 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.

4. Cele kształcenia i wychowania

W kolejnych etapach edukacji informatycznej uczniowie powinni nie tylko pogłębiać swoją wiedzę, ale także rozwijać zainteresowania i umiejętności pozwalające między innymi na:

- sprawne posługiwanie się zaawansowanymi środkami, narzędziami i metodami informatyki;
- korzystanie z usług sieci komputerowych;
- wyszukiwanie informacji w bazach danych i tworzenie prostych relacyjnych baz danych;

- formułowanie sytuacji problemowej, jej modelowanie i projektowanie rozwiązania przy użyciu komputera z zastosowaniem podejścia algorytmicznego;
- ocenianie wybranych własności uzyskanych rozwiązań oraz dokumentowanie tych rozwiązań;
- wykorzystywanie sieci do publikowania własnych materiałów oraz zbierania i przetwarzania danych;
- planowanie współpracy i zespołowe wykonywanie projektów informatycznych;
- ocenianie korzyści oraz dostrzeganie etycznych i społecznych aspektów rozwoju zastosowań informatyki.

Nie wolno jednak zapominać o realizacji celów wychowawczych. Przestrzeganie ustalonych zachowań oraz przyjętych zasad jest bardzo ważne w społeczeństwie informacyjnym żyjącym w świecie techniki komputerowej i różnorodnych informacji.

4.1. Cele edukacyjne

Podstawowym celem edukacyjnym jest przygotowanie ucznia do zaawansowanego stosowania środków, narzędzi i metod technologii informacyjnej oraz informatyki w różnych dziedzinach życia i kolejnych etapach kształcenia. Jest to więc dążenie do wykształcenia osób sprawnie funkcjonujących w społeczeństwie informacyjnym, korzystających z technik komunikacji i z różnych źródeł informacji, a także do wspomagania ich w przygotowywaniu się do dorosłego życia, świadomego wyboru kierunku dalszej nauki w zakresie informatyki (lub pokrewnej dziedziny).

Ważnym celem edukacyjnym jest ponadto wykształcenie umiejętności samodzielnego korzystania z komputera przy realizacji wybranych zadań edukacyjnych oraz innych celów poznawczych. Chodzi tu o naukę świadomego i sprawnego posługiwania się komputerem, a także narzędziami i metodami informatyki. Ważne jest również wskazanie uczniowi drogi do celowego i odpowiedzialnego tworzenia warsztatu pracy intelektualnej.

Kolejnym celem edukacyjnym jest przygotowanie ucznia do świadomego wyboru kierunku dalszego kształcenia w zakresie informatyki lub innych, pokrewnych dziedzin.

4.1.1. Ogólne cele edukacyjne — kształcenie w zakresie rozszerzonym

W tabeli 4.1 przedstawiono ogólne cele edukacyjne kształcenia w zakresie rozszerzonym.

Tabela 4.1. *Ogólne cele edukacyjne kształcenia w zakresie rozszerzonym*

DZIAŁ	OGÓLNE CELE EDUKACYJNE
ALGORYTMIKA I PROGRAMOWANIE	Analizowanie i formułowanie informatycznych rozwiązań problemów z różnych dziedzin przez dobór odpowiedniego algorytmu i struktury danych oraz ich realizację w wybranej notacji: języka programowania, schematu blokowego, listy kroków.
	Znajomość podstawowych i złożonych algorytmów oraz stosowanie ich przy rozwiązywaniu problemów.
	Znajomość i umiejętność stosowania podstawowych technik algorytmicznych.
	Stosowanie odpowiednich struktur danych przy realizacji algorytmów, w tym struktur dynamicznych.
	Znajomość i umiejętność stosowania zasad strukturalizacji programów.
	Konstruowanie algorytmów przez projektowanie i realizację poszczególnych etapów prowadzących do otrzymania poprawnego rozwiązania postawionego problemu.
	Konstruowanie prostych baz danych z wykorzystaniem operacji na plikach w wybranym języku programowania.
	Konstruowanie programów w wybranym języku programowania wysokiego poziomu.
	Stosowanie arkusza kalkulacyjnego do wizualizacji zależności funkcyjnych oraz do zapisywania algorytmów.
	Zapisywanie, uruchamianie i testowanie programów w wybranym środowisku programistycznym.
	Określanie, analizowanie i uzasadnianie własności algorytmów, w tym poprawności, złożoności obliczeniowej (czasowej i pamięciowej), efektywności, skończoności oraz optymalności.
	Wskazywanie źródła błędów w obliczeniach komputerowych, w tym wyznaczanie błędu względnego i bezwzględnego.
	Planowanie i realizacja indywidualnych oraz zespołowych projektów programistycznych.
	Znajomość i stosowanie sposobów reprezentowania danych liczbowych w komputerze oraz systemów liczbowych wykorzystywanych w informatyce.

Tabela 4.1. *Ogólne cele edukacyjne kształcenia w zakresie rozszerzonym (cd.)*

DZIAŁ	OGÓLNE CELE EDUKACYJNE
BAZY DANYCH	Znajomość form organizacji informacji w bazach danych.
	Umiejętność wykorzystania operacji bazodanowych arkusza kalkulacyjnego do wyszukiwania danych według określonych kryteriów.
	Znajomość podstawowych pojęć związanych z relacyjnymi bazami danych.
	Umiejętność doboru odpowiedniego programu narzędziowego do rozwiązania zadania.
	Projektowanie relacyjnych, wielotabelowych baz danych z uwzględnieniem odpowiednich związków (relacji) oraz kontroli integralności danych.
	Umiejętność określania zaawansowanych właściwości danych gromadzonych w bazie.
	Znajomość metod wyszukiwania i przetwarzania informacji w relacyjnych bazach danych.
	Umiejętność stosowania zaawansowanych metod wyszukiwania informacji w relacyjnych bazach danych z użyciem języka zapytań (język SQL).
	Definiowanie złożonych kryteriów wyszukiwania oraz podsumowań danych w bazie danych.
	Rozumienie mechanizmów związanych z ochroną i bezpieczeństwem danych.
	Planowanie współpracy i zespołowe wykonywanie projektu relacyjnej bazy danych.
MULTIMEDIA I GRAFIKA KOMPUTEROWA	Umiejętność sprawnego i odpowiedzialnego korzystania z multimediów.
	Umiejętność przetwarzania informacji w różnej postaci, w tym wizualnej i dźwiękowej.
	Znajomość metod kompresji plików multimedialnych. Znajomość pojęć kompresji stratnej i bezstratnej.
	Znajomość podstawowych modeli barw i ich zastosowanie.
	Rozróżnianie oraz znajomość wad i zalet grafiki wektorowej i rastrowej. Znajomość podstawowych formatów plików graficznych i ich zastosowania. Znajomość podstaw tworzenia obrazu 3D.
	Umiejętność tworzenia, przetwarzania i edytowania obrazów rastrowych i wektorowych z uwzględnieniem przekształceń i warstw.
	Umiejętność tworzenia własnych materiałów multimedialnych.

Tabela 4.1. *Ogólne cele edukacyjne kształcenia w zakresie rozszerzonym (cd.)*

DZIAŁ	OGÓLNE CELE EDUKACYJNE
	Umiejętność tworzenia zaawansowanych animacji z wykorzystaniem elementów programowania.
	Znajomość reprezentacji danych nieliczbowych w komputerze (obrazów, animacji, dźwięków, filmu).
	Umiejętność zespołowego tworzenia opracowań i publikacji multimedialnych.
SYSTEMY OPERACYJNE I SIECI KOMPUTEROWE	Znajomość budowy komputera oraz roli jego poszczególnych podzespołów.
	Znajomość podstawowych zadań systemu operacyjnego.
	Znajomość różnych systemów operacyjnych.
	Korzystanie z różnych systemów operacyjnych.
	Znajomość budowy i działania oraz eksploatacja sieci komputerowych.
	Znajomość warstwowego modelu sieci komputerowych.
	Umiejętność określenia ustawień sieciowych danego komputera i jego lokalizacji w sieci.
	Znajomość zasady administrowania siecią komputerową w architekturze klient-serwer.
	Znajomość usług w sieciach komputerowych i sprawne korzystanie z nich.
	Umiejętność wykorzystywania sieci do publikowania własnych materiałów oraz gromadzenia danych.
	Umiejętność tworzenia i publikowania w sieci własnych materiałów oraz opracowań multimedialnych z wykorzystaniem języka skryptowego PHP.
	Umiejętność wyszukiwania i selekcji odpowiednich informacji niezbędnych do realizacji projektów z różnych dziedzin.
	Znajomość zagrożeń związanych z pracą komputera w sieci oraz sposobów zabezpieczania się przed nimi.
	Znajomość mechanizmów związanych z bezpieczeństwem danych: szyfrowanie, klucz, certyfikat, zapora sieciowa.
	Znajomość zagrożeń związanych z przestępczością komputerową, w tym piractwo komputerowe, nielegalne transakcje w sieci.
	Korzystanie z zasobów edukacyjnych udostępnianych w portalach przeznaczonych do kształcenia na odległość.

Tabela 4.1. *Ogólne cele edukacyjne kształcenia w zakresie rozszerzonym (cd.)*

DZIAŁ	OGÓLNE CELE EDUKACYJNE
KIERUNKI ROZWOJU TECHNOLOGII INFORMATYCZNYCH ORAZ ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	Wskazywanie korzyści i zagrożeń wynikających z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.
	Opisywanie najważniejszych elementów procesu rozwoju informatyki i technologii informacyjno-komunikacyjnych.
	Znajomość i przestrzeganie przepisów prawnych dotyczących ochrony własności intelektualnej.
	Przestrzeganie zasad etyki.
	Znajomość najnowszych osiągnięć w zakresie technik informatycznych i tendencji w dalszym ich rozwoju.
	Zapoznanie z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi (nowe programy i systemy oprogramowania).
	Znajomość wielu zastosowań komputerów w różnych dziedzinach życia.

4.1.2. Szczegółowe cele edukacyjne

Cele zawarte w podstawie programowej mają charakter ogólny — nie uściślają zadań i umiejętności ucznia, a jedynie wyznaczają główne kierunki zamierzeń edukacyjnych. Mają więc one charakter hasłowy i nie precyzują zakresu materiału, tematyki zajęć edukacyjnych ani szczegółowych umiejętności uczniów. Na przykład stwierdzenie „definiowanie złożonych kryteriów wyszukiwania oraz podsumowań danych w bazie danych" nie daje nauczycielowi szczegółowych wskazówek, które pozwoliłyby na osiągnięcie tego celu ogólnego. Do tego służą cele szczegółowe, wyraźnie określające, jakie wiadomości i umiejętności mają zostać opanowane. „Uczeń potrafi..." — po tych słowach najczęściej określany jest cel operacyjny mówiący o szczegółowym zadaniu czy umiejętności, jakie uczeń powinien opanować. W rozdziale 8., „Treści nauczania i przewidywane osiągnięcia uczniów", cele szczegółowe zostały przedstawione jako operacyjne, określające wiadomości i umiejętności, czyli zamierzone osiągnięcia edukacyjne. Cele operacyjne opisują więc przewidywane osiągnięcia młodego człowieka przez nazwanie szczegółowych zadań (operacji), które uczeń potrafi wykonać (na przykład: „Uczeń potrafi przy użyciu kreatora sum częściowych dokonać podsumowań danych zawartych w tabeli"). Zaproponowany zestaw celów może być traktowany przez nauczyciela realizującego ten program nauczania jako baza wyjściowa, którą może dodatkowo wzbogacać o kolejne cele.

4.2. Cele wychowawcze

Podczas realizacji programu nauczania należy zwrócić uwagę nie tylko na realizację celów edukacyjnych, ale również na jego bardzo istotne aspekty wychowawcze. Cele wychowawcze należy wiązać z aktualną tematyką zajęć i realizować w całym cyklu kształcenia. Znajomość zasad bezpiecznego korzystania ze sprzętu komputerowego i użytkowania oprogramowania oraz przyjęcie odpowiednich postaw i zachowań są bardzo ważne w czasach wszechstronnego i powszechnego zastosowania komputerów. Dotyczy to zwłaszcza tych uczniów, którzy świadomie podejmują decyzję o wyborze kierunku i zakresu dalszego kształcenia informatycznego.

Wszechobecne i ciągle rozwijające się technologie informacyjno-komunikacyjne wywierają duży wpływ na zachowania społeczne. Dlatego młodzi użytkownicy komputerów i internetu powinni dostrzegać korzyści i zagrożenia związane zarówno z bezpieczeństwem informacji, jak i z odpowiedzialnością za zamieszczane przez siebie treści.

Realizacja celów wychowawczych jest absolutną koniecznością. Nie wolno przy tym pominąć takich celów jak:

- przestrzeganie i propagowanie prawa autorskiego poprzez wpojenie poszanowania własności intelektualnej;
- zwrócenie uwagi na etyczne aspekty przestrzegania prawa autorskiego ze szczególnym naciskiem na publikację w sieci i korzystanie z licencjonowanego oprogramowania;
- przestrzeganie zasad etycznego zachowania się w sieci;
- wyrobienie nawyku weryfikowania informacji zamieszczonych w sieci;
- uświadomienie odpowiedzialności za zamieszczane informacje w sieci;
- przestrzeganie i bezwzględne stosowanie profilaktyki antywirusowej;
- wyrobienie nawyku poszukiwania informacji o nowych wynalazkach i ich zastosowaniach mających wpływ na rozwój technologii informacyjno-komunikacyjnych;
- dostrzeganie zalet i zagrożeń wynikających z rozwoju informatyki i technologii informacyjno-komunikacyjnych;
- przestrzeganie zasad związanych z bezpieczeństwem i higieną pracy;
- kształtowanie umiejętności samodzielnej pracy i współdziałania w zespole;
- przygotowanie do podejmowania samodzielnych decyzji; kształtowanie odpowiedzialności za rozwój osobisty.

5. Procedury osiągania celów

5.1. Osiąganie celów edukacyjnych

Informatyka na poziomie rozszerzonym jest przedmiotem fakultatywnym, co oznacza, że na jej lekcje uczęszczają jedynie ci uczniowie, którzy wybrali taki profil kształcenia.

Od nauczyciela informatyki szkoła i uczniowie oczekują, że — oprócz nauczania i przygotowywania uczniów do egzaminu maturalnego oraz do podjęcia przez nich dalszego kształcenia informatycznego — będzie wzbogacał swój warsztat pracy, ciągle się doskonalił, posiadał wiedzę, umiejętności i doświadczenie niezbędne do nauczania tego przedmiotu w szkole ponadgimnazjalnej. W szczególności oczekuje się, że będzie on:

- znał podstawy informatyki i jej matematycznego podłoża oraz powiązań z innymi dziedzinami nauki;
- znał zaawansowane metody wykorzystywania oraz programowania narzędzi informatycznych, takich jak edytor tekstu, arkusz kalkulacyjny, system zarządzania bazą danych, wybrane programy edukacyjne;
- posiadał praktyczną znajomość przynajmniej dwóch języków programowania wyższego poziomu, w tym języka algorytmicznego;
- znał podstawowe metody algorytmicznego rozwiązywania problemów;
- znał podstawowe struktury danych;
- znał budowę typowej lokalnej sieci komputerowej oraz sieci internet, w tym zasady funkcjonowania architektury klient-serwer;
- miał doświadczenie w zakresie pracy w różnych systemach operacyjnych i systemach sieciowych oraz administrowania siecią komputerową na potrzeby prowadzonych zajęć.

Od nauczyciela wymagane jest również stosowanie różnorodnych i dość specyficznych metod nauczania. Oto kilka wskazówek dotyczących realizacji programu nauczania, w których nauczyciel:

- dostosowuje cele i metody nauczania do zmian zachodzących w informatyce, technologii informatycznej i technologii informacyjnej;
- opracowuje i stosuje w praktyce metody nauczania pojęć i kształtowania umiejętności informatycznych;

- opracowuje i stosuje w praktyce metody modelowania problemów i ich rozwiązywania, przedstawiania rozwiązań w postaci algorytmicznej oraz otrzymywania rozwiązań za pomocą wybranych narzędzi informatycznych, w tym również własnych programów;
- zna podstawowe zastosowania informatyki w innych dziedzinach kształcenia i stosuje w praktyce nauczanie informatyki zintegrowane z innymi przedmiotami oraz współpracuje w tym zakresie z nauczycielami innych przedmiotów;
- projektuje i stosuje w nauczaniu zespołowe metody pracy, w tym zespołowy projekt programistyczny;
- doradza uczniom w zakresie wyboru drogi dalszego kształcenia w szkole wyższej o specjalności informatycznej lub pokrewnej;
- projektuje i rozwija metody oceniania osiągnięć uczniów w zdobywaniu umiejętności i kompetencji informatycznych;
- rozwiązuje wraz z uczniami zadania maturalne i omawia samodzielne rozwiązania uczniów, analizując klucze odpowiedzi i wzorcowe rozwiązania; wskazuje błędy i wyjaśnia przyczyny ich powstawania.

Jak uczyć? Nie ma tu gotowych recept, nie istnieje jedna spójna teoria dydaktyki informatyki. Nie należy jednak zamiast lekcji informatyki prowadzić kursów komputerowych uczących obsługi konkretnego programu, gdyż znacznie ważniejsze są ogólne zasady realizacji pewnych zadań i umiejętność użycia komputera do ich rozwiązywania.

Czego więc uczymy? Ważne są cechy i możliwości programów oraz to, czego należy od nich oczekiwać. Nie uczymy bowiem obsługi edytora tekstu, arkusza kalkulacyjnego, programu graficznego czy prezentacyjnego, lecz edycji tekstu, obliczeń komputerowych, grafiki komputerowej i sposobów przekazywania informacji. Uczymy zatem, do czego służą pewne typy programów, co programy powinny oferować, co można zrobić w danym programie, czego warto szukać, w jaki sposób komputer ułatwia życie.

Jak dobierać zadania dla uczniów? Zadania muszą być konkretne, przydatne i interesujące, skłaniające ucznia do aktywności i zaangażowania.

5.2. Propozycje metod nauczania

Specyfika przedmiotu wymusza stosowanie przez nauczycieli takich sposobów pracy z uczniami, które umożliwiają uczniom opanowanie wiedzy wraz z umiejętnościami posługiwania się nią w praktyce oraz rozwijanie zdolności i zainteresowań. Dlatego na lekcji informatyki będą wykorzystywane

metody oparte na obserwacji, na słowie i przede wszystkim na czynnościach praktycznych.

W myśl założeń reformy mamy przygotowywać uczniów do pracy w zespole, a jednocześnie realizować interdyscyplinarność w praktyce szkolnej. To właśnie na lekcjach informatyki możliwa jest realizacja nauczania interdyscyplinarnego z wykorzystaniem metod aktywizujących.

Jedną z metod aktywnego kształcenia uczniów jest metoda projektów. Projekt jest atrakcyjny, ciekawy i skutecznie aktywizuje zaangażowanych w jego realizację uczniów. Dzieci i młodzież najlepiej budują swoją wiedzę na podstawie przeżytego przez siebie doświadczenia. Mają możliwość nauczenia się planowania swojej pracy, rozwiązywania problemów, wyszukiwania informacji i prezentowania rozwiązań na wiele sposobów.

Podczas realizacji projektu korzystamy z narzędzi informatycznych. Planując zadania, dążymy do tego, aby uczniowie aktywnie i właściwie nauczyli się korzystać z zasobów internetu, posługiwali się multimediami, edytorem tekstu, arkuszem kalkulacyjnym, wspólnie projektowali i tworzyli bazę danych oraz poznali sposoby wykorzystania języka programowania do rozwiązywania problemów algorytmicznych.

Podczas takiej pracy nauczyciel organizuje pracę uczniów i czuwa nad całością projektu. Doradza, wspomaga uczniów, dokonuje krótkiego wprowadzenia, wskazuje źródła informacji, jasno precyzuje problem, rozdziela zadania dla członków grupy, stawia wymagania. Uczniowie zaś muszą sami aktywnie poszukiwać informacji, gromadzić i selekcjonować dane oraz dobierać odpowiednie narzędzia pozwalające na rozwiązanie problemu.

Lekcje informatyki odbywają się zawsze w pracowni komputerowej. Materiał nauczania jest tak zróżnicowany, że niezbędne jest stosowanie różnych metod nauczania. Najczęściej stosujemy następujące metody:

- podające i oglądowe — stosowane podczas omawiania nowych pojęć i wprowadzania nowych zagadnień;
- poszukujące — wykorzystywane podczas rozwiązywania zagadnień, odkrywania i zdobywania nowej wiedzy;
- eksponujące i ćwiczenia praktyczne — stosowane na lekcjach informatyki podczas rozwiązywania problemów z wykorzystaniem odpowiedniego oprogramowania, korzystania z różnych źródeł informacji i prezentowania efektów pracy.

Metody podające i oglądowe należy traktować jako pomocnicze w procesie nauczania. Stosowane są one jako wprowadzenie do kolejnych zagadnień przez udostępnienie gotowej wiedzy.

Metody eksponujące znajdują zastosowanie na zajęciach, podczas których za pomocą komputera wykonuje się pokaz czynności, programów, zjawisk, przebiegu procesów.

Założeniem metod problemowych jest znajdowanie rozwiązań w nowych sytuacjach. Na lekcjach informatyki może to być na przykład poszukiwanie algorytmu postępowania, rozwiązywanie problemów zaczerpniętych z życia codziennego z wykorzystaniem arkusza kalkulacyjnego.

Podczas pracy ze zdolnymi uczniami można wykorzystać metodę nauczania programowego, która polega na zwiększaniu stopnia trudności stawianego zadania przez zmniejszanie liczby wskazówek naprowadzających.

Na zajęciach informatyki warto też zastosować metodę debaty lub burzy mózgów. Uczniowie dyskutują, podają argumenty „za" i „przeciw". Taka forma zajęć stanowi jedynie wstęp do dalszej części lekcji lub jej podsumowanie. Zagadnieniami, które można podjąć na tego typu zajęciach, są:

- zalety i wady grafiki bitmapowej i wektorowej,
- zagrożenia i korzyści wynikające ze stosowania komputerów,
- różne aspekty powszechnego dostępu do informacji,
- bezpieczeństwo w sieci,
- różne aspekty prawa autorskiego.

Dobór metod nauczania zależy od wielu czynników. Należy tu wymienić choćby specyfikę konkretnych lekcji, dotychczasowy poziom wiedzy uczniów, dostępne środki dydaktyczne czy indywidualne predyspozycje nauczyciela.

Zachęcamy uczniów do udziału w różnych konkursach szkolnych, międzyszkolnych i światowych oraz w ogólnopolskiej olimpiadzie informatycznej.

5.3. Osiąganie celów wychowawczych

Nauczyciel musi być nie tylko ciągle doskonalącym się źródłem wiedzy, ale również kreatorem postaw i zachowań. Uczniowie powinni czuć się w pracowni komputerowej współodpowiedzialnymi gospodarzami, którym w miarę możliwości należy przydzielić większe prawa użytkowników sieci lokalnych aniżeli uczniom klas nieinformatycznych. Zostaną dzięki temu obdarzeni nie tylko większym zaufaniem, ale i odpowiedzialnością. Jest to więc świadome działanie wychowawcze.

Aby osiągnąć założone w programie cele wychowawcze, nauczyciel powinien:

- dbać o przestrzeganie nakazów prawnych i norm etycznych w korzystaniu z technologii informacyjnej w pracowni szkolnej;
- zapewnić i chronić bezpieczeństwo danych;
- wykorzystywać pełne możliwości narzędzi sieciowych do poszukiwania, gromadzenia i przetwarzania informacji, komunikacji elektronicznej, tworzenia stron i serwisów internetowych, prezentacji multimedialnych oraz udostępniania ich w sieci z poszanowaniem własności intelektualnej i prawa autorskiego;
- korzystać z licencjonowanego oprogramowania;
- w miarę możliwości tworzyć własne pomoce z wykorzystaniem narzędzi informatycznych, na przykład języka programowania;
- zarządzać realizacją zespołowych projektów programistycznych;
- wpajać uczniom zasadę weryfikowania informacji pochodzących z zasobów globalnej sieci oraz bezwzględnego przestrzegania prawa autorskiego;
- pobudzać w uczniach ciekawość poznawania nowych technologii;
- motywować uczniów do samodzielnego rozwoju;
- przeciwdziałać zagrożeniom związanym z rozwojem technologii informacyjno-komunikacyjnych;
- dbać o szkolną pracownię komputerową i stwarzać z uczniami atmosferę współodpowiedzialności za powierzony sprzęt.

6. Zadania szkoły w zakresie edukacji informatycznej

Zgodnie z obowiązującą podstawą programową kształcenia ogólnego w zakresie informatyki na szkoły zostały nałożone określone zadania. Obejmują one następujące sfery działań:

- Stworzenie warunków umożliwiających realizację edukacji informatycznej oraz poznawanie określonych zagadnień i metod związanych z informatyką jako dyscypliną naukową.
- Pobudzanie rozwoju i indywidualnej pracy ucznia z zastosowaniem narzędzi i metod informatycznych. Praca ta powinna wpływać na kształtowanie samodzielności intelektualnej, świadomości i odpowiedzialności za własny rozwój i podejmowane decyzje.

Należy tu także uwzględnić rozwijanie umiejętności rozwiązywania zadań o charakterze złożonym, wymagających własnej inicjatywy, analizy i oceny wybranego rozwiązania.

- Pobudzanie rozwoju umiejętności pracy w zespole, świadomego i przemyślanego podziału zadań oraz odpowiedzialności za realizowany fragment danego przedsięwzięcia przez wykonywanie projektów grupowych, na przykład o charakterze programistycznym.

6.1. Baza sprzętowa — wyposażenie szkolnej pracowni komputerowej

Pracownia komputerowa, w której odbywają się zajęcia z informatyki, musi być tak zorganizowana i zawierać taki sprzęt komputerowy, aby możliwa była realizacja zajęć z wykorzystaniem oprogramowania obowiązującego na egzaminie maturalnym. W miarę upływu czasu, zgodnie ze zmianami oprogramowania egzaminacyjnego, sprzęt komputerowy powinien być aktualizowany odpowiednio do wymagań programowych.

Ponadto komputery w pracowni, w której prowadzone są lekcje, powinny być połączone siecią lokalną z dostępem do internetu. Dodatkowym wyposażeniem pracowni powinny być urządzenia peryferyjne, takie jak drukarka udostępniona sieciowo, skaner, kamery internetowe, cyfrowy aparat fotograficzny, słuchawki i mikrofony przy każdym stanowisku. Ogromną pomoc dla nauczyciela prowadzącego zajęcia stanowi projektor multimedialny lub tablica interaktywna.

6.2. Oprogramowanie

Komputery znajdujące się w pracowni komputerowej, w której odbywają się zajęcia z informatyki, powinny być wyposażone w oprogramowanie zgodne z listą programów do wyboru dla zdającego egzamin maturalny z informatyki. Uczniowie przygotowujący się do egzaminu muszą pracować z wykorzystaniem programów, których będą mogli używać podczas egzaminu. Ma to bezpośredni wpływ na wyniki zdawanego egzaminu oraz ułatwia pracę nauczycielowi przygotowującemu zdających. W miarę upływu czasu należy aktualizować oprogramowanie, jednocześnie obserwując zmiany programów obowiązujących na egzaminie maturalnym z informatyki. Aktualna lista środowisk, języków programowania (kompilatorów) i programów użytkowych, z której mogą wybierać zdający egzamin maturalny z informatyki, jest ogłaszana na stronie Centralnej Komisji Egzaminacyjnej w Warszawie: *www.cke.edu.pl*.

W skład oprogramowania niezbędnego do prowadzenia zajęć z informatyki powinny wchodzić następujące programy:

- system operacyjny,
- edytor tekstu,
- arkusz kalkulacyjny,
- program do tworzenia prezentacji multimedialnych,
- program do tworzenia baz danych i zarządzania nimi,
- edytory grafiki wektorowej i rastrowej,
- edytor plików dźwiękowych,
- program do montażu filmu,
- edytor stron WWW,
- narzędzia komunikacyjne sieci lokalnej i rozległej,
- kompilator języka wysokiego poziomu,
- program archiwizujący,
- przeglądarka internetowa,
- oprogramowanie antywirusowe.

Powyższa lista nie obejmuje wszystkich programów — można je uzupełnić o inne programy przydatne podczas realizacji materiału nauczania, na przykład o oprogramowanie niezbędne do uruchomienia skryptów PHP, witryn internetowych współpracujących z bazą danych MySQL czy oprogramowanie potrzebne do uruchomienia systemów operacyjnych i aplikacji na maszynie wirtualnej.

7. Ramowy plan nauczania

Ramowy plan nauczania dokładnie określa, ile godzin lekcyjnych mamy przeznaczyć na realizację danego materiału. Minimalna liczba godzin, jaką musimy zrealizować w trzyletnim cyklu, wynosi 180. Przy jednej godzinie tygodniowo w danym roku szkolnym mamy do dyspozycji od 35 do 38 godzin lekcyjnych. Średnio przyjmujemy, że zostanie zrealizowanych minimalnie 30 godzin. Na realizację programu przeznaczamy więc 6 godzin w cyklu kształcenia. Jeśli jednak tych godzin pojawi się więcej, należy je przeznaczyć na inne działania nauczyciela. Te dodatkowe godziny mogą być wykorzystane na sprawdzanie wiadomości i umiejętności, powtarzanie zrealizowanego materiału, wprowadzanie nowości technicznych dotyczących sprzętu komputerowego czy nowego oprogramowania, przedłużenie czasu przeznaczonego na realizację projektów czy pisanie programów.

W klasach, w których jest realizowana informatyka w zakresie rozszerzonym, prowadzone są również zajęcia z informatyki na poziomie podstawowym. Stanowią one bazę dla zakresu rozszerzonego, dlatego przedmiot ten pojawia się w klasie pierwszej i jest realizowany we wszystkich klasach. W klasie drugiej i trzeciej przewidziana jest informatyka na poziomie rozszerzonym tylko w klasach, w których wybrano zakres rozszerzony dla tego przedmiotu. Podział materiału nauczania informatyki jest uzależniony od liczby godzin tego przedmiotu w cyklu nauczania danej klasy. Minimalny wymiar godzin obowiązkowych zajęć edukacyjnych wynosi 180. W tabeli 7.1 przedstawiono propozycję **podziału materiału nauczania**.

Tabela 7.1. *Podział materiału nauczania*

LP.	DZIAŁ	PRZYDZIAŁ GODZIN DLA ZAKRESU ROZSZERZONEGO
1.	Algorytmika i programowanie	90
2.	Bazy danych	50
3.	Multimedia i grafika komputerowa	16
4.	Systemy operacyjne i sieci komputerowe	20
5.	Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki	4

Propozycja rozkładu godzinowego dla minimalnego wymiaru godzin obowiązkowych zajęć edukacyjnych, czyli dla 180 godzin:

kl. II — 120 godzin lekcyjnych,
kl. III — 60 godzin lekcyjnych.

8. Treści nauczania i przewidywane osiągnięcia uczniów

Przedstawione w tym rozdziale treści nauczania i przewidywane osiągnięcia uczniów odpowiadają realizacji podstawy programowej na poziomie rozszerzonym. Stanowią one szczegółową informację — zarówno dla nauczyciela, jak i dla ucznia — dotyczącą zakresu wiadomości i umiejętności obowiązujących na egzaminie maturalnym z informatyki.

Treści nauczania i szczegółowe umiejętności ucznia podzielono na działy i zamieszczono w tabelach.

8.1. Algorytmika

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
WPROWADZENIE DO ALGORYTMIKI	
Wprowadzenie do algorytmów. Pojęcie algorytmu.	Rozumie pojęcie: algorytm. Formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego. Potrafi podać przykładowe algorytmy związane z życiem codziennym, innymi nauczanymi przedmiotami itp.
Etapy rozwiązywania zadań za pomocą komputera. Specyfikacja zadania: dane wejściowe i dane wyjściowe. Zastosowanie specyfikacji zadania w realizacji algorytmów.	Poprawnie definiuje problem i formułuje jego specyfikację. Stosuje podejście algorytmiczne do rozwiązywanego problemu. Zna i realizuje etapy rozwiązywania zadań za pomocą komputera (od sformułowania specyfikacji problemu po testowanie rozwiązania). Rozwiązuje problem zgodnie z etapami: • zaplanowanie działań, • wydzielenie podproblemów i określenie zależności między nimi, • skonstruowanie algorytmów rozwiązujących podproblemy,

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
WPROWADZENIE DO ALGORYTMIKI	
	• analiza tych algorytmów i określenie ogólnego algorytmu rozwiązującego problem. Dokonuje analizy prostego i umiarkowanie złożonego zadania oraz opracowuje algorytm zgodny ze specyfikacją. Ocenia zgodność algorytmu ze specyfikacją problemu.
Sposoby reprezentowania algorytmów: opis słowny, lista kroków, schemat blokowy, drzewo algorytmu, program.	Zna różne sposoby reprezentowania algorytmów, w tym opis słowny, listę kroków, schemat blokowy, drzewo algorytmu, program.
Realizacja wybranych algorytmów przy użyciu różnych sposobów ich reprezentacji: • schemat blokowy — graficzna prezentacja algorytmu, • lista kroków — słowna prezentacja algorytmu, • program w wybranym języku programowania wysokiego poziomu — praktyczna realizacja algorytmu, • arkusz kalkulacyjny.	Zapisuje algorytmy, stosując różne sposoby ich reprezentowania, w tym schemat blokowy, listę kroków, program w języku programowania wysokiego poziomu, arkusz kalkulacyjny. Wyodrębnia elementy składowe algorytmu. Dobiera właściwy sposób rozwiązania i prezentacji algorytmu do konkretnego problemu. Zapisuje rozwiązanie zadania w postaci algorytmu ze specyfikacją w wybranej przez siebie notacji. Wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i zapisywania algorytmów. Formułuje informatyczne rozwiązanie problemu (realizację algorytmu), dobiera odpowiednie struktury danych (w tym struktury dynamiczne) i możliwie najlepszy algorytm oraz zapisuje go w wybranym języku programowania. Stosuje metodę zstępującą i wstępującą przy rozwiązywaniu problemu. Wykorzystuje różne metody algorytmiczne i algorytmy klasyczne do rozwiązywania problemu oraz świadomie wybiera właściwy sposób rozwiązania zadania.

	Wykorzystuje zdobytą wiedzę i umiejętności do rozwiązywania prostych i umiarkowanie złożonych zadań z różnych dziedzin.
Algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami). Realizacja wybranych algorytmów. Rozwiązywanie równań: liniowego i kwadratowego.	Rozumie pojęcia: algorytm liniowy, algorytm z warunkami (rozgałęzieniami). Stosuje algorytmy liniowe i algorytmy z warunkami (rozgałęzieniami) przy rozwiązywaniu problemów. Identyfikuje wykorzystaną metodę programowania liniowego i z warunkami w przykładowych algorytmach. Zapisuje i stosuje algorytmy liniowe i warunkowe: rozwiązywanie równania liniowego, rozwiązywanie równania kwadratowego, stabilny algorytm rozwiązujący równanie kwadratowe itp.
Iteracja. Definiowanie iteracji. Realizacja algorytmów z zastosowaniem iteracji: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, algorytmy wykonywane na tekstach (łańcuchach) i tablicach.	Rozumie pojęcia: iteracja, wzór iteracyjny. Potrafi zdefiniować iterację. Stosuje metodę iteracji przy realizacji algorytmów. Identyfikuje wykorzystaną metodę iteracji w przykładowych algorytmach. Stosuje iterację do zapisu algorytmów: wyznaczanie elementów, obliczanie sumy i iloczynu oraz inne działania na ciągach liczbowych, proste i złożone algorytmy wykonywane na tekstach (łańcuchach) oraz tablicach (na przykład znajdowanie podciągów o określonych własnościach) itp.
Rekurencja. Definiowanie rekurencji. Procedury-funkcje rekurencyjne. Zamiana iteracji na rekurencję. Realizacja algorytmów z zastosowaniem rekurencji: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi. Analiza wartości zmiennych w przykładowych algorytmach rekurencyjnych.	Rozumie pojęcia: rekurencja, wzór rekurencyjny, zależność rekurencyjna. Potrafi podać przykłady zastosowania rekurencji oraz powiązać sytuacje życiowe z rozwiązaniami rekurencyjnymi. Potrafi definiować zależności rekurencyjne oraz odpowiednie procedury-funkcje rekurencyjne. Potrafi dokonywać zamiany metody iteracyjnej wykorzystanej w algorytmie na rekurencyjną. Identyfikuje wykorzystaną metodę rekurencji w przykładowych algorytmach.

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
WPROWADZENIE DO ALGORYTMIKI	
	Zapisuje i stosuje algorytmy rekurencyjne: obliczanie silni liczby naturalnej, wyznaczanie elementów ciągu Fibonacciego, wieże Hanoi itp. Potrafi zastosować iterację do realizacji tych algorytmów. Dokonuje analizy wartości zmiennych w przykładowych algorytmach rekurencyjnych. Stosuje rekurencję w prostych sytuacjach problemowych.
Metoda „dziel i zwyciężaj".	Rozumie pojęcie: metoda „dziel i zwyciężaj". Potrafi definiować i stosować metodę „dziel i zwyciężaj" w odpowiednich sytuacjach. Identyfikuje wykorzystaną metodę „dziel i zwyciężaj" w przykładowych algorytmach. Zapisuje i stosuje algorytmy z wykorzystaniem metody „dziel i zwyciężaj".
Programowanie zachłanne.	Rozumie pojęcie: programowanie zachłanne. Identyfikuje wykorzystaną metodę programowania zachłannego w przykładowych algorytmach. Stosuje podejście zachłanne w rozwiązywaniu problemów.
Kryptografia i kryptoanaliza. Metody szyfrowania.	Rozumie pojęcia: kryptografia, kryptoanaliza, algorytmy kryptograficzne. Potrafi wymienić podstawowe metody szyfrowania.
ALGORYTMY I ICH ZASTOSOWANIE	
Algorytmy badające własności geometryczne: • sprawdzanie warunku trójkąta, • określanie prostopadłości i równoległości prostych, • obliczanie odległości punktu od prostej, • obliczanie odległości punktów na płaszczyźnie, • badanie położenia punktu względem prostej, • badanie przynależności punktu do odcinka.	Potrafi definiować prostą w postaci ogólnej i kierunkowej. Na podstawie wzorów wyznacza odległość punktu od prostej oraz odległość punktów na płaszczyźnie. Na podstawie równań kierunkowych określa prostopadłość i równoległość prostych. Zapisuje algorytm sprawdzający warunek trójkąta. Zapisuje i stosuje algorytmy badające położenie punktu względem prostej oraz przynależność punktu do odcinka.
Wyznaczanie największego wspólnego dzielnika i najmniejszej wspólnej wielokrotności dwóch liczb naturalnych: • wyznaczanie największego wspólnego dzielnika — algorytm Euklidesa, • wyznaczanie najmniejszej wspólnej wielokrotności.	Wyznacza największy wspólny dzielnik dwóch liczb naturalnych algorytmem Euklidesa. Stosuje iteracyjną i rekurencyjną realizację algorytmu. Wyznacza najmniejszą wspólną wielokrotność dwóch liczb naturalnych, wykorzystując algorytm obliczający największy wspólny dzielnik.
Wyznaczanie wartości wielomianu, pozycyjne systemy liczbowe i reprezentacja danych liczbowych w komputerze: • wyznaczanie wartości wielomianu schematem Hornera, • szybkie podnoszenie do potęgi, • pozycyjne systemy liczbowe stosowane w informatyce — system dwójkowy (binarny), ósemkowy (oktalny) i szesnastkowy (heksadecymalny) — definicja, zamiana liczb zapisanych w podanych systemach, realizacja podstawowych operacji arytmetycznych w podanych systemach,	Zapisuje i stosuje algorytm obliczający wartość wielomianu za pomocą schematu Hornera. Zapisuje i stosuje algorytm szybkiego podnoszenia do potęgi z wykorzystaniem schematu Hornera. Rozumie pojęcia: system liczbowy, pozycyjny system liczbowy. Zna reprezentację liczb w dowolnym systemie pozycyjnym, w tym dwójkowym (binarnym), ósemkowym (oktalnym) i szesnastkowym (heksadecymalnym), oraz zamienia liczby zapisane w tych systemach liczbowych na system dziesiętny i odwrotnie. Zamienia liczby z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera. Wykonuje podstawowe operacje arytmetyczne w dowolnych pozycyjnych systemach liczbowych. Zna reprezentacje danych liczbowych w komputerze, w tym reprezentację binarną liczb całkowitych i niecałkowitych, stałopozycyjną reprezentację liczb, zmiennopozycyjną reprezentację liczb, pojęcie mantysy i cechy. Wyjaśnia źródło błędów w obliczeniach komputerowych, w tym błąd względny i bezwzględny.

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
ALGORYTMY I ICH ZASTOSOWANIE	
• zamiana liczb z dowolnego pozycyjnego systemu liczbowego na dziesiętny z zastosowaniem schematu Hornera, • zamiana liczb z systemu dziesiętnego na dowolny pozycyjny system liczbowy, • wykonywanie podstawowych działań arytmetycznych w różnych systemach liczbowych, • reprezentacja danych liczbowych w komputerze — reprezentacja binarna liczb całkowitych i niecałkowitych, reprezentacja stałopozycyjna i zmiennopozycyjna liczb, błędy w obliczeniach.	
Liczby pierwsze: • sprawdzanie, czy liczba jest liczbą pierwszą, • rozkładanie liczby na czynniki pierwsze, • sito Eratostenesa.	Rozumie pojęcie: liczba pierwsza. Zapisuje i stosuje algorytm sprawdzający, czy dana liczba jest pierwsza, dokonuje rozkładu liczby na czynniki pierwsze. Zapisuje i stosuje algorytm generujący liczby pierwsze — sito Eratostenesa.
Przeszukiwanie ciągu liczbowego: • przeszukiwanie liniowe tablicy jednowymiarowej, • przeszukiwanie liniowe tablicy jednowymiarowej z wartownikiem.	Wykorzystuje metody liniowe przy przeszukiwaniu ciągu liczbowego. Zapisuje i stosuje algorytm przeszukiwania liniowego tablicy jednowymiarowej. Zapisuje i stosuje algorytm przeszukiwania liniowego tablicy jednowymiarowej z wartownikiem.

Znajdowanie największego lub najmniejszego elementu w ciągu liczbowym: • znajdowanie największego elementu w ciągu liczbowym, • znajdowanie najmniejszego elementu w ciągu liczbowym.	Zapisuje i stosuje algorytm znajdowania największego elementu w ciągu liczbowym. Zapisuje i stosuje algorytm znajdowania najmniejszego elementu w ciągu liczbowym.
Znajdowanie lidera w zbiorze.	Rozumie pojęcie: lider w zbiorze. Zapisuje i stosuje algorytm znajdowania lidera w zbiorze.
Sprawdzanie monotoniczności ciągu liczbowego: • sprawdzanie, czy ciąg liczbowy jest rosnący lub malejący, • sprawdzanie, czy ciąg liczbowy jest nierosnący lub niemalejący, • sprawdzanie, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.	Rozumie pojęcie: monotoniczność ciągu liczbowego. Zapisuje i stosuje algorytm sprawdzający, czy ciąg liczbowy jest rosnący lub malejący. Zapisuje i stosuje algorytm sprawdzający, czy ciąg liczbowy jest nierosnący lub niemalejący. Zapisuje i stosuje algorytm sprawdzający, czy ciąg liczbowy jest monotoniczny lub niemonotoniczny.
Liniowe sortowanie ciągu liczbowego: • porządkowanie przez wybór, • porządkowanie przez wstawianie, • porządkowanie bąbelkowe.	Zapisuje i stosuje algorytm porządkowania przez wybór. Zapisuje i stosuje algorytm porządkowania przez wstawianie. Zapisuje i stosuje algorytm porządkowania bąbelkowego.
Zastosowanie metody „dziel i zwyciężaj": • jednoczesne znajdowanie najmniejszego i największego elementu, • przeszukiwanie binarne ciągu uporządkowanego,	Wykorzystuje metodę „dziel i zwyciężaj" przy realizacji algorytmów. Zapisuje i stosuje algorytm jednoczesnego znajdowania najmniejszego i największego elementu. Zapisuje i stosuje algorytm przeszukiwania binarnego uporządkowanego ciągu liczbowego.

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
ALGORYTMY I ICH ZASTOSOWANIE	
• sortowanie przez scalanie, • sortowanie szybkie, • znajdowanie przybliżonej wartości miejsca zerowego funkcji ciągłej — metoda połowienia przedziałów.	Zapisuje i stosuje algorytm sortowania przez scalanie. Zapisuje i stosuje algorytm sortowania szybkiego. Zapisuje i stosuje algorytm znajdowania przybliżonej wartości miejsca zerowego funkcji ciągłej — metodą połowienia przedziałów.
Obliczenia przybliżone. Realizacja wybranych algorytmów numerycznych: • obliczanie wartości pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona), • obliczanie pola obszaru ograniczonego wykresem funkcji (całkowanie numeryczne), • **przybliżanie wartości liczby metodą Monte Carlo.**	Rozumie pojęcia: obliczenia przybliżone, metody numeryczne. Zapisuje i stosuje algorytm obliczający wartość pierwiastka kwadratowego z liczby dodatniej — algorytm Newtona-Raphsona (metoda Herona). Zapisuje i stosuje algorytm obliczający pole obszaru ograniczonego wykresem funkcji — metoda prostokątów, metoda trapezów. **Zapisuje i stosuje algorytm wykonujący przybliżanie wartości liczby metodą Monte Carlo.**
Algorytmy na tekstach (łańcuchach): • sprawdzanie, czy tekst jest palindromem, • porządkowanie alfabetyczne tekstu, • sprawdzanie, czy teksty są anagramami, • wyszukiwanie wzorca w tekście, • wyznaczanie wartości wyrażenia zapisanego w odwrotnej notacji polskiej ONP.	Zapisuje i stosuje algorytmy wykonywane na tekstach, na przykład sprawdzanie, czy tekst jest palindromem, porządkowanie alfabetyczne tekstu, sprawdzanie, czy teksty są anagramami, wyszukiwanie wzorca w tekście. Zna zasady notacji wyrażenia podanego w postaci ONP. Zapisuje i stosuje algorytm obliczający wartość wyrażenia zapisanego w postaci ONP.

Zastosowanie programowania zachłannego: • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana.	Stosuje metodę zachłanną do zapisu algorytmów: • algorytm zachłanny dla problemu plecakowego, • algorytm wydawania reszty, • algorytm Huffmana.
Wybrane algorytmy kryptograficzne.	Zapisuje i stosuje przykładowe algorytmy kryptograficzne, w tym algorytmy symetryczne (np. szyfr Cezara, szyfr przestawieniowy, szyfr płotowy, szyfry monoalfabetyczne i wieloalfabetyczne), algorytmy asymetryczne (np. szyfr z kluczem jawnym RSA), podpis cyfrowy itp.
WŁASNOŚCI ALGORYTMÓW	
Złożoność czasowa algorytmów. Analiza złożoności czasowej wybranych algorytmów.	Rozumie pojęcia: złożoność obliczeniowa, złożoność czasowa. Potrafi wyznaczać, oceniać oraz uzasadniać złożoność czasową algorytmów. Analizuje złożoność czasową wybranych algorytmów (oblicza liczbę operacji dominujących wykonywanych przez algorytm). Porównuje złożoność czasową algorytmów rozwiązujących ten sam problem. Dobiera możliwie najszybszy algorytm rozwiązujący postawiony problem.
Złożoność pamięciowa algorytmów. Analiza złożoności pamięciowej wybranych algorytmów.	Rozumie pojęcie: złożoność pamięciowa. Potrafi wyznaczać, oceniać oraz uzasadniać złożoność pamięciową algorytmów. Analizuje złożoność pamięciową wybranych algorytmów (szacuje wielkość pamięci potrzebnej do komputerowej realizacji algorytmu). Porównuje złożoność pamięciową algorytmów rozwiązujących ten sam problem. Dobiera odpowiednie struktury danych w rozwiązaniu postawionego problemu.

Tabela 8.1. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie algorytmiki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
WŁASNOŚCI ALGORYTMÓW	
Własności algorytmów: efektywność, poprawność, skończoność, optymalność. Analiza własności wybranych algorytmów.	Rozumie pojęcia: własności algorytmów, efektywność algorytmu, poprawność algorytmu, skończoność algorytmu, optymalność algorytmu. Opisuje własności algorytmów na podstawie ich analizy. Rozwiązując problem, wybiera algorytm najkorzystniejszy ze względu na jego własności. Bada efektywność komputerowych rozwiązań problemów. Ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

8.2. Programowanie

Tabela 8.2. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
Języki programowania — pojęcie, klasyfikacja, przykłady. Prezentacja przykładowych programów w różnych językach programowania. Podstawowe zasady i metody programowania. Znajomość pojęć: translator, kompilator, interpreter, debugger, linker.	Rozumie pojęcia: język programowania, translator, kompilator, interpreter, debugger, linker. Zna klasyfikacje języków programowania: imperatywne i deklaratywne, niskiego i wysokiego poziomu. Potrafi wymienić i sklasyfikować podstawowe języki programowania. Kompiluje i uruchamia przykładowe programy napisane w różnych językach programowania. Zna i rozumie podstawowe zasady i metody programowania.

Elementy zintegrowanego systemu programowania, w tym umiejętność pisania, kompilowania i uruchamiania programów za pomocą wybranego kompilatora.	Korzysta z wybranego środowiska programistycznego (na przykład kompilatora), w którym zapisuje, kompiluje, uruchamia i testuje programy.
Wprowadzenie do programowania: • struktura programu, • operacje wejścia-wyjścia, • zmienne i ich deklaracja, • stałe i ich deklaracja, • wyrażenia arytmetyczne, relacje i operatory logiczne, • priorytety relacji i działań, • komentarze.	Zna podstawową strukturę programu. Korzysta w programach z podstawowych operacji wejścia i wyjścia. Potrafi deklarować zmienne i wykorzystywać je w programach. Potrafi deklarować stałe i wykorzystywać je w programach. Zna podstawowe wyrażenia arytmetyczne, relacje i operatory logiczne oraz stosuje je w programach. Zna priorytety relacji i działań charakterystyczne dla danego języka programowania oraz uwzględnia je przy pisaniu programów. Stosuje komentarze przy pisaniu programów.
Podstawowe konstrukcje algorytmiczne: • instrukcja przypisania, • instrukcja złożona, • instrukcje warunkowe, • instrukcja wyboru, • instrukcje iteracyjne. Umiejętność korzystania w implementacjach z podstawowych konstrukcji algorytmicznych.	Stosuje podstawowe konstrukcje algorytmiczne, w tym instrukcję przypisania, instrukcję złożoną, instrukcje warunkowe, instrukcję wyboru, instrukcje iteracyjne. Stosuje w programach powyższe konstrukcje algorytmiczne.

Tabela 8.2. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
Proste typy danych: • typy całkowite, • typy rzeczywiste, • typ znakowy, • typ logiczny. Realizacja operacji z wykorzystaniem prostych typów danych oraz ich zastosowanie w programach.	Rozumie pojęcie: proste typy danych. Zna proste typy danych, w tym typy całkowite, typy rzeczywiste, typ znakowy, typ logiczny. Wykonuje operacje na prostych typach danych. Korzysta w programach z prostych typów danych. Potrafi w danej sytuacji problemowej wybrać właściwy typ danych dla wykorzystywanych zmiennych.
Strukturalizacja programu: • definiowanie procedur-funkcji, • zmienne lokalne i globalne, • parametry formalne i aktualne, • przekazywanie parametrów w procedurach-funkcjach, • **przeciążanie funkcji**. Zastosowanie w implementacjach procedur-funkcji z parametrami. Umiejętność wyboru właściwego w danej sytuacji sposobu przekazywania parametrów.	Rozumie podstawowe zasady i cel strukturalizacji programu. Potrafi wydzielać fragmenty programu i definiować procedury-funkcje. Rozumie różnice między zmiennymi lokalnymi i globalnymi oraz świadomie korzysta z wybranego typu zmiennych w programach. Rozumie różnice między parametrami formalnymi i aktualnymi. Zna sposoby przekazywania parametrów w procedurach-funkcjach i świadomie stosuje je podczas realizacji programów. **Zna podstawowe sposoby przeciążania funkcji i stosuje je w programach.**
Strukturalne typy danych: • łańcuchy (w tym operacje na tekście), • tablice (w tym jednowymiarowe i wielowymiarowe), • rekordy-struktury (w tym tablice rekordów-struktur), • struktury dynamiczne (w tym listy, stosy, kolejki, drzewa binarne).	Rozumie pojęcia: strukturalne typy danych, abstrakcyjne typy danych. Potrafi definiować typ łańcuchowy. Zna predefiniowane operacje na tekście i z nich korzysta. Zapisuje programy z wykorzystaniem typu łańcuchowego. Potrafi definiować typ tablicowy, w tym tablice jedno- i wielowymiarowe. Zapisuje programy z wykorzystaniem typu tablicowego.

Realizacja programów z wykorzystaniem strukturalnych typów danych.	Potrafi definiować typ rekordowy-strukturalny. Potrafi definiować tablice rekordów-struktur. Zapisuje programy z wykorzystaniem typu rekordowego-strukturalnego, w tym tablic rekordów-struktur. Potrafi definiować dynamiczne struktury danych, w tym listy, stosy, kolejki, drzewa binarne. Zapisuje programy z wykorzystaniem dynamicznych struktur danych. Dobiera struktury danych (w tym dynamiczne struktury danych) odpowiednio do przetwarzanych informacji. Wykorzystuje typy strukturalne, w tym typ łańcuchowy, tablicowy i rekordowy-strukturalny, oraz dynamiczne struktury danych do implementacji algorytmów.
Plikowe operacje wejścia-wyjścia: • rodzaje plików, • definicja i etapy przetwarzania plików, • podstawowe operacje na plikach. Realizacja programów z zastosowaniem operacji plikowych.	Zna i potrafi definiować różne typy plików, w tym tekstowe, binarne. Zna i realizuje etapy przetwarzania plików. Zna podstawowe operacje na plikach i z nich korzysta. Zapisuje programy z wykorzystaniem typu plikowego, w tym importuje dane z plików zewnętrznych, eksportuje wyniki do plików zewnętrznych. Potrafi skonstruować prostą bazę danych z zastosowaniem operacji na plikach.
Programowanie zorientowane obiektowo: **• klasy i obiekty,** **• dziedziczenie i hierarchia klas,** **• konstruktory i destruktory,** **• polimorfizm,** **• metody wirtualne.** **Realizacja implementacji z zastosowaniem programowania zorientowanego obiektowo.**	**Rozumie pojęcia: programowanie obiektowe, klasa, obiekt, enkapsulacja, dziedziczenie, hierarchia klas, polimorfizm, konstruktor, destruktor.** **Potrafi definiować typ obiektowy.** **Tworzy proste klasy i statyczne obiekty.** **Stosuje metody konstruktora i destruktora.** **Projektuje i realizuje hierarchię klas.** **Potrafi definiować i stosować w programach dynamiczne obiekty.** **Stosuje metody wirtualne.** **Realizuje programy z wykorzystaniem typu obiektowego.**

Tabela 8.2. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie programowania (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
Projekt programistyczny.	Zna pojęcie: inżynieria oprogramowania. Potrafi określić kolejne fazy konstruowania oprogramowania. Znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin. Realizuje indywidualnie lub zespołowo projekt programistyczny, w tym określa etapy pracy i dzieli zadanie na moduły, wykonuje określone części zadania, łączy poszczególne moduły w całość, sporządza dokumentację projektu.

8.3. Bazy danych

Tabela 8.3. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
BAZODANOWE FUNKCJE ARKUSZA KALKULACYJNEGO	
Podstawowe zasady tworzenia tabeli stanowiącej bazę danych.	Zna i rozumie pojęcie: baza danych. Zna i stosuje reguły, jakim podlega tabela stanowiąca źródło danych. Potrafi zgromadzić w pliku tekstowym dane będące bazą danych. Importuje dane umieszczone w pliku tekstowym do tabeli arkusza kalkulacyjnego. Zapisuje zgromadzone w tabeli dane w pliku tekstowym, stosując odpowiednie znaki separacji.
Wyszukiwanie informacji w tabeli przy użyciu autofiltru oraz filtru zaawansowanego.	Korzysta z autofiltru w celu wyselekcjonowania danych. Potrafi stosować złożone kryteria wyboru w filtrach zaawansowanych. Stosuje odpowiedni rodzaj filtru w celu wybrania informacji potrzebnych w danej sytuacji.

Tworzenie podsumowań danych.	Potrafi zastosować funkcje standardowe arkusza do podsumowań danych. Zna pojęcie: suma pośrednia. Wie, jak i do czego używa się sum pośrednich. Potrafi dokonać podsumowań danych zawartych w tabeli przy użyciu kreatora sum częściowych. Potrafi dokonać korekty i zamiany kryteriów podsumowań w istniejącym zestawieniu sum pośrednich. Rozumie pojęcie: tabela przestawna. Wie, do czego służą tabele przestawne. Wskazuje, do jakich danych w celu ich podsumowania można zastosować tabele przestawne. Potrafi wykonać zestawienie podsumowań danych przy użyciu kreatora tabel przestawnych. Potrafi zmienić ustawienia w istniejącej tabeli przestawnej.
Graficzna prezentacja danych na wykresach.	Potrafi dobrać odpowiedni typ wykresu do prezentowanych na nim danych. Umie tworzyć różnego typu wykresy do danych zawartych w tabelach i tabelach przestawnych. Potrafi ustalić odpowiednie opcje wykresu i formatować poszczególne jego elementy. Potrafi zaprezentować graficznie rozwiązania zadań i problemów z innych dziedzin nauki szkolnej (równanie i układ równań, wyniki pomiarów fizycznych lub chemicznych, analiza rynku pracy).
RELACYJNA BAZA DANYCH	
Podstawy relacyjnej bazy danych.	Zna różne typy baz danych — proste i złożone. Zna i rozumie pojęcia: system zarządzania bazą danych, relacyjna baza danych. Zna zasady projektowania bazy danych. Rozumie potrzebę unikania nadmiarowości danych. Rozumie, czym są pola i rekordy oraz jaką rolę pełni klucz w bazie danych. Na przykładzie istniejącej relacyjnej bazy danych wymienia obiekty związane z takimi bazami (tabele, kwerendy, formularze, raporty, makra).

Tabela 8.3. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
RELACYJNA BAZA DANYCH	
Tabele w relacyjnej bazie danych i relacje między nimi.	Potrafi zaimportować tabele z istniejącej bazy danych, arkusza kalkulacyjnego czy pliku tekstowego. Potrafi zaprojektować układ tabel, unikając powtarzania danych. Tworzy tabele przy użyciu kreatora tabel oraz w widoku projektu. Potrafi ustalić właściwości pól. Zna i poprawnie stosuje typy danych w tabelach. Ustala podstawowe właściwości pól tabeli, takie jak rozmiar pola, maska wprowadzania czy reguła sprawdzania poprawności. Rozumie, jaką rolę w bazie danych pełni indeks. Zna pojęcia: klucz główny oraz klucz obcy. Poprawnie definiuje związki między tabelami. Rozróżnia podstawowe typy relacji między tabelami, potrafi je nazwać. Kontroluje i zmienia warunki wymuszania więzów integralności podczas edytowania właściwości określonej relacji. Potrafi tworzyć i modyfikować tabele, uwzględniając różne typy danych oraz ustalone dla nich właściwości pól (rozmiar, ograniczenia, wymagania, komunikaty dla użytkownika itp.). Planuje współpracę i zespołowe wykonanie projektu relacyjnej bazy danych.
Wprowadzanie, wyszukiwanie, edycja i usuwanie danych bezpośrednio w tabelach i poprzez formularze.	Zna różne formaty danych. Wykonuje czynności edycyjne na tabelach bazy danych. Wie, jak wprowadzić dane bezpośrednio do tabeli lub poprzez formularz. Wie, jak korzystać z formularzy do wprowadzania, wyszukiwania i edycji danych. Potrafi porządkować tabele i przeglądać wybrane rekordy przez sortowanie jedno- i wieloparametrowe.

	Zmienia sposób prezentowania danych poprzez ich sortowanie lub filtrowanie. Potrafi drukować tabele i formularze. Tworzy i modyfikuje formularze oraz formanty za pomocą kreatora i w widoku projektu. Dodaje etykiety, formanty, nagłówek lub stopkę do formularza, zmienia jego szatę graficzną.
Wyszukiwanie informacji w relacyjnej bazie danych z użyciem kwerend.	Zna podstawowe typy kwerend. Samodzielnie projektuje zapytania, korzystając z widoku projektu oraz kreatorów różnego typu kwerend wybierających dane z jednej, kilku tabel lub z wcześniej zdefiniowanych kwerend. Stosuje w zapytaniach znaki wieloznaczne (zastępcze). Konstruuje pola obliczeniowe, wprowadza korekty. Określa złożone kryteria liczbowe i definiuje wyrażenia. Stosuje odpowiednie kryteria i parametry do wyszukiwania danych. Wprowadza wiele kryteriów wyboru w jednym polu, stosując odpowiednie operatory. Tworzy kwerendy obliczające podsumowania. Tworzy zestawienia krzyżowe, odpowiednio definiując kryteria, nagłówki wierszy i kolumn. Tworzy kwerendy funkcjonalne generujące tabele, aktualizujące dane, usuwające i dołączające dane. Zmienia właściwości sprzężenia między tabelami, na podstawie których tworzona jest kwerenda. Analizuje kod zapisany w języku zapytań SQL. Modyfikując kryteria zapytania, wprowadza zmiany w kodzie języka zapytań SQL. Zna podstawy języka zapytań SQL pozwalające na tworzenie i modyfikowanie kwerend wybierających, parametrycznych, krzyżowych i modyfikujących tabele.
Przygotowanie zestawień wybranych danych w raportach.	Tworzy proste raporty przy użyciu narzędzia Autoraport. Wie, jak korzystać z kreatora raportów do generowania dowolnych rodzajów raportów. Potrafi grupować informacje w raporcie. Wie, kiedy korzystać z podsumowania statystycznego. Przygotowuje raport do druku.

Tabela 8.3. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie baz danych (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
RELACYJNA BAZA DANYCH	
Makropolecenia.	Definiuje makropolecenia złożone z kilku sekwencji akcji. Rozumie znaczenie komunikatu dotyczącego ostrzeżenia o zabezpieczeniach związanych z wyłączeniem określonej zawartości bazy danych.
Ochrona bazy danych.	Rozumie konieczność chronienia zgromadzonych danych przed osobami niepowołanymi do korzystania z nich i ich przetwarzania. Potrafi stosować podstawowe mechanizmy chroniące plik bazy danych przed jego otwarciem i ewentualną przypadkową lub celową modyfikacją. Zna efekt działania szyfrowania (kodowania) bazy danych. Rozumie konieczność wykonywania kopii bezpieczeństwa (na przykład przy wprowadzaniu i testowaniu kwerend funkcjonalnych).

8.4. Multimedia i grafika komputerowa

Tabela 8.4. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimediów i grafiki komputerowej*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
MULTIMEDIA I GRAFIKA KOMPUTEROWA	
Dźwięk w komputerze.	Potrafi wymienić kilka typów plików dźwiękowych oraz ich zastosowanie. Wie, jak powstał cyfrowy zapis dźwięku. Rozumie jego podstawowe parametry: częstotliwość próbkowania, poziomy kwantyzacji. Zna i charakteryzuje metody kompresji plików dźwiękowych. Potrafi przeprowadzić konwersję między różnymi formatami dźwięku. Potrafi nagrać głos i specjalne efekty dźwiękowe.

	Potrafi zastosować filtry korygujące i wzbogacające dźwięk. Potrafi przeprowadzić edycję, miksowanie i montaż dźwięku, wykorzystując materiały własne oraz bazy gotowych efektów dźwiękowych. Posługuje się fachową terminologią. Zna i charakteryzuje sposoby reprezentowania dźwięku w komputerze.
Grafika komputerowa.	Zna podstawowe pojęcia związane z grafiką rastrową i wektorową. Objaśnia podstawowe różnice pomiędzy grafiką rastrową a wektorową. Zna wady i zalety grafiki rastrowej i wektorowej. Zapisuje pliki graficzne w różnych formatach. Stosuje kompresję stratną i bezstratną plików graficznych oraz rozumie, jaki ma ona wpływ na wielkość plików i jakość obrazu. Zna podstawowe algorytmy kompresji: RLE, LZW i kodowanie Huffmana. Zna i charakteryzuje sposoby reprezentowania obrazu w komputerze. Projektuje grafikę przeznaczoną na stronę WWW. Potrafi wyjaśnić zasadę tworzenia obrazu 3D. Potrafi wyjaśnić zasadę powstawania animacji. Tworzy własną animację. Rozumie, jaki wpływ na wydrukowaną grafikę mają wymiary obrazu, rozdzielczość obrazu i rozdzielczość wydruku. Rozumie pojęcia: wymiary obrazu i rozdzielczość obrazu. Wykonuje czynności edycyjne, uwzględniając warstwy, transformacje, filtry, zmianę rozmiaru obrazu, kadrowanie obrazu. Rozumie, w jaki sposób jest reprezentowany obraz w komputerze. Opisuje podstawowe modele barw i ich zastosowanie. Zna własności barwy. Wie, jak działa skaner i jak powstaje cyfrowy obraz ze skanera.

Tabela 8.4. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie multimediów i grafiki komputerowej (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
Cyfrowy zapis filmu.	Rozumie pojęcie: kontener multimedialny. Wie, jak działają kodeki wideo na przykładzie pojedynczego obrazu. Potrafi zmontować film i opublikować go.

8.5. Systemy operacyjne i sieci komputerowe

Tabela 8.5. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
SYSTEMY OPERACYJNE	
Przegląd systemów operacyjnych.	Zna budowę komputera oraz rozumie rolę jego poszczególnych podzespołów. Rozumie rolę BIOS-u i wie, na czym polega rozruch komputera. Wymienia podstawowe funkcje systemu operacyjnego: • wykonywanie programów, • operacje wejścia-wyjścia, • zarządzanie systemem plików, • komunikacja, • przekazywanie komunikatów, • wykrywanie błędów. Zna budowę systemu operacyjnego. Rozróżnia podstawowe systemy operacyjne. Korzysta z różnych systemów operacyjnych. Rozumie pojęcia: system plików, klaster, sektor. Dokonuje podziału systemów operacyjnych. Uruchamia maszynę wirtualną z systemem operacyjnym DR-DOS, Linux i wykonuje podstawowe operacje na plikach. Zna pojęcia: środowisko wielodostępowe, wielozadaniowość, jądro systemu, GUI, maszyna wirtualna.

Podstawowe zadania systemu operacyjnego.	Zna podstawowe zadania systemu operacyjnego, do których należą: • zarządzanie procesami, • zarządzanie pamięcią operacyjną, • zarządzanie plikami, • zarządzanie systemem wejścia-wyjścia, • zarządzanie pamięcią pomocniczą (dyskową), • praca w sieci, • system ochrony. Rozumie, na czym polega i jak przebiega interpretacja poleceń (wiersz poleceń, powłoka).
SIECI KOMPUTEROWE	
Budowa, działanie i eksploatacja sieci komputerowych.	Rozróżnia rodzaje sieci ze względu na ich zakresy: LAN, MAN, WAN. Rozróżnia podstawowe rodzaje sieci ze względu na topologię. Potrafi opisać topologie sieciowe (topologia magistrali, gwiazdy, pierścienia, pełnych połączeń, mieszana). Potrafi opisać niektóre urządzenia i elementy sieci komputerowych (karty sieciowe, okablowanie sieciowe, urządzenia sieciowe). Potrafi opisać technologie sieciowe (metody dostępu do sieci). Potrafi opisać rolę urządzeń służących do rozbudowy sieci (wzmacniaki, koncentratory, mosty, przełączniki, routery, bramy). Zna i opisuje zasady administrowania siecią komputerową. Zna podstawowe usługi oraz narzędzia sieciowe i korzysta z nich.
Protokoły sieciowe. Model OSI.	Zna warstwowy model sieci komputerowych. Potrafi wyjaśnić, czym jest protokół i zestaw protokołów. Zna funkcje poszczególnych warstw modelu OSI. Potrafi określić, czy dany komputer może się komunikować z innymi komputerami w sieci. Potrafi wymienić nazwy podstawowych protokołów sieciowych i opisać ich własności.

Tabela 8.5. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie systemów operacyjnych i sieci komputerowych na poziomie rozszerzonym (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
SIECI KOMPUTEROWE	
Zestaw protokołów TCP/IP.	Rozumie, na czym polega podział na warstwy zestawu protokołów TCP/IP i porównuje go z modelem OSI. Omawia sposób przesyłania danych między protokołami TCP/IP. Zna i stosuje pojęcie: pakiet danych. Szczegółowo analizuje warstwę sieciową, używając pojęć: protokół IP, datagram IP, klasy adresów IP, adresacja IP, adresy zastrzeżone, rodzaje adresowania, maska podsieci, serwer DNS. Potrafi określić ustawienia sieciowe danego komputera i jego lokalizację w sieci.
Bezpieczeństwo informacji w sieciach.	Zna podstawowe zasady administrowania siecią komputerową w architekturze klient-serwer. Zna zagrożenia związane z pracą komputera w sieci: niszczące programy i najczęstsze ataki, sniffer, podszywanie i naśladownictwo. Potrafi podjąć działania prewencyjne. Zna różne sposoby zabezpieczeń przed zagrożeniami, takie jak: szyfrowanie SSL, certyfikaty cyfrowe, zabezpieczenia systemów operacyjnych, zapora sieciowa, filtrowanie pakietów. Rozumie konieczność stosowania programów antywirusowych i systematycznej aktualizacji bazy wirusów. Zna zagrożenia związane z przestępczością komputerową, w tym z piractwem komputerowym i nielegalnymi transakcjami w sieci. Potrafi wykonać kopie bezpieczeństwa danych. Zna podstawy kodowania danych.
Tworzenie i publikowanie własnych materiałów w sieci.	Zna strukturę oraz podstawowe znaczniki dokumentu HTML. Potrafi stworzyć i modyfikować witrynę WWW z wykorzystaniem tekstu, tabel, odpowiednich formatów grafiki i opracowań multimedialnych. Dostrzega korzyści związane ze stosowaniem arkuszy stylów. Przygotowuje własne materiały (grafiki, galerie zdjęć, wykresy, prezentacje multimedialne, animacje, dźwięk), które zamieszcza w dokumencie HTML, dodając odpowiednie menu i odsyłacze. Publikuje w sieci własne materiały oraz opracowania multimedialne. Zna i stosuje obowiązujące zasady prawne dotyczące prawa autorskiego. Zna podstawy języka PHP i wykorzystuje go w dokumentach HTML. Planuje współpracę i zespołowo wykonuje projekt witryny internetowej.

8.6. Kierunki rozwoju technologii informatycznych oraz aspekty etyczne, prawne i społeczne w zastosowaniach informatyki

Tabela 8.6. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
ASPEKTY ETYCZNE, PRAWNE I SPOŁECZNE W ZASTOSOWANIACH INFORMATYKI	
Aspekty etyczne, prawne i społeczne w zastosowaniach informatyki.	Zna przepisy prawne dotyczące wykorzystywania, przetwarzania i publikowania informacji — prawo autorskie. Zna rodzaje licencji programów komputerowych. Przestrzega zasad etyki. Dostrzega bariery w korzystaniu z technologii informacyjnej przez różne grupy społeczne. Określa korzyści i zagrożenia wynikające z rozwoju i wszechstronnego wykorzystania informatyki w różnych dziedzinach życia.

Tabela 8.6. *Treści nauczania i szczegółowe umiejętności ucznia w zakresie kierunków rozwoju technologii informatycznych oraz aspektów etycznych, prawnych i społecznych w zastosowaniach informatyki (cd.)*

TREŚCI NAUCZANIA	OSIĄGNIĘCIA UCZNIA
KIERUNKI ROZWOJU INFORMATYKI I JEJ ZASTOSOWAŃ	
Kierunki rozwoju informatyki i jej zastosowań.	Śledzi na bieżąco nowości i zna najnowsze osiągnięcia rozwoju technik informatycznych. Potrafi omówić możliwości współczesnych komputerów i tendencje w ich rozwoju. Potrafi wymienić wiele zastosowań komputerów w różnych dziedzinach życia. Śledzi na bieżąco i gromadzi informacje na temat zmian w rozwoju oprogramowania i systemów operacyjnych, ich nowych dystrybucji. Korzysta z zasobów edukacyjnych udostępnianych w portalach przeznaczonych do kształcenia na odległość.

9. Metody kontroli i oceny osiągnięć uczniów

9.1. Metody oceniania osiągnięć edukacyjnych

Analizując metody oceniania osiągnięć ucznia, należy zwrócić uwagę na wiele elementów, w tym na:

- oceniane formy aktywności specyficzne dla danego przedmiotu,
- sposób stopniowania ocen, na przykład zastosowanie ocen z wagami,
- poziomy wymagań i ich związek z ocenianiem,
- konstruowanie planów wynikowych lub rozkładów materiału i ich związek z hierarchizacją wymagań programowych.

Wszystkie wyżej wymienione elementy mają bezpośredni wpływ na sposób oceniania osiągnięć ucznia.

9.1.1. Oceniane formy aktywności

Mówiąc o formach aktywności ucznia, które podlegają ocenie, mamy na myśli działania ucznia realizowane w ramach prowadzonych zajęć informatyki, nadzorowane przez nauczyciela. Specyfika przedmiotów informatycznych związana z praktycznym podejściem do wprowadzanych zagadnień powoduje, że formy aktywności będą się tutaj różnić od tych przejawianych podczas nauki innych przedmiotów. Ponadto mogą one być odmiennie stopniowane pod względem ważności. Proponowane formy aktywności to:

- praca klasowa — obejmuje większy zakres zrealizowanego materiału (na przykład cały dział), zapowiadana jest co najmniej z tygodniowym wyprzedzeniem, może być poprawiana jeden raz; może być praktyczna (przy komputerze) lub teoretyczna (bez użycia komputera);
- sprawdzian — obejmuje materiał z dwóch, trzech ostatnich lekcji; może być praktyczny (przy komputerze) lub teoretyczny (bez użycia komputera);
- projekt indywidualny — indywidualnie wykonana praca, np. prezentacja multimedialna, strona internetowa, program komputerowy bądź inna forma pracy zrealizowanej przez ucznia z wykorzystaniem komputera;
- odpowiedź ustna — obejmuje zagadnienia dotyczące aktualnie realizowanego materiału;
- referat — praca opisowa na określony temat;
- praca na lekcji — aktywność na lekcji, ćwiczenia na lekcji, prowadzenie zajęć;
- praca w domu — zadanie domowe, praca dodatkowa wykonana przez ucznia w domu;
- projekt zespołowy — praca nad projektem grupowym realizowanym pod kierunkiem nauczyciela; ocena jest uzależniona zarówno od efektu końcowego dotyczącego zrealizowanego zadania, jak i umiejętności współdziałania w grupie oraz zaangażowania wszystkich uczestników grupy;
- korelacja międzyprzedmiotowa — realizacja projektów z innych przedmiotów z wykorzystaniem narzędzi informatycznych; nauczyciel informatyki ocenia wykonaną pracę od strony informatycznej, natomiast nauczyciel przedmiotu, którego dotyczy wykonywana praca, sprawdza efekty działań ucznia od strony tego przedmiotu;

- inne formy aktywności, np. działania wynikające z własnej inicjatywy ucznia, udział w konkursach, dodatkowe działania na rzecz szkoły, samodzielne dodatkowe projekty uczniowskie.

9.1.2. Przykładowy przedmiotowy system oceniania

Przed określeniem sposobu oceniania osiągnięć ucznia należy się zapoznać z wewnątrzszkolnym systemem oceniania. Przedmiotowy system oceniania musi być zgodny z wewnątrzszkolnym systemem oceniania, powinien być jednak rozszerzony i uszczegółowiony.

Przykładowy przedmiotowy system oceniania stanowi jedynie **propozycję i podpowiedź skierowaną do nauczyciela**. Prowadzący zajęcia powinien dostosować go do warunków, w jakich realizuje lekcje, mając na uwadze wewnątrzszkolny system oceniania, specyfikę zespołu klasowego i inne dodatkowe czynniki.

PRZEDMIOTOWY SYSTEM OCENIANIA OSIĄGNIĘĆ UCZNIÓW Z INFORMATYKI

a) Przy ocenianiu stosowana jest średnia ważona. Ocenie podlegają formy aktywności, którym odpowiadają określone wagi, co przedstawiono w tabeli 9.1.

Tabela 9.1. *Formy aktywności i ich wagi*

Forma aktywności	Waga
Praca klasowa	4
Sprawdzian	2
Projekt indywidualny	3
Odpowiedź ustna	2
Referat	2
Praca na lekcji	1
Praca w domu	1
Projekt zespołowy	2
Korelacja międzyprzedmiotowa	1

b) Jeżeli uczeń z przyczyn losowych nie może napisać pracy klasowej z całą klasą, ma obowiązek napisać ją w terminie uzgodnionym z nauczycielem, lecz nie później niż dwa tygodnie od dnia powrotu do szkoły.

c) Ocena niedostateczna z pracy klasowej może być poprawiona jeden raz w terminie uzgodnionym z nauczycielem. Na poprawę oceny uczeń ma dwa tygodnie od dnia otrzymania wyników z pracy klasowej. Poprawa pracy jest dobrowolna. Pod uwagę brane są obydwie oceny.

d) Jeżeli uczeń nie napisał pracy klasowej w ustalonych z nauczycielem terminach, we wzorze na obliczanie oceny śródrocznej (semestralnej) wstawia się 0.

e) Każdy uczeń ma prawo dwa razy w ciągu semestru być nieprzygotowany (nie wykonać zadania, nie być przygotowany do odpowiedzi ustnej), nie dotyczy to jednak prac klasowych. Brak przygotowania należy zgłosić przed lekcją.

f) Uczeń otrzymuje poprawioną pracę klasową do wglądu na lekcji.

g) Uczeń może poprawić ocenę niedostateczną z pierwszego semestru w terminie uzgodnionym z nauczycielem.

h) Zależność oceny semestralnej od średniej ważonej przedstawiono w tabeli 9.2.

Tabela 9.2. *Zależność oceny semestralnej od średniej ważonej*

Średnia ważona	Ocena semestralna
[0; 1,40)	1
(1,39; 1,65)	1+
(1,64; 1,90)	2–
(1,89; 2,25)	2
(2,24; 2,60)	2+
(2,59; 2,90)	3–
(2,89; 3,25)	3
(3,24; 3,60)	3+
(3,59; 3,90)	4–
(3,89; 4,25)	4
(4,24; 4,60)	4+
(4,59; 4,90)	5–
(4,89; 5,25)	5
(5,24; 5,60)	5+
(5,59; 6,00]	6

i) Ocena roczna jest wystawiana na podstawie średniej ważonej ocen uzyskanych podczas całego roku szkolnego, co zostało przedstawione w tabeli 9.3.

Tabela 9.3. *Zależność oceny rocznej od średniej ważonej*

Średnia ważona	Ocena roczna
[0; 1,75)	1
(1,74; 2,70)	2
(2,69; 3,70)	3
(3,69; 4,70)	4
(4,69; 5,70)	5
(5,69; 6,00]	6

9.2. Kryteria oceniania

9.2.1. Hierarchizacja wymagań programowych — poziomy wymagań i wymagania na poszczególne oceny szkolne

Wymagania programowe to opis opanowanych przez ucznia czynności, w tym wiedzy i umiejętności. W związku z brakiem możliwości ustalenia jednego poziomu nauczania dla wszystkich uczniów wymagania programowe są stopniowane, czyli dzielone na poziomy.

Spełnianie wymagań programowych jest powiązane bezpośrednio ze stopniami szkolnymi, co przedstawia tabela 9.4.

Zaprezentowany układ wymagań ma charakter hierarchiczny, co oznacza, że spełnienie wymagań wyższego poziomu jest równoznaczne z realizacją wymagań wszystkich niższych poziomów. Widoczne jest również, że spełnienie określonych wymagań warunkuje uzyskanie odpowiednich ocen szkolnych.

Model pełny, który przewiduje podział na pięć poziomów, jest niezwykle trudny do realizacji. Opisanie kolejnych poziomów w taki sposób, aby stanowiły one układ hierarchiczny, wymagałoby od nauczyciela wyjątkowej szczegółowości i precyzji, co nie zawsze jest możliwe. Stąd pojawienie się pozostałych modeli, uwzględniających podział na dwa lub trzy poziomy. Wybór właściwego modelu hierarchizacji wymagań zależy przede wszystkim od nauczyciela, ale należy wziąć również pod uwagę inne, dodatkowe czynniki, które mają później wpływ na tworzenie rozkładów materiału i planów wynikowych.

Tabela 9.4. *Powiązanie wymagań programowych ze stopniami szkolnymi*

MODELE HIERARCHIZACJI WYMAGAŃ PROGRAMOWYCH			OCENY SZKOLNE (STOPNIE)					
			celujący (6)	bardzo dobry (5)	dobry (4)	dostateczny (3)	dopuszczający (2)	niedostateczny (1)
P	P	K	+	+	+	+	+	–
		P	+	+	+	+	–	–
PP	R	R	+	+	+	–	–	–
	D	D	+	+	–	–	–	–
W	W	W	+	–	–	–	–	–

K — wymagania konieczne
P — wymagania podstawowe
R — wymagania rozszerzające
D — wymagania dopełniające
PP — wymagania ponadpodstawowe (pełne), czyli R + D
W — wymagania wykraczające

9.2.2. Rozkłady materiału i plany wynikowe

Rozkład materiału lub plan wynikowy to dokument, który opracowuje nauczyciel — zgodnie z warunkami, w jakich realizuje nauczanie. Treści planu uzależnia on zarówno od czynników, na które ma wpływ, jak i od tych, na które wpływu nie ma. Czynniki, które nie zależą bezpośrednio od nauczyciela, ponieważ są na przykład opracowywane dla całej szkoły bądź grupy nauczycieli, to wewnątrzszkolny system oceniania, przedmiotowy system oceniania, specyfika dydaktyki przedmiotowej, specyfika zespołu klasowego, podstawa programowa. Natomiast czynniki, na które nauczyciel ma wpływ, to wybór programu nauczania bądź samodzielne jego opracowanie oraz jego własny system dydaktyczny.

Od wielu lat nauczyciele przygotowują rozkłady materiału, jednak pojawiły się również plany wynikowe. Rozkład materiału ma na celu określenie podziału treści, które mają być zrealizowane na poszczególnych lekcjach. Na jego podstawie można stwierdzić, jakie wiadomości zostaną przekazane, jakimi środkami dydaktycznymi posłuży się nauczyciel i jakie cele osiągnie. W planie wynikowym najważniejszym elementem są efekty nauczania, czyli

odpowiedź na pytanie: „Jaką wiedzę i umiejętności powinni zdobyć uczniowie?". Plan wynikowy dotyczy więc bezpośrednio ucznia i jego opracowanie jest od niego uzależnione. Nauczyciel, pisząc plan wynikowy, powinien wiedzieć, w jakiej klasie i z jakimi uczniami będzie go realizował. Wskazane jest nawet, aby opracowywać plany stopniowo dla każdego kolejnego działu, a nie od razu na cały rok szkolny. Rozkład materiału jest powiązany z nauczycielem, przygotowywaniem zajęć i ich realizacją. Nie można na jego podstawie określić efektów nauczania. Z kolei plan wynikowy jest ukierunkowany na ucznia i na jego podstawie można określić oczekiwane efekty kształcenia.

W praktyce plan wynikowy to szczegółowo określone wymagania programowe obejmujące wiadomości i umiejętności, jakie uczeń powinien zdobyć. Aby poprawnie skonstruować taki plan, należy przede wszystkim dokonać wyboru treści nauczania, a więc określić działy i jednostki tematyczne. Kolejnym krokiem jest wyodrębnienie celów nauczania i opisanie wymagań edukacyjnych. Najważniejsze jest jednak zhierarchizowanie tych wymagań, czyli określenie poziomów wymagań. Należy tutaj wybrać jeden z modeli opisanych w punkcie 9.2.1, „Hierarchizacja wymagań programowych — poziomy wymagań i wymagania na poszczególne oceny szkolne", a więc podział na dwa poziomy (wymagania podstawowe i ponadpodstawowe) lub podział na trzy poziomy (wymagania podstawowe, rozszerzające i dopełniające). Ostatnim krokiem jest opracowanie schematu planu wynikowego. Tabele 9.5 i 9.6 przedstawiają przykładowe struktury planów wynikowych.

Tabela 9.5. *Struktura planu wynikowego uwzględniająca podział na trzy poziomy*

DZIAŁ	TEMATY (jednostki lekcyjne)	WYMAGANIA PROGRAMOWE		
		PODSTAWOWE	ROZSZERZAJĄCE	DOPEŁNIAJĄCE
dział 1.	temat 1.	czynności ucznia	czynności ucznia	czynności ucznia
	temat 2.	czynności ucznia	czynności ucznia	czynności ucznia
	temat 3.	czynności ucznia	czynności ucznia	czynności ucznia
dział 2.	temat 4.	czynności ucznia	czynności ucznia	czynności ucznia
	temat 5.	czynności ucznia	czynności ucznia	czynności ucznia
	…	…	…	…

Tabela 9.6. *Struktura planu wynikowego uwzględniająca podział na dwa poziomy*

DZIAŁ	TEMATY (jednostki lekcyjne)	WYMAGANIA PROGRAMOWE	
		PODSTAWOWE	PONADPODSTAWOWE
dział 1.	temat 1.	czynności ucznia	czynności ucznia
	temat 2.	czynności ucznia	czynności ucznia
	temat 3.	czynności ucznia	czynności ucznia
dział 2.	temat 4.	czynności ucznia	czynności ucznia
	temat 5.	czynności ucznia	czynności ucznia
	…	…	…

Przy opracowywaniu planów wynikowych należy zwrócić uwagę na to, aby opisywane wymagania były jednoznaczne, co wyklucza błędne interpretowanie zadań i poleceń. Istotna jest również realna możliwość spełnienia przedstawianych wymagań. Muszą one uwzględniać możliwości intelektualne uczniów. Kolejną ważną cechą jest opracowywanie wymagań w taki sposób, aby można było sprawdzić ich realizację, czyli określić, czy oraz w jakim stopniu wiadomości i umiejętności zostały opanowane.

10. Zakres możliwych modyfikacji programu nauczania

W klasie, w której realizowana jest informatyka w zakresie rozszerzonym, wskazane jest również nauczanie matematyki w zakresie rozszerzonym. Uczeń, który zamierza przystąpić do egzaminu maturalnego z informatyki, powinien wykazywać się bardzo dobrym przygotowaniem z innych przedmiotów ścisłych. Dotyczy to przede wszystkim matematyki i fizyki. Przedmiotem wspomagającym powinien być język angielski.

W zakres możliwych modyfikacji programu wchodzą następujące poprawki:

- zmiana liczby godzin przeznaczonych na poszczególne bloki tematyczne w ramowym planie nauczania, w tym zwiększenie liczby godzin bądź przesunięcie ich pomiędzy poszczególnymi działami,
- zmiana kolejności realizacji bloków tematycznych,
- samodzielne określenie poziomu wymagań dla każdego celu operacyjnego.

W każdej grupie uczniów mamy do czynienia z innym poziomem wiedzy i umiejętności oraz różnymi możliwościami intelektualnymi. Zadaniem nauczyciela jest dostosowanie realizowanego programu do potencjału uczniów. Ten cel osiąga się na poziomie rozkładów materiału i planów wynikowych, które powinny być modyfikowane na bieżąco dla poszczególnych klas.

Modyfikując program, należy zachować jego zgodność z obowiązującą podstawą programową.