

VII. Treści nauczania i przewidywane osiągnięcia ucznia

Treści nauczania oparte zostały na standardach wymagań egzaminacyjnych obowiązujących na maturze z informatyki, począwszy od matury 2005 roku, oraz podstawie programowej przedmiotu. W poniższej tabeli będziemy się odnosić bezpośrednio do wymagań egzaminacyjnych zgodnych z założeniami nowej matury, które umieszczone są w rozdziale X.

Algorytmika i programowanie

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Algorytmika		
Reprezentacja danych w komputerze – systemy zapisu liczb.	<ul style="list-style-type: none"> – zna definicje bitu, bajtu oraz pojęcie systemu pozycyjnego – potrafi przeliczać liczby zapisane w dowolnym systemie liczbowym na inny system 	I. 1.
Pojęcie algorytmu, formułowanie sytuacji i zjawisk, które da się przedstawić za pomocą algorytmu.	<ul style="list-style-type: none"> – umie powiązać pewne sytuacje życiowe z algorytmem – potrafi wskazać przykłady algorytmów w materiale nauczania innych przedmiotów (np. matematyki, fizyki) i w życiu codziennym 	I. 1. II. 5. III. 1.
Określenie specyfikacji problemu, wyróżnienie danych wejściowych oraz wyjściowych otrzymywanych w wyniku działania algorytmu. Wyszczególnienie kolejnych kroków działania algorytmu.	<ul style="list-style-type: none"> – dla klasycznych przypadków algorytmów wyszczególnia kolejne etapy algorytmu, identyfikuje dane wejściowe, wyjściowe, zmienne pomocnicze – poprawnie zapisuje specyfikację problemu algorytmicznego 	I. 1. II. 7. III. 1. III. 2. III. 3. III. 4.
Zapis algorytmu za pomocą: – pseudojęzyka, z wyszczególnieniem listy kroków, – schematu blokowego.	<ul style="list-style-type: none"> – umie zapisać algorytm – wybiera najkorzystniejszą formę zapisu ze względu na jakość algorytmu i problem, który algorytm rozwiązuje 	I. 1. II. 5. II. 7. III. 1. III. 2.
Pojęcia związane z algorytmem: poprawność, skończoność, złożoność algorytmu, uniwersalność. Analiza porównawcza różnych algorytmów rozwiązujących ten sam problem.	<ul style="list-style-type: none"> – oblicza złożoność algorytmów dla prostych przykładów – wybiera algorytm najbardziej optymalny – umie ocenić poprawność algorytmu, jego zgodność ze specyfikacją problemu 	I. 1. II. 5. II. 7. III. 1. III. 4.
Analiza schematów blokowych algorytmów, odczytywanie sposobu działania na podstawie schematu blokowego. Odczytywanie danych wyjściowych algorytmu dla zadanych danych wejściowych.	<ul style="list-style-type: none"> – odczytuje i nazywa problem na podstawie schematu blokowego – dla zadanych danych wejściowych potrafi przez analizę odczytać zbiór wyników działania algorytmu 	I. 1. II. 7. II. 8. III. 4.

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Programowanie		
Język programowania. Klasyfikacja języków programowania. Wybór języka programowania, w którym uczniowie będą zapisywać algorytmy, i uzasadnienie tego wyboru.	<ul style="list-style-type: none"> – umie wymienić kilka podstawowych języków programowania – wie, na czym polegają różnice pomiędzy językami niższego i wyższego poziomu 	I. 1. I. 3. II. 1.
Wyróżnienie etapów programowania: specyfikacja problemu, zapis za pomocą schematu blokowego, implementacja w języku programowania, kompilacja, uruchomienie, testowanie przypadków granicznych. Rola kompilatora, linkera, debugera. Podstawowe błędy kompilacji i ich likwidowanie. Konstruowanie kodu źródłowego z zachowaniem estetyki i czytelności (wcięcia, komentarze).	<ul style="list-style-type: none"> – zna podstawowe pojęcia związane z programowaniem w danym języku – zna rolę kompilatora, debugera, linkera – umie wyjaśnić pojęcie kompilacji – rozumie potrzebę dołączania plików nagłówkowych – rozumie rolę pamięci operacyjnej i procesora w działaniu programu – ma świadomość potrzeby zachowania zasad estetyki tworzonego przez siebie kodu i stosuje je w praktyce 	I. 1. II. 1. II. 7. II. 8. III. 4.
Stałe i zmienne w programie. Podstawowe typy zmiennych – typy proste. Nazewnictwo stałych i zmiennych, słowa kluczowe.	<ul style="list-style-type: none"> – potrafi wymienić podstawowe typy zmiennych – umie dla danej sytuacji problemowej określić potrzebę zastosowania zmiennych wybranego przez siebie typu – zna potrzebę zastosowania stałych w programie 	III. 1. III. 2. III. 3.
Instrukcje: przypisania i porównania.	<ul style="list-style-type: none"> – umie zastosować instrukcje przypisania i porównania – potrafi rozróżnić sytuacje ze względu na zastosowanie obu instrukcji 	III. 1. III. 2. III. 3.
Podstawowe operatory: matematyczne, relacji i logiczne.	– umie zastosować podstawowe operatory w celu rozwiązania problemu algorytmicznego	III. 1.
Instrukcja warunkowa i instrukcja wyboru, zastosowanie w programach (np. rozwiązywanie równania liniowego, równania kwadratowego).	– stosuje instrukcje warunkowe i wyboru przy rozwiązywaniu prostych problemów algorytmicznych	I. 5. a II. 7. II. 8. III. 1. III. 2. III. 4.
Implementacja prostych algorytmów liniowych i rozgałęzionych w arkuszu kalkulacyjnym.	– umie tworzyć arkusze kalkulacyjne, rozwiązujące proste problemy algorytmiczne	I. 5. II. 1 III. 1
Algorytmy z instrukcjami pętli, algorytmy iteracyjne (np. szukanie miejsca zerowego funkcji w zadanym przedziale za pomocą metody kolejnych przybliżeń, algorytm znajdowania losowo wygenerowanej liczby z zadanego przedziału), zastosowanie generatora liczb losowych (np. metoda Monte Carlo, ruchy Browna). Algorytm badania, czy liczba jest pierwsza, algorytm znajdowania największe-	<ul style="list-style-type: none"> – umie stosować podstawowe instrukcje języka przy rozwiązywaniu problemów numerycznych – rozumie i stosuje metody iteracyjne oraz metody kolejnych przybliżeń – ma świadomość istnienia generatora liczb pseudolosowych i stosuje go w implementowanych przez siebie programach 	I. 1. I. 5. a II. 5. II. 7. II. 8. III. 1. III. 2. III. 4. III. 6.

go wspólnego dzielnika dla pary liczb. Znajdowanie pierwiastka kwadratowego z liczby dodatniej metodą Newtona–Raphsona.	<ul style="list-style-type: none"> – zapoznaje się z klasycznymi algorytmami iteracyjnymi – wie, czym są metody numeryczne, pozna je na przykładach 	
Procedury i funkcje. Deklaracja, sposób przekazywania parametrów do funkcji: parametry formalne i aktualne. Zmienne globalne i lokalne.	<ul style="list-style-type: none"> – rozumie potrzebę tworzenia autonomicznych i uniwersalnych funkcji w programie, tworzy je i potrafi wykorzystać – zna sposoby przekazywania argumentów do funkcji i umie je dopasowywać do rozwiązywanego problemu 	II. 8. III. 2. III. 4.
Całka Riemanna – implementacja programu obliczającego całkę oznaczoną bezpośrednio z definicji.	<ul style="list-style-type: none"> – potrafi powiązać wiedzę matematyczną z umiejętnością implementacji programów – korzystając z metody definicyjnej całki Riemanna, oblicza pole powierzchni ograniczonej wykresem funkcji – stosuje poznane metody i środki do napisania programu liczącego wartość całki na podstawie definicji 	III. 1. III. 2. III. 3. III. 4. III. 6. III. 7.
Typy tablicowe – tablica jedno- i wielowymiarowa, sposób przekazywania tablic do funkcji. Wykorzystanie tablic do implementacji klasycznych algorytmów, np. zamiana zapisu liczby z systemu dziesiętnego na system binarny.	<ul style="list-style-type: none"> – rozumie sposób działania i obsługi tablic – umie w sposób poprawny stosować tablice w programach – zapisuje liczbę w systemie binarnym lub systemie o innej podstawie liczenia (np. w systemie trójkowym, czwórkowym itp.) 	II. 5. II. 7. III. 2.
Przeszukiwanie tablicy w celu znalezienia wyróżnionego elementu, znajdowanie powtarzających się elementów tablicy. Metoda „dziel i zwyciężaj” – zastosowanie w pracy z tablicą jednowymiarową.	<ul style="list-style-type: none"> – umie obsługiwać tablicę jednowymiarową – porównuje algorytmy rozwiązujące ten sam problem (np. przeszukiwanie tablicy z wartownikiem i bez) – poznaje, na czym polega metoda „dziel i zwyciężaj” 	I. 5. b II. 5. II. 7. III. 1. III. 2. III. 4.
Sortowanie tablicy metodą bąbelkową, metodą przez wstawianie i wybór – implementacja programów. Porównanie złożoności obliczeniowej poszczególnych typów sortowań. Przeszukiwanie binarne.	<ul style="list-style-type: none"> – sortuje tablicę – potrafi wybrać metodę sortowania w zależności od typu i wielkości tablicy – umie samodzielnie określić złożoność obliczeniową wymienionych algorytmów sortujących 	I. 5. c I. 5. d II. 5. II. 7. II. 8. III. 1. III. 2. III. 3. III. 4.
Obsługa tablicy dwuwymiarowej (macierzy): dodawanie, mnożenie, wyszukiwanie w macierzy kolumny, której suma elementów jest maksymalna, wypełnianie macierzy w charakterystyczny sposób (np. kolejnymi wyrazami zadanego ciągu arytmetycznego).	<ul style="list-style-type: none"> – operuje tablicami dwuwymiarowymi – umie podać przykłady, w których wykorzystuje się tablice dwuwymiarowe, oraz implementuje przykładowe programy (np. gry w kółko i krzyżyk, w statki) – zauważa analogię pomiędzy grami planszowymi a tablicami dwuwymiarowymi 	I. 5. h II. 5. II. 7. II. 8. III. 1. III. 2. III. 3. III. 4.
Sprowadzanie macierzy do postaci trójkątnej, rozwiązywanie układu n równań liniowych o n niewiadomych metodą Gaussa – implementacja programu¹.	<ul style="list-style-type: none"> – potrafi wykorzystać tablice dwuwymiarowe do rozwiązywania zaawansowanych zagadnień numerycznych 	I. 5. g I. 5. h II. 7. II. 8.

¹ Zakładamy, że jest to temat realizowany przez uczniów najzdolniejszych, na przykład w czasie, gdy pozostali piszą prostszy program z wykorzystaniem tablic dwuwymiarowych.

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
	<ul style="list-style-type: none"> – zna podstawowe metody numeryczne stosowane na tablicach – zna niektóre pojęcia związane z teorią macierzy (macierz trójkątna, wyznacznik macierzy) 	<ul style="list-style-type: none"> III. 1. III. 2. III. 3. III. 4.
<p>Rekurencja – pojęcie, przykłady implementacji programów rekurencyjnych: działanie silnia, ciąg Fibonacciego, potęga o wykładniku naturalnym. Znajdowanie wartości wyrazów ciągów zdefiniowanych rekurencyjnie. Odwracanie ciągu znaków wprowadzonego z klawiatury. Rekurencyjna metoda zamiany liczb pomiędzy systemami liczenia. Schemat Hornera. Złożoność obliczeniowa algorytmów rekurencyjnych.</p>	<ul style="list-style-type: none"> – umie stosować rozwiązania rekurencyjne do grupy zagadnień algorytmicznych – rozumie pojęcie rekurencji – potrafi samodzielnie stworzyć przykłady zastosowań rekurencyjnych i powiązać sytuacje życiowe z rozwiązaniami rekurencyjnymi – poznaje grupę metod sortowania ciągu opartą na rekurencji – potrafi wymienić zarówno zalety, jak i wady rozwiązań rekurencyjnych 	<ul style="list-style-type: none"> I. 1. I. 5. e I. 5. f II. 5. II. 7. II. 8. III. 1. III. 3. III. 4.
<p>Sortowanie przez scalanie. Sortowanie szybkie (ang. <i>quicksort</i>).</p>		
<p>Implementacja programu „Wieże Hanoi” w sposób rekurencyjny.</p>	<ul style="list-style-type: none"> – potrafi przeanalizować, określić i rozwiązać problem w sposób rekurencyjny 	<ul style="list-style-type: none"> II. 5. II. 7. II. 8.
<p>Implementacja programu „Problem skoczka szachowego” lub „Problem ośmiu hetmanów” w sposób rekurencyjny².</p>	<ul style="list-style-type: none"> – zauważa modyfikację czystej metody rekurencyjnej: rekurencję z powrotami – umie zastosować ją w praktyce 	<ul style="list-style-type: none"> III. 1. III. 2. III. 3. III. 4. III. 7.
<p>Typy strukturalne. Definiowanie własnej, wielopolowej struktury. Przekazywanie zmiennych strukturalnych do funkcji.</p>	<ul style="list-style-type: none"> – potrafi podać przykłady struktur i zastosować je w praktyce – umie utworzyć tablicę przechowującą elementy typu strukturalnego 	<ul style="list-style-type: none"> II. 8. III. 1. III. 2. III. 5.
<p>Obsługa wejścia/wyjścia. Odczyt z plików, zapis do plików.</p>	<ul style="list-style-type: none"> – umie obsługiwać pliki – potrafi importować dane z plików zewnętrznych oraz eksportować wyniki do plików zewnętrznych 	<ul style="list-style-type: none"> II. 2. III. 5.
<p>Tworzenie prostej bazy danych z uwzględnieniem możliwości zapisu elementów bazy do pliku i ich odczytu z pliku, sortowania elementów bazy według wybranego klucza oraz wyświetlania elementów bazy.</p>	<ul style="list-style-type: none"> – potrafi zastosować zdobytą wiedzę i umiejętności z zakresu obsługi tablic oraz wykorzystania typów strukturalnych do stworzenia prostej bazy danych – umie utworzyć funkcje obsługujące bazę 	<ul style="list-style-type: none"> II. 2. II. 8. III. 1. III. 2. III. 5.
<p>Metody szyfrowania: XOR, szyfr Cezara – implementacja programów szyfrujących.</p>	<ul style="list-style-type: none"> – potrafi wymienić kilka metod szyfrowania i deszyfrowania – umie zaimplementować program szyfrujący – proponuje własną metodę szyfrowania 	<ul style="list-style-type: none"> II. 5. II. 8. III. 1. III. 3. III. 4.
<p>Pojęcie wskaźnika, wskaźniki na zmienne typów prostych, obsługa i przeglądanie tablicy za pomocą wskaźnika, zmiana wartości elementów tablicy. Tablice dynamiczne.</p>	<ul style="list-style-type: none"> – rozpoznaje sytuacje, w których potrzebne jest tworzenie dynamicznych struktur danych – świadomie wybiera metody „pamięciooszczędne” 	<ul style="list-style-type: none"> I. 8. III. 1. III. 2. III. 3. III. 4.

² Propozycja dla uczniów zdolniejszych do realizacji w czasie, gdy inni uczniowie rozwiązują prostsze zadania z zastosowaniem rekurencji.

Zmienne dynamiczne na przykładzie listy jedno- i dwukierunkowej.	<ul style="list-style-type: none"> - stosuje tablice dynamiczne - rozpoznaje sytuacje, w których najkorzystniej jest zastosować strukturę listy - tworzy strukturę listy i ją obsługuje - podaje elementy listy, kasuje wybrane elementy, sortuje elementy listy - potrafi zaprojektować prostą klasę wraz z poprawnymi metodami (np. klasę liczb ułamkowych) - stosuje konstruktory klasy - tworzy obiekty zaprojektowanej klasy 	III. 6. III. 7.
Programowanie obiektowe; pojęcie klasy, obiektu, implementacja programu prezentującego klasę liczb ułamkowych. Pojęcie konstruktora, zastosowanie w projekcie klasy.		
Praca przy grupowym projekcie programistycznym.	<ul style="list-style-type: none"> - potrafi dzielić duży projekt na podproblemy - wykonuje odpowiedzialnie przydzieloną mu część pracy - pozostaje w bieżącym kontakcie z pozostałymi członkami grupy, wykonującymi inną część projektu 	II. 1. II. 2. II. 8. III. 1.

Bazodanowe funkcje arkusza kalkulacyjnego

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Wymagania dotyczące tworzenia tabeli nadającej się do wykorzystania mechanizmów bazodanowych arkusza.	<ul style="list-style-type: none"> - zna pojęcie bazy danych - potrafi wymienić, jakie wymagania musi spełniać tabela, która będzie pełnił funkcję bazy danych 	II. 1. II. 8. III. 1.
Autofiltr jako najprostsze narzędzie ułatwiające selekcjonowanie informacji.	<ul style="list-style-type: none"> - potrafi włączyć autofiltr - wykorzystuje autofiltr do selekcji danych 	II. 1. II. 8.
Filtr zaawansowany – narzędzie do wyszukiwania informacji o złożonym kryterium wyboru.	<ul style="list-style-type: none"> - zna zasadę tworzenia filtra zaawansowanego - potrafi wskazać sytuacje wymagające stosowania filtra zaawansowanego - umie wyselekcjonować informacje z tabeli, spełniające złożone kryterium filtrowania 	II. 1. II. 8. III. 4.
Synteza informacji zawartych w tabeli za pomocą sum pośrednich. Użycie funkcji Suma, Licznik, Średnia.	<ul style="list-style-type: none"> - zna pojęcie suma pośrednia i wie, do czego służy to narzędzie - używa sum pośrednich do analizy tabel z zastosowaniem funkcji Suma, Licznik i Średnia 	I. 1. II. 8. III. 4. III. 6.
Tabele przestawne jako narzędzie do automatycznego gromadzenia zbiorczych podsumowań.	<ul style="list-style-type: none"> - wie, do czego służą tabele przestawne i potrafi je zastosować w praktyce 	I. 1. II. 1. II. 8.
Graficzna prezentacja wyników, typy wykresów.	<ul style="list-style-type: none"> - zna możliwości tworzenia różnych typów wykresów - rozumie celowość graficznej interpretacji danych 	II. 1. II. 8. III. 4.
Tworzenie wykresu przestawnego na podstawie tabeli przestawnej.	<ul style="list-style-type: none"> - zna zasady tworzenia wykresu przestawnego i potrafi stworzyć taki wykres 	II. 1. II. 8. III. 4. III. 6.

Przetwarzanie informacji w relacyjnych bazach danych

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
<p>Pojęcie i przykłady zastosowań relacyjnych baz danych – przypomnienie wiadomości. Prezentacja przykładowej relacyjnej bazy danych.</p>	<ul style="list-style-type: none"> – potrafi uzasadnić konieczność gromadzenia informacji w relacyjnych bazach danych – umie wskazać przykłady baz danych w różnych dziedzinach życia – wyjaśnia zalety elektronicznych baz danych w porównaniu z klasycznymi archiwami gromadzonymi na papierze 	<p>I. 1. I. 6. II. 1. II. 3. II. 6. II. 8. III. 1. III. 5.</p>
<p>Pojęcie relacji. Typy relacji: jeden do jednego, jeden do wielu, wiele do wielu.</p>	<ul style="list-style-type: none"> – potrafi na przykładzie prostej bazy danych zaproponować podział informacji na tabele w taki sposób, by uniknąć powtarzania informacji w bazie – umie samodzielnie połączyć tabele odpowiednimi relacjami 	<p>I. 1. II. 1. II. 4.</p>
<p>Projekt i wykonanie relacyjnej bazy danych dotyczących np. wyników uczniów w zawodach sportowych. Odpowiedni dobór tabel i relacji.</p>	<ul style="list-style-type: none"> – wykorzystuje zdobytą wiedzę do tworzenia baz danych użytecznych w życiu klasy i szkoły – współpracuje z innymi uczniami w celu zgromadzenia danych do wypełnienia rekordami stworzonej bazy 	<p>II. 1. II. 2. II. 4. II. 8. III. 1. III. 4. III. 5. III. 7.</p>
<p>Tworzenie formularza wprowadzania danych za pomocą kreatora.</p>	<ul style="list-style-type: none"> – dba o stworzenie możliwie przyjaznego interfejsu do wprowadzania danych 	<p>II. 1. II. 2. II. 8. III. 1. III. 4. III. 5.</p>
<p>Uzupełnianie stworzonej relacyjnej bazy danych nowymi rekordami, wyszukiwanie rekordów, zastępowanie rekordów.</p>	<ul style="list-style-type: none"> – korzysta z metod wyszukiwania rekordów według prostego klucza – potrafi zmodyfikować rekord, usunąć go lub dodać inny 	<p>II. 1. II. 8. III. 5.</p>
<p>Importowanie danych do tabel z plików tekstowych i z arkusza kalkulacyjnego.</p>	<ul style="list-style-type: none"> – potrafi wykorzystać zapisane informacje, na przykład w edytorze tekstu lub w tabeli arkusza kalkulacyjnego do umieszczenia ich w swojej bazie danych – wyjaśnia zasadność umieszczania informacji w bazie danych 	<p>II. 1. II. 2. II. 3. III. 4. III. 5.</p>
<p>Praca z kwerendami: wyszukującą bezparametryczną i parametryczną oraz aktualizującą.</p>	<ul style="list-style-type: none"> – rozumie pojęcie kwerendy i potrafi wskazać sytuacje wymagające zastosowania kwerend – wskazuje funkcjonalne różnice pomiędzy poszczególnymi rodzajami kwerend 	<p>I. 1. II. 4. III. 1.</p>
<p>Tworzenie kwerend wyszukiwujących informacje.</p>	<ul style="list-style-type: none"> – potrafi wykorzystać umiejętność pisania kwerendy do wyszukiwania informacji z bazy danych spełniających zadane kryterium 	<p>II. 3. II. 4. III. 1. III. 4.</p>

Zastosowanie kwerendy aktualizującej jako narzędzia do zautomatyzowanego modyfikowania bazy danych.	– tworzy użyteczne kwerendy usprawniające aktualizację bazy danych	I. 1. II. 4. III. 1. III. 5.
Generowanie raportów z bazy danych za pomocą kreatorów.	– umie wygenerować z bazy danych raport zawierający wybrane informacje	I. 1. II. 1. II. 2. II. 3. II. 8.
Tworzenie i korzystanie z formularzy, dopisywanie i modyfikacja danych z wykorzystaniem formularzy.	– dba o estetyczną stronę bazy danych – poznaje mechanizmy ułatwiające operowanie na bazie danych – tworzy formularze ułatwiające praktyczne zastosowanie stworzonej bazy danych	II. 1. II. 4.
Mechanizmy ochrony bazy danych.	– wskazuje zagrożenia spowodowane brakiem zabezpieczeń bazy danych – potrafi stosować mechanizmy ochrony danych przed nieupoważnioną bądź przypadkową modyfikacją	I. 1. II. 1. III. 4.
Podstawy języka SQL, zasady składni i podstawowe instrukcje.	– wie, co to jest język zapytań SQL i do czego służy – zna podstawowe polecenia języka i zasady składni	II. 4. III. 2. III. 5.
Analiza poleceń zapisanych w języku SQL.	– potrafi analizować fragmenty kodów napisanych w języku SQL, wyjaśniać polecenia i użyte komendy	II. 4. III. 5. III. 6.

Systemy i sieci komputerowe

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Obsługa i konserwacja używanego systemu operacyjnego.	– opisuje budowę nazwy pliku oraz jego atrybuty w systemie Windows – poprawnie definiuje pojęcia: wirus komputerowy, robak, trojan oraz wyjaśnia ich działanie – wyjaśnia pojęcie kompresji danych – wyszukuje pliki według zadanych kryteriów (np. data, nazwa, rozszerzenie itp.) i nadaje plikom atrybuty – uruchamia i konfiguruje program antywirusowy – tworzy kopie zapasowe systemu i danych oraz odzyskuje pliki usunięte za pomocą odpowiednich programów narzędziowych	I. 2. I. 3. II. 1. II. 6. III. 8.
Podstawy obsługi systemu Linux.	– umie wykonać podstawowe operacje systemu Linux: przeglądać zasoby w systemie, kopiować, usuwać pliki, sprawdzać i ustawiać prawa dostępu do swoich zasobów, zmieniać hasło	I. 2. I. 3. II. 1. II. 3.

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Topologie i technologie sieciowe. Rodzaje sieci ze względu na zasięg i architekturę, media transmisji, urządzenia usprawniające pracę sieci. Pojęcie protokołu i rodzaje protokołów, pojęcie serwera i rodzaje serwerów.	<ul style="list-style-type: none"> – zna podstawowe pojęcia związane z pracą sieci, na przykład: protokół połączeniowy i bezpołączeniowy, serwer DNS, serwer dedykowany, ruting dynamiczny, pakiet danych itp. – zna nazewnictwo i zastosowanie urządzeń pracujących w sieci 	I. 1. I. 2. II. 3. III. 7.
Projektowanie prostych sieci komputerowych.	<ul style="list-style-type: none"> – umie zaprojektować prostą sieć lokalną z uwzględnieniem warunków, w jakich projekt ma być realizowany, oraz funkcji, jakie ma spełniać, z możliwością wyjścia do Internetu – potrafi udostępnić zasoby komputera do sieci 	II. 3. II. 6. II. 8. III. 4. III. 6. III. 7.
Adresacja w sieci Internet, budowa adresu IP.	<ul style="list-style-type: none"> – potrafi zinterpretować adres IP – umie wymienić i wyjaśnić poszczególne elementy adresu, wyjaśnić znaczenie serwera DNS 	I. 2. II. 1. II. 3.
Bezpieczeństwo w sieci.	<ul style="list-style-type: none"> – wymienia zagrożenia płynące z pracy komputera w sieci – zna środki i metody bezpiecznej pracy w sieci – zna pojęcia związane z zabezpieczeniem sieci, na przykład: zapora ogniowa (ang. <i>firewall</i>) – konfiguruje program do zabezpieczenia zasobów komputera – zna podstawy kodowania danych 	II. 6. III. 7. III. 8.

Multimedia

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Dźwięk w komputerze.	<ul style="list-style-type: none"> – potrafi wymienić kilka typów plików dźwiękowych – wyjaśnia różnice i obszary zastosowań poszczególnych formatów – wymienia metody kompresji plików dźwiękowych – podaje zasady poszczególnych rodzajów kompresji 	I. 1. I. 2. III. 7.
Grafika 2D.	<ul style="list-style-type: none"> – wymienia formaty plików bitmapowych – zna podstawowe pojęcia związane z grafiką bitmapową – przedstawia metody kompresji plików graficznych – wyjaśnia podstawowe różnice pomiędzy grafiką wektorową a bitmapową 	I. 1. II. 2. II. 6. II. 8. III. 7.

Podstawy grafiki 3D.	<ul style="list-style-type: none"> - potrafi wyjaśnić zasadę tworzenia sceny 3D metodą „śledzenia promienia” - tworzy prostą scenę 3D za pomocą programu komputerowego 	<p>I. 1. II. 6. II. 8.</p>
Animacja komputerowa.	<ul style="list-style-type: none"> - zna zasady łączenia obrazów statycznych w animację - tworzy własną animację w postaci połączonych kilku klatek, np. w formacie gif 	<p>I. 1. II. 6. II. 8. III. 7.</p>
Tworzenie stron WWW.	<ul style="list-style-type: none"> - zna podstawowe znaczniki HTML i potrafi za ich pomocą stworzyć statyczną stronę internetową z wykorzystaniem stylu, zawierającą tabele, hiperłącza, grafikę, dźwięk - wyszukuje w Internecie potrzebne informacje i wykorzystuje je do tworzenia własnych stron - zna i stosuje obowiązujące zasady prawne dotyczące prawa autorskiego - umieszcza w kodzie strony istniejące skrypty pisane w Javie 	<p>I. 4. II. 1. II. 2. II. 3. II. 6. III. 8.</p>

Tendencje w rozwoju informatyki i jej zastosowań

Treści nauczania	Cele edukacyjne i umiejętności ucznia	Standardy wymagań
Historia i stan obecny architektury komputerów i systemów operacyjnych.	<ul style="list-style-type: none"> - potrafi odnaleźć i zaprezentować informacje dotyczące zmian w budowie komputerów i systemów operacyjnych - dyskutuje na temat możliwych scenariuszy rozwoju - wyszukuje informacje na temat stosowania urządzeń komputerowych w różnych dziedzinach życia 	<p>I. 1. I. 2. I. 3. I. 6. II. 1. II. 3. III. 7. III. 8.</p>
Przedstawienie z najnowszych osiągnięć technik informacyjnych i informatyki.	<ul style="list-style-type: none"> - zna najnowsze osiągnięcia w rozwoju technik informacyjnych - śledzi na bieżąco nowe trendy z nimi związane - potrafi określić ważniejsze etapy w rozwoju informatyki dotyczące pojęć, metod i środków technicznych 	<p>I. 1. I. 2. I. 3. I. 6. II. 1. II. 3. III. 7. III. 8.</p>
Informacje o możliwościach komputerów i ich granicach.	<ul style="list-style-type: none"> - potrafi wymienić możliwości nowoczesnych komputerów - ocenia możliwości i granice stosowania komputerów (systemy równoległe, sieci neuronowe – sztuczna inteligencja) 	<p>I. 1. I. 3. I. 6. II. 1. II. 3. III. 7. III. 8.</p>