

INFORMATYKA

dla szkół ponadgimnazjalnych. Zakres rozszerzony

Opis założonych osiągnięć ucznia – przykłady wymagań na poszczególne oceny szkolne

Grażyna Koba

Spis treści

1. Podstawy algorytmiki i programowania	2
2. Komputer i sieci komputerowe	6
3. Opracowywanie informacji za pomocą komputera	9
4. Bazy danych	13
5. Algorytmika i programowanie – problemy zaawansowane	16
6. Aplikacje bazodanowe dostępne za pośrednictwem Sieci	19

1. Podstawy algorytmiki i programowania

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wie, co to jest algorytm. Określa dane do zadania oraz wyniki. Zna podstawowe zasady graficznego prezentowania algorytmów: podstawowe rodzaje bloków, ich przeznaczenie i sposoby umieszczania w schemacie blokowym. Potrafi narysować (odręcznie) schemat blokowy algorytmu liniowego. Potrafi napisać prosty program, wyświetlający napis na ekranie monitora.	Wymienia przykłady czynności i działań w życiu codziennym oraz zadań szkolnych, które uważa się za algorytmy. Zna pojęcie specyfikacji zadania. Zna wybrane sposoby prezentacji algorytmów. Przedstawia algorytm liniowy w postaci listy kroków. Podczas rysowania schematów blokowych potrafi wykorzystać Autokształty z edytora tekstu. Określa pojęcia <i>program komputerowy, język programowania</i>. Zapisuje prosty algorytm liniowy w wybranym języku programowania. Potrafi go skompilować i uruchomić.	Określa zależności między problemem, algorytmem a programem komputerowym. Potrafi odpowiedzieć na pytanie, czy istnieją działania, które nie mają cech algorytmów, i podać przykłady. Przedstawia dokładną specyfikację dowolnego zadania. Analizuje poprawność budowy schematu blokowego. Wyjaśnia pojęcia: <i>program wynikowy, kompilacja, translacja, interpretacja</i>. Realizuje przykładowy algorytm liniowy w wybranym języku programowania. Wykonuje program i testuje go, podstawiając różne dane.	Zapisuje dowolny algorytm w wybranej przez siebie postaci (notacji). Potrafi samodzielnie zapoznać się z nowym programem edukacyjnym przeznaczonym do konstrukcji schematów blokowych. Potrafi przeprowadzić szczegółową analizę poprawności konstrukcji schematu blokowego. Analizuje działanie algorytmu dla przykładowych danych. Potrafi posłużyć się kompilatorem danego języka. Potrafi wskazać i poprawić błędy w programie.	Przestrzega zasad zapisu algorytmów w zadanej postaci (notacji). Stosuje poznane metody prezentacji algorytmów w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia. Potrafi samodzielnie zapoznać się z kompilatorem wybranego języka programowania. Samodzielnie pisze program realizujący algorytm liniowy.

Podstawowe zasady programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna klasyfikację języków programowania. Zna ogólną budowę programu i najważniejsze elementy języka programowania – słowa kluczowe, instrukcje, wyrażenia, zasady składni. Potrafi zrealizować prosty algorytm liniowy i z warunkami w języku wysokiego poziomu; potrafi skompilować i uruchomić program.	Zapisuje program w czytelnej postaci – stosuje wcięcia, komentarze. Rozumie znaczenie i działanie podstawowych instrukcji (m.in. iteracyjnych, warunkowych) wybranego języka programowania wysokiego poziomu. Wie, na czym polega programowanie strukturalne. Rozróżnia i poprawia błędy kompilacji i błędy wykonania. Potrafi zrealizować algorytmy iteracyjne w języku wysokiego poziomu.	Wymienia i omawia modele programowania. Potrafi prezentować złożone algorytmy (z podprogramami) w wybranym języku programowania. Zna i stosuje instrukcje wyboru. Prezentuje wybrane algorytmy iteracyjne w postaci programu komputerowego. Zna rekurencyjne realizacje prostych algorytmów. Rozumie i stosuje zasady programowania strukturalnego. Deklaruje procedury i funkcje bez parametrów. Wie, na czym polega różnica pomiędzy przekazywaniem parametrów przez zmienną i przez wartość w procedurach i funkcjach. Rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą.	Wie, jaka jest różnica między językiem wysokiego poziomu a językiem wewnętrznym; potrafi określić rolę procesora i pamięci operacyjnej w działaniu programów. Deklaruje procedury i funkcje z parametrami. Wie, jakie znaczenie ma zasięg zmiennej. Definiuje funkcje rekurencyjne. Potrafi prezentować algorytmy rekurencyjne w postaci programu. Zapisuje w postaci programu wybrane algorytmy sortowania, algorytmy na tekstach, definiując odpowiednie procedury lub funkcje.	Ocenia efektywność działania programu. Wskazuje podobieństwa i różnice dotyczące tworzenia programów zapisanych w różnych językach programowania; wyjaśnia działanie poszczególnych instrukcji, sposób deklaracji zmiennych. Sprawnie definiuje i stosuje procedury i funkcje w programach. Sprawnie korzysta z dodatkowej, fachowej literatury. Rozwiązuje przykładowe zadania z matury i olimpiady informatycznej.
Wymienia przykłady prostych struktur danych. Potrafi zadeklarować zmienne typu liczbowego (całkowite, rzeczywiste) i stosować je w zadaniach.	Wie, czym jest zmienna w programie i co oznacza przypisanie jej konkretnej wartości. Rozróżnia struktury danych: proste i złożone. Podaje przykłady.	Potrafi zastosować łańcuchowy i tablicowy typ danych w zadaniach. Deklaruje typ tablicowy i łańcuchowy.	Rozumie, na czym polega dobór struktur danych do algorytmu. Wczytuje i wyprowadza elementy tablicy. Wprowadza dane tekstowe. Tworzy programy, dobierając odpowiednie struktury danych do programu.	Dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu.

Techniki algorytmiczne i wybrane algorytmy				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Określa sytuacje warunkowe. Podaje przykłady zadań, w których występują sytuacje warunkowe. Wie, na czym polega powtarzanie tych samych operacji. Potrafi omówić na konkretnym przykładzie algorytm znajdowania najmniejszego z trzech elementów.	Potrafi odróżnić algorytm liniowy od algorytmu z warunkami (z rozgałęzieniami). Zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego. Podaje ich przykłady. Wie, od czego zależy liczba powtórzeń. Tworzy schemat blokowy algorytmu z warunkiem prostym i pętlą. Testuje rozwiązanie dla wybranych danych. Określa problemy, w których występuje rekurencja i podaje przykłady „zjawisk rekurencyjnych” – wziętych z życia i zadań szkolnych. Potrafi omówić algorytm porządkowania elementów (metodą przez wybór) na praktycznym przykładzie, np. wybierając najwyższego ucznia z grupy. Omawia wybrane algorytmy sortowania. Omawia wybrany algorytm na tekstach (np. tworzenie anagramów).	Analizuje algorytmy, w których występują powtórzenia (iteracje). Ocenia zgodność algorytmu ze specyfikacją. Zna sposoby zakończenia iteracji. Określa kroki iteracji. Potrafi zapisać w wybranej notacji np. algorytm sumowania n liczb, algorytm obliczania silni, znajdowania minimum w ciągu n liczb, algorytm rozwiązywania równania liniowego. Potrafi zapisać algorytm z warunkami zagnieżdżonymi i pętlą w wybranej postaci. Zna iteracyjną postać algorytmu Euklidesa. Zna rekurencyjną realizację wybranego algorytmu, np. silni. Zna przynajmniej dwie techniki sortowania (np. bąbelkowe, przez wybór) i zapisuje wybrany algorytm w postaci programu komputerowego. Omawia wybrane algorytmy na tekstach. Potrafi wyjaśnić, na czym polega wydawanie reszty metodą zachłanną i napisać listę kroków tego algorytmu.	Zapisuje algorytmy z pętlą zagnieżdżoną. Zna metodę „dziel i zwyciężaj”, algorytm generowania liczb Fibonacciego, schemat Hornera. Omawia ich iteracyjną realizację i potrafi przedstawić jeden z nich w wybranej notacji. Zna inne algorytmy sortowania, np. pozycyjne, przez wstawianie. Wskazuje różnicę między rekurencją a iteracją. Zna rekurencyjną realizację wybranych algorytmów, np. obliczania silni i algorytm Euklidesa. Potrafi zamienić algorytm zapisany iteracyjnie na postać rekurencyjną. Zapisuje wybrany algorytm na tekstach (np. tworzenie anagramów, zliczanie znaków w tekście, sprawdzanie, czy dany ciąg jest palindromem) w postaci programu komputerowego. Zapisuje algorytm wydawania reszty metodą zachłanną w postaci programu komputerowego.	Rozumie dokładnie technikę rekurencji (znaczenie stosu). Potrafi ocenić, kiedy warto stosować iterację, a kiedy rekurencję. Zna trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, szukanie wzorca w tekście. Potrafi zapisać je w różnych notacjach (również w języku programowania wysokiego poziomu). Korzysta samodzielnie z dodatkowej literatury fachowej.

Elementy analizy algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wymienia własności algorytmów. Potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków lub w postaci schematu blokowego dla przykładowych danych i ocenić w ten sposób jego poprawność.	Zna i omawia własności algorytmów. Potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją. Określa liczbę prostych działań zawartych w algorytmie.	Rozumie, co to jest złożoność czasowa algorytmu i potrafi określić liczbę operacji wykonywanych na elementach zbioru w wybranym algorytmie sortowania. Rozróżnia złożoność czasową i pamięciową.	Wie, jak ocenić złożoność pamięciową algorytmu. Potrafi porównać złożoność obliczeniową różnych algorytmów tego samego zadania dla tych samych danych. Wie, kiedy algorytm jest efektywny.	Określa złożoność czasową i pamięciową wybranych algorytmów. Zna odpowiednie wzory. Określa efektywność algorytmów.

2. Komputer i sieci komputerowe

Reprezentacja danych w komputerze				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna pojęcie systemu pozycyjnego. Wie, co to jest system binarny. Korzystając z przykładów, potrafi obliczyć wartość dziesiętną liczby zapisanej w systemie dwójkowym.	Wyjaśnia, co to jest system binarny, i potrafi dokonać zamiany liczby z systemu dziesiętnego na binarny i odwrotnie.	Zna system szesnastkowy i potrafi wykonać konwersję liczb dziesiętnych na liczby w systemie szesnastkowym i odwrotnie. Zna zależność między systemem binarnym i szesnastkowym.	Potrafi wykonać dowolną konwersję pomiędzy systemem dziesiętnym, dwójkowym i szesnastkowym. Potrafi napisać program obliczający wartość dziesiętną liczby dwójkowej.	Potrafi napisać program (w wersji iteracyjnej i rekurencyjnej) realizujący algorytm zamiany liczby dziesiętnej na postać binarną. Potrafi napisać program realizujący algorytm umożliwiający zamianę liczb z systemu szesnastkowego na dziesiętny i odwrotnie.

Kompresja i szyfrowanie danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wie, co to jest kompresja danych. Zna przynajmniej jeden algorytm kompresji danych. Wie, czym jest szyfrowanie danych. Zna przynajmniej jeden algorytm szyfrowania danych.	Rozumie, na czym polega kompresja danych i w jakim celu się ją wykonuje. Wymienia rodzaje kompresji. Omawia jeden przykładowy algorytm kompresji. Koduje tekst, używając alfabetu Morse'a. Wymienia przykładowe algorytmy szyfrowania. Potrafi zaszyfrować i odszyfrować prosty tekst.	Wie, co to jest współczynnik kompresji. Omawia rodzaje kompresji: kompresję stratną i bezstratną. Podaje przykłady algorytmów kompresji stratnej i bezstratnej. Omawia algorytm statyczny i słownikowy. Omawia przynajmniej dwa algorytmy szyfrowania: szyfr podstawieniowy i przestawieniowy.	Potrafi policzyć współczynnik kompresji. Wyjaśnia różnicę pomiędzy algorytmem statycznym a słownikowym. Stosuje algorytm słownikowy do kompresji ciągu znaków. Omawia przykładowy szyfr z kluczem. Stosuje szyfr Vigenère'a do zaszyfrowania ciągu znaków. Omawia wykorzystanie algorytmów szyfrowania w podpisie elektronicznym.	Wyszukuje dodatkowe informacje na temat kompresji i szyfrowania danych. Omawia inne algorytmy kompresji i szyfrowania. Potrafi zapisać wybrany algorytm kompresji lub szyfrowania w postaci programu. Zapoznaje się samodzielnie z kodem Huffmana i pokazuje na przykładzie jego zastosowanie. Omawia, czym się zajmuje stenografia, samodzielnie wyszukując informacje na ten temat. Wyjaśnia, w jaki sposób tworzy się podpis elektroniczny.

Komputer i system operacyjny				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wymienia części składowe zestawu komputerowego, podaje ich parametry i przeznaczenie. Rozróżnia rodzaje pamięci komputera, określa ich własności i przeznaczenie. Wie, co to jest bit i bajt. Wie, co to jest system operacyjny, wymienia i omawia jego podstawowe funkcje oraz z nich korzysta. Podaje przykłady systemów operacyjnych.	Potrafi sklasyfikować środki (urządzenia) i narzędzia (oprogramowanie) technologii informacyjnej. Wie, jak działa komputer. Wyjaśnia rolę procesora. Rozumie sposób organizacji pamięci komputerowej. Zna jednostki pamięci, pojemności nośników i programów. Omawia dwa przykładowe systemy operacyjne.	Analizuje model komputera zgodny z ideą von Neumanna. Potrafi wymienić i omówić rodzaje aktualnie używanych komputerów. Omawia, jak działa procesor. Wymienia i omawia popularne systemy operacyjne: Microsoft Windows, Unix, Linux, Mac Os.	Omawia szczegółowo model komputera zgodny z ideą von Neumanna. Wyjaśnia, w jaki sposób procesor wykonuje dodawanie liczb. Porównuje cechy różnych systemów operacyjnych, np. Microsoft Windows, Unix, Linux, Mac Os. Omawia przykładowe systemy operacyjne dla urządzeń mobilnych. Samodzielnie zapoznaje się z możliwościami nowych urządzeń związanych z TIK.	Omawia szczegółowo system Linux, porównując go do systemu Microsoft Windows. Korzystając z dodatkowych źródeł, omawia kierunek rozwoju systemów operacyjnych. Korzystając z dodatkowych źródeł, omawia najnowsze osiągnięcia dotyczące systemów operacyjnych stosowanych w urządzeniach mobilnych.

Sieci komputerowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wyjaśnia pojęcia: <i>sieć komputerowa, zasoby sieciowe, klient, serwer</i> . Podaje podział sieci ze względu na wielkość, ze względu na model funkcjonowania i na topologię. Potrafi wymienić kilka cech pracy w sieci, odróżniających ją od pracy na autonomicznym komputerze.	Wymienia korzyści płynące z korzystania z sieci. Wyjaśnia, na czym polega wymiana informacji w sieci. Omawia podstawowe klasy i topologie sieciowe. Potrafi wymienić urządzenia i elementy sieciowe oraz omówić ich ogólne przeznaczenie. Zna cechy systemu działającego w szkolnej pracowni. Wie, co to jest protokół komunikacyjny.	Posługuje się terminologią sieciową. Potrafi wymienić zalety i wady różnych topologii sieci. Charakteryzuje topologie gwiazdy, magistrali i pierścienia. Zna znaczenie protokołu w sieciach (w tym TCP/IP). Definiuje funkcje i usługi poszczególnych warstw modelu warstwowego sieci. Wie, co to jest adres sieciowy. Wyjaśnia, co to jest adres	Swobodnie posługuje się terminologią sieciową. Zna schemat działania sieci komputerowych. Omawia ogólnie określanie ustawień sieciowych danego komputera i jego lokalizacji w sieci (podsieci IPv4, protokół DHCP, DNS, TCP). Wie, co określa maska podsieci. Potrafi z pomocą nauczyciela zrealizować małą sieć komputerową – skonfigurować jej	Omawia szczegółowo model warstwowy sieci. Omawia różne systemy sieciowe. Dokonuje ich analizy porównawczej.

	Zna zasady pracy w sieci, m.in. zasady udostępniania zasobów. Wie, z jakich warstw składa się warstwowy model sieci. Ogólnie je omawia.	domenowy i omawia jego strukturę. Potrafi omówić ogólne zasady administrowania siecią komputerową w architekturze „klient-serwer”.	składniki, udostępnić pliki, dyski, drukarki, dodać użytkowników.	
--	---	--	--	--

Zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna etapy pracy nad projektem i bierze udział w pracy grupowej jako członek zespołu. Potrafi omówić historię komputerów. Umie wskazać ogólny kierunek zmian w technologiach komputerowych. Zna i stosuje podstawowe zasady netykiety.	Omawia etapy pracy nad projektem i bierze aktywny udział w pracy grupowej jako członek zespołu. Potrafi określić nowoczesne trendy w zastosowaniu urządzeń komputerowych. Omawia wybrane normy etyczne i prawne, m.in.: zasady korzystania z programów komputerowych, rodzaje licencji, rozpowszechnianie programów komputerowych. Omawia społeczne aspekty zastosowania informatyki.	Bierze aktywny udział w pracy grupowej jako członek zespołu, gromadząc i selekcjonując materiały do projektu. Potrafi wskazać nowości w zakresie usług internetowych oraz odszukać informacje na temat najnowszych pomysłów na komputery. Omawia wybrane normy etyczne i prawne, m.in. korzystanie z cudzych materiałów. Omawia szanse i zagrożenia związane z rozwojem informatyki i TIK, m.in.: uzależnienie od komputera i Internetu.	Pełni funkcje koordynatora w grupie. Komunikuje się z innymi członkami grupy z wykorzystaniem nowoczesnych technologii, m.in. za pomocą urządzeń mobilnych. Omawia zagadnienia bezpieczeństwa i ochrony danych oraz informacji w komputerze i w sieciach komputerowych (rodzaje zagrożeń, sposoby ochrony). Omawia rozwój informatyki i technologii informacyjno-komunikacyjnych, m.in. najważniejsze elementy procesu rozwoju informatyki i TIK.	Przygotowuje analizę porównawczą, pokazującą na przestrzeni lat rozwój informatyki, w tym sieci komputerowych, oraz multimediów. Wskazuje tendencje w rozwoju informatyki i jej zastosowań, dostrzegając przeobrażenia w tej dziedzinie w kraju i na świecie. Przygotowuje indywidualny projekt na wybrany przez siebie temat.

3. Opracowywanie informacji za pomocą komputera

Algorytmy i zależności funkcyjne w arkuszu kalkulacyjnym				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna zasady tworzenia formuł i stosowania funkcji arkusza kalkulacyjnego. Zna i stosuje zasady adresowania względnego i bezwzględnego w arkuszu kalkulacyjnym. Zapisuje w arkuszu kalkulacyjnym algorytm liniowy i z warunkami. Stosuje wybrane funkcje arkusza kalkulacyjnego. Potrafi utworzyć wykres w arkuszu kalkulacyjnym.	Zapisuje w arkuszu kalkulacyjnym algorytm z warunkami zagnieżdżonymi. Zna i stosuje zasady adresowania mieszanego w arkuszu kalkulacyjnym. Potrafi zrealizować iterację w arkuszu kalkulacyjnym. Rysuje wykres funkcji liniowej i kwadratowej.	Potrafi zrealizować pętlę zagnieżdżoną w arkuszu kalkulacyjnym. Wie, co to jest fraktal i w jaki sposób się go tworzy. Podaje przykłady fraktali. Rysuje wykres wybranej funkcji trygonometrycznej. Dobiera odpowiedni typ wykresu do prezentowanych danych.	Rysuje wykres funkcji liniowej, wielomianu, wybranej funkcji trygonometrycznej i funkcji logarytmicznej. Zna możliwości zastosowania algorytmów iteracyjnych w rysowaniu fraktali, m.in.: śnieżynki Kocha, dywanu i trójkąta Sierpińskiego. Stosuje wybrane możliwości arkusza kalkulacyjnego do rozwiązywania zadań z różnych dziedzin.	Potrafi samodzielnie zobrazować wybraną zależność funkcyjną w arkuszu kalkulacyjnym, np. algorytm rozwiązywania układu równań liniowych metodą wyznaczników. Rozumie, w jaki sposób narysować paprotkę Barnsleya w arkuszu kalkulacyjnym. Korzystając z Internetu i innych źródeł, wyszukuje dodatkowe informacje na temat geometrii fraktalnej, m.in. dotyczące jej zastosowań.

Multimedia i grafika komputerowa				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna urządzenia multimedialne, wymienia przykładowe nazwy, określa ogólnie przeznaczenie urządzeń multimedialnych. Posługuje się drukarką i skanerem. Po zeskanowaniu zapisuje obraz w pliku w formacie domyślnym. Wymienia programy do tworzenia i obróbki grafiki. Posługuje się jednym z nich w celu tworzenia własnych rysunków. Tworzy i edytuje obrazy w wybranym programie graficznym, korzystając z podstawowych narzędzi do edycji obrazu. Wie, czym różni się grafika rastrowa od wektorowej.	Korzysta z różnych urządzeń multimedialnych, zna ich działanie, podaje ich przeznaczenie. Definiuje pojęcie <i>komputer multimedialny</i>. Zna sposoby reprezentacji obrazu i dźwięku w komputerze. Zna modele barw. Zapisuje plik graficzny w różnych formatach. Zna zastosowanie poszczególnych formatów, ich zalety i wady. Wymienia formaty zapisu dźwięku. Zna podstawowe możliwości wybranych programów do edycji obrazu rastrowego i wektorowego. Potrafi wybrać fragmenty obrazu i wykonać na nich różne operacje, np. selekcje, przekształcenia (obroty, odbicia). Potrafi, korzystając z gotowego pliku video, wykonać proste operacje, tj. podzielić film na fragmenty, przyciąć film, dodać efekty.	Posługuje się sprawnie wybranymi urządzeniami multimedialnymi. Zna różne możliwości komputera w zakresie edycji obrazu, dźwięku, animacji i wideo. Omawia model barw. Zna pojęcia: RGB i CMYK. Omawia formaty plików dźwiękowych. Potrafi stosować różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy. Wie, na czym polega praca z warstwami. Wykonuje rysunki, korzystając z warstw. Potrafi tworzyć przykładowe fotomontaże. Tworzy obraz w grafice wektorowej, rysuje figury, ścieżki. Wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy w grafice rastrowej i wektorowej.	Zna i stosuje w praktyce zaawansowaną obróbkę grafiki rastrowej i wektorowej. Przy użyciu odpowiednich narzędzi potrafi zaznaczyć fragmenty obrazu nawet o skomplikowanym kształcie. Potrafi zapisywać pliki multimedialne w różnych formatach, ze szczególnym uwzględnieniem formatów internetowych. Zna pojęcia: <i>filtr</i>, <i>histogram</i>, <i>krzywa barw</i>. Potrafi zdefiniować barwy i wykonać na nich operacje. Stosuje filtry. Wykonuje ćwiczenia z maskami (wybiera fragmenty obrazu). Potrafi retuszować obraz. Wie, czym są krzywe Béziera i rysuje je. Opracowuje samodzielnie krótki film.	Samodzielnie zapoznaje się z programami do obróbki grafiki rastrowej i wektorowej. Tworzy obrazy, wykorzystując różne możliwości programów. Korzystając z Pomocy i innych źródeł, poznaje możliwości programów graficznych. Przygotowuje grafikę do własnej strony internetowej lub prezentacji multimedialnej. Uczestniczy w konkursach dotyczących grafiki komputerowej.

Opracowywanie tekstu i prezentacji multimedialnej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna i stosuje podstawowe zasady redagowania i formatowania tekstu.	Przygotowuje poprawnie zredagowany i sformatowany tekst, dostosowując formę tekstu do jego przeznaczenia.	Tworzy dokumenty tekstowe, stosując poprawnie wszystkie poznane zasady redagowania i formatowania tekstu. Zna możliwości śledzenia zmian w dokumencie tekstowym. Potrafi śledzić zmiany w dokumencie tekstowym Zapisuje dokument tekstowy w formacie PDF.	Tworzy wielostronicowe dokumenty tekstowe, stosując poprawnie wszystkie poznane zasady pracy z tekstem wielostronicowym. Potrafi korzystać z możliwości śledzenia zmian w dokumencie, wstawiać komentarze, porównywać dokumenty.	Samodzielnie odkrywa nowe możliwości edytora tekstu, przygotowując dokumenty tekstowe. Korzysta z możliwości śledzenia zmian w dokumencie, pracując w grupie kilku osób nad jednym dokumentem.
Tworzy prezentację składającą się z kilku slajdów. Wstawia teksty i obrazy, stosuje animacje. Zna ogólne zasady tworzenia prezentacji. Potrafi dobrać tło, atrybuty czcionek, odpowiednio rozmieścić tekst i grafikę na slajdzie.	Rozróżnia sposoby przygotowania prezentacji wspomagającej wystąpienie prelegenta oraz prezentacji typu kiosk. Wie, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Zna i stosuje poprawne zasady tworzenia prezentacji wspomagającej wystąpienie prelegenta. Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Posługuje się szablonem projektu. Wyszukuje i gromadzi gotowe materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animację. Stosuje zasady prezentowania pokazu slajdów.	Zna i stosuje metody projektowania różnych rodzajów prezentacji. Potrafi zaprojektować prezentację wspomagającą własne wystąpienie. Wybiera temat, przygotowuje scenariusz, wyszukuje oraz tworzy własne materiały (teksty i obrazy, dźwięk). Komponuje układ slajdów i ich animacje. Posługuje się widokiem sortowania slajdów. Stosuje zasady referowania konkretnego tematu wspomaganej prezentacją. Zna i stosuje zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się. Dodaje efekty multimedialne:	Na gotowym, poprawnie wykonanym przykładzie przedstawia zasady tworzenia prezentacji multimedialnych. Wyjaśnia, na czym polega dostosowanie treści i formy do rodzaju prezentacji. Wyjaśnia różnice w zasadach projektowania prezentacji wspomagającej wystąpienie prelegenta, prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się. Potrafi organizować prezentację w widoku konspektu. Wykorzystuje możliwości tworzenia schematu organizacyjnego oraz możliwości tworzenia wykresów. Nagrywa narrację. Poprawnie ustawia i testuje	Przygotowuje profesjonalnie prezentację dowolnego typu. Potrafi, korzystając z prezentacji wspomagającej wystąpienie prelegenta, przeprowadzić profesjonalny pokaz. Dodaje do prezentacji materiały ze skanera, aparatu cyfrowego i kamery cyfrowej. Publikuje prezentację w Internecie. Dopasowuje parametry konwersji do formatu HTML. Przygotowuje materiały ułatwiające opracowanie prezentacji, np. wydruk miniatur slajdów wraz z notatkami. Nagrywa narrację i dodaje ją do prezentacji.

	Zna zasady przygotowania prezentacji do samodzielnego przeglądania przez odbiorcę oraz prezentacji samouruchamiającej się.	animacje, grafikę, dźwięki, podkład muzyczny. Ustawia i testuje chronometraż. Stosuje hiperłącza. Zapisuje prezentację w formacie PDF. Aktywnie współpracuje z grupą przy projektowaniu prezentacji.	chronometraż, stosuje hiperłącza. Uczestniczy w przygotowaniu w formie projektów grupowych: prezentacji do samodzielnego przeglądania przez odbiorcę oraz samouruchamiającej się. Zna sposoby umieszczania prezentacji w Internecie.	
--	---	---	---	--

4. Bazy danych

Projektowanie relacyjnej bazy danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych. Podaje przykłady programów do tworzenia baz danych. Potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja). Potrafi utworzyć prostą kwerendę, jeden formularz i raport.	Rozumie metody przetwarzania danych na przykładzie gotowej bazy danych. Określa podstawowe pojęcia (<i>rekord, pole, typ pola, relacja, klucz podstawowy</i>). Tworzy bazę danych składającą się z dwóch tabel, w każdej po kilka pól różnych typów. Projektuje formularze i raporty. Tworzy proste kwerendy wybierające. Potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów. Zna zasady przygotowania korespondencji seryjnej.	Projektuje relacyjną bazę danych (na zadany temat) składającą się z trzech tabel połączonych relacją. Omawia typy relacji w bazie danych. Zna zasady definiowania kluczy podstawowych. Projektuje formularze i raporty według wskazówek nauczyciela. Potrafi utworzyć formularz z podformularzem. Umieszcza przyciski nawigacyjne. Tworzy kwerendy wybierające. Importuje dane z tabel arkusza kalkulacyjnego i dokumentu tekstowego do tabel bazy danych. Eksportuje dane z tabel bazy danych do tabel arkusza kalkulacyjnego i do dokumentu tekstowego.	Potrafi wytłumaczyć pojęcie relacji. Projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel. Samodzielnie ustala zawartość bazy (rodzaj informacji). Zna kilka rodzajów formularzy i raportów. Umie zaprojektować samodzielnie wygląd formularza i raportu. Na formularzach umieszcza pola kombi, ogranicza wartości, wstawia (gdy jest taka potrzeba) bieżącą datę, umieszcza przyciski poleceń. Stosuje funkcje standardowe w kwerendach i standardowe operatory w kryteriach wyszukiwania. Korzysta z parametrów w kwerendzie.	Zna dokładnie wybrany program do projektowania baz danych. Potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu). Projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu. Korzysta z dodatkowej, fachowej literatury.

Tworzenie kwerend z wykorzystaniem języka SQL				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wyszukuje informacje w bazie, korzystając wyłącznie z gotowych kwerend i narzędzi wbudowanych do programu.	Tworzy samodzielnie kwerendy (proste i złożone), korzystając z wbudowanych do programu narzędzi. Wie, co to jest język SQL. Potrafi przeanalizować przykład zapytania utworzonego w języku SQL.	Zna zasady wyszukiwania informacji w bazie z wykorzystaniem języka zapytań. Zna składnię i działanie podstawowych instrukcji. Potrafi zapisać prostą kwerendę, korzystając z języka zapytań.	Potrafi zapisać złożone kwerendy, korzystając z wybranej instrukcji. Stosuje instrukcję <code>SELECT</code> i jej główne klauzule, by wybrać kolumny z tabel bazy danych. Wykorzystuje klauzulę <code>JOIN</code> do łączenia informacji z wielu tabel i kwerend oraz przedstawiania wyników jako jednego logicznego połączenia rekordów. Stosuje instrukcje <code>INSERT</code> do dopisywania rekordów i <code>UPDATE</code> do modyfikowania rekordów w bazie. Usuwa rekordy, korzystając z instrukcji <code>DELETE</code>.	Opierając się na profesjonalnej literaturze, potrafi samodzielnie zapisywać złożone kwerendy z wykorzystaniem języka zapytań SQL.

Realizacja projektu programistycznego				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje. Bierze udział w testowaniu projektu.	Zna wszystkie etapy projektowania systemów informatycznych. Uczestniczy czynnie w poszczególnych etapach projektu, wykonując zlecone zadania szczegółowe. Planuje temat projektu.	Wie, co to jest system informatyczny. Potrafi omówić zakres prac na każdym etapie. Realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania. Uczestniczy czynnie w analizie systemu informacyjnego, przygotowuje dokumentację. Przygotowuje założenia w postaci dokumentów edytora tekstu, korzystając z szablonów. Współpracuje przy projektowaniu tabel, formularzy i raportów.	Realizuje projekt zgodnie z zamierzoną organizacją pracy zespołowej i wytyczonymi wcześniej etapami projektowania. Wykonuje trudniejsze prace związane z projektowaniem systemu. Przeprowadza analizę systemu informacyjnego. Projektuje złożone kwerendy, formularze, raporty. Uczestniczy we wdrażaniu systemu informatycznego.	Potrafi pełnić funkcję koordynatora projektu. Przydziela zadania szczegółowe, dba o ich prawidłowe wykonanie, nadzoruje pracę innych, dba o dobrą atmosferę w grupie.

5. Algorytmika i programowanie – problemy zaawansowane

Wprowadzanie danych i wyprowadzanie wyników				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna i stosuje podstawowe sposoby wprowadzania danych i wyprowadzania wyników w wybranym języku programowania.	Potrafi poprawnie zadeklarować dane potrzebne do rozwiązania zadania. Wie, co określa typ danych i format danych. Potrafi odpowiednio sformatować wyprowadzane wyniki.	Tworzy własne typy danych (typ okrojony, typ wyliczeniowy) i stałe. Zna procedury i funkcje umożliwiające generowanie danych losowych. Stosuje w programach generowanie losowych danych.	Poprawnie formatuje wyprowadzane wyniki. Omawia stos jako przykład struktury danych. Wykonuje operacje na stosie. Implementuje stos z wykorzystaniem tablicy. Zna i omawia zastosowanie stosu (odwrotną notację polską – ONP).	Dobiera sposób wprowadzania danych do rozwiązywanego zadania. Potrafi sprawdzić poprawność danych wprowadzanych do programu. Definiuje własne typy danych, potrzebne do rozwiązania danego zadania. Korzystając z możliwości generowania danych losowych, tworzy własne gry komputerowe.

Wybrane typy i struktury danych, w tym dynamiczne				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna i omawia strukturę typów danych w wybranym języku programowania. Wie, jakiego typu dane zapisuje się w rekordzie. Wie, czym charakteryzują się dane typu tablicowego.	Tworzy prosty program, w którym deklaruje dane typu rekordowego (strukturalnego). Wczytuje dane do rekordu. Potrafi omówić ogólnie zasady przetwarzania plików w wybranym języku programowania.	Potrafi zastosować rekordowy typ danych do przetwarzania danych różnego rodzaju. Zna zasady przetwarzania plików w wybranym języku programowania (w językach programowania). Stosuje odpowiednie procedury i funkcje. Deklaruje zmienne typu plikowego. Korzystając z przykładów, odczytuje dane z pliku i zapisuje dane do pliku.	Wykonuje operacje na plikach w wybranym języku programowania (w językach programowania). Przetwarza pliki tekstowe. Tworzy własne programy, w których wykorzystuje przetwarzanie plików. Zna dynamiczne struktury danych i typ wskaźnikowy danych. Deklaruje zmienne typu wskaźnikowego. Stosuje w programach zmienne wskaźnikowe. Tworzy zmienne	Stosuje w programach wybrane struktury dynamiczne. Tworzy listę jednokierunkową. Tworzy i przegląda binarne drzewo poszukiwań. Pisze program wyszukujący wartości w binarnym drzewie uporządkowanym. Korzystając z dodatkowej literatury, zapoznaje się z innymi strukturami dynamicznymi, np. z listą dwukierunkową. Tworzy programy z zastosowaniem struktur

			dynamiczne. Zna wybrane struktury dynamiczne. Analizuje gotowe programy, w których zastosowano listę jednokierunkową i binarne drzewo poszukiwań; uruchamia je i testuje dla wybranych danych.	dynamicznych. Rozwiązuje zadania z matury i olimpiady informatycznej i bierze w niej udział.
--	--	--	---	--

Programowanie modułowe i obiektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wymienia modele programowania. Omawia model programowania strukturalnego. Wie, jakie są korzyści z definiowania procedur i funkcji. Potrafi zdefiniować procedury i funkcje w wybranym języku programowania. Wie, co to jest moduł.	Omawia szczegółowo modele programowania. Zna zasady programowania modułowego. Analizując przykładowe programy, tworzy własne moduły.	Wie, na czym polega programowanie obiektowe. Zna podstawowe pojęcia programowania obiektowego: <i>klasa, obiekt, pola, metody</i>.	Rozumie i potrafi zastosować typ obiektowy. Definiuje klasy. Deklaruje pola prywatne. Omawia cechy programowania obiektowego: dziedziczenie i polimorfizm. Wie, czym są metody wirtualne. Pozna je zastosowanie tych cech, analizując gotowe programy. Modyfikuje programy według wskazówek nauczyciela.	Tworzy własne programy, stosując poznane zasady programowania modułowego i obiektowego. Korzysta z fachowej literatury. Wie, na czym polega programowanie zdarzeniowe. Rozwiązuje zadania z olimpiady informatycznej i bierze w niej udział.

Wybrane algorytmy				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Zna i potrafi omówić wybrane algorytmy sortowania. Analizuje gotowe listy kroków wybranych algorytmów sortowania, wykonuje algorytmy dla wybranych danych. Potrafi omówić algorytm rozkładu liczby na czynniki pierwsze. Zna łamigłówkę Wież Hanoi. Wykonuje praktyczne ćwiczenie, odpowiednio przekłada trzy krążki.	Zna jeden z wybranych algorytmów sortowania: np. przez wstawianie. Sprawdza liczbę porównań elementów w tym algorytmie. Zna algorytmy zamiany liczb między dowolnymi systemami pozycyjnymi: obliczania wartości dziesiętnej liczby, zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie. Wie, jak sprawdzić, czy dana liczba jest liczbą doskonałą. Zna reprezentację danych numerycznych w komputerze: reprezentację binarną liczb ujemnych, reprezentację stałopozycyjną liczb, reprezentację zmiennopozycyjną liczb.	Zna algorytmy sortowania: np. przez wstawianie, przez scalanie, metodą szybką. Sprawdza liczbę porównań elementów w przypadku każdego z algorytmów. Zna algorytm przeszukiwania binarnego. Potrafi utworzyć listę kroków tego algorytmu. Zna przykładowe algorytmy na liczbach naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia wybrany algorytm numeryczny, np. obliczanie wartości pierwiastka kwadratowego. Zna sposób zapisu liczby całkowitej i rzeczywistej (zmiennoprzecinkowej).	Omawia i stosuje w zadaniach algorytmy na liczbach naturalnych: generowanie liczb pierwszych (podejście naiwne, sito Eratostenesa). Omawia algorytm szybkiego podnoszenia do potęgi i algorytmy badające własności geometryczne (np. przynależność punktu do odcinka, badanie położenia punktu względem prostej). Omawia wybrany algorytm numeryczny, np. wyznaczanie miejsca zerowego funkcji, obliczanie wartości pierwiastka kwadratowego, obliczanie pola obszaru ograniczonego. Zna właściwości arytmetyki komputerowej. Na konkretnych przykładach potrafi sprawdzić, jak zmienia się wartość błędu względnego. Tworzy program komputerowy, stosując wybrany algorytm. Wykonuje projekt programistyczny, stosując zasady pracy zespołowej.	Zapisuje wybrane algorytmy sortowania (np. przez wstawianie, przez scalanie, metodą szybką) w postaci programu komputerowego. Potrafi zapisać w języku programowania wysokiego poziomu algorytm konwersji liczb z dowolnego systemu pozycyjnego na inny. Tworzy programy komputerowe, stosując wybrane algorytmy. Rozwiązuje zadania z matury i olimpiady informatycznej.

6. Aplikacje bazodanowe dostępne za pośrednictwem Sieci

Aplikacje bazodanowe dostępne za pośrednictwem Sieci				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
Wymienia przykładowe programy do projektowania i tworzenia stron internetowych. Potrafi wymienić podstawowe elementy, z których składa się strona WWW. W stopniu podstawowym posługuje się wybranym programem do tworzenia stron. Zna podstawowe znaczniki języka HTML. Tworzy nieskomplikowaną stronę internetową. Wstawia tytuł, formatuje tekst, umieszcza obraz. Wie, na czym polega technologia dynamicznego generowania stron internetowych.	Potrafi omówić strukturę pliku w języku HTML. Zna podstawy języka znaczników HTML i potrafi wykonać prostą stronę na zadany przez nauczyciela temat. Projektuje wygląd strony. Planuje jej zawartość (teksty, rysunki, dźwięki, animacje) i umieszcza na niej ww. elementy. Zna zasady dynamicznego przetwarzania stron internetowych. Potrafi napisać proste skrypty w języku PHP.	Potrafi samodzielnie zaprojektować wygląd strony. Zna reguły poprawnego projektowania układu strony, m.in. dba o jej czytelność i przejrzystość, o poprawność redakcyjną, i merytoryczną oraz prawą umieszczanych na niej tekstów i materiałów. Zna zaawansowane możliwości języka HTML: tabele, ramki, style. Zna sposoby publikowania stron w Internecie. Tworzy skrypty w języku PHP. Wyświetla dane instrukcją echo. Stosuje kodowanie UTF-8. Stosuje zmienne i operatory. Wie, jak utworzyć witrynę internetową opartą na bazie danych, m.in. w jaki sposób utworzyć prostą księgę gości w MySQL.	Potrafi samodzielnie wykorzystać poznane funkcje języka HTML do udoskonalenia własnych stron internetowych Włącza licznik odwiedzin na stronie. Dodaje inne typowe elementy: forum, księgę gości. Zna podstawy języka skryptowego PHP (lub JavaScript). Używa go dla osiągnięcia nieskomplikowanych efektów wizualnych na stronie. Tworzy skrypty przesyłające dane za pomocą formularzy HTML. Piszze kod wyświetlający prosty formularz i odbierający dane z formularza. Tworzy witrynę internetową opartą na bazie danych. Tworzy konta użytkownika i bazy danych na serwerze MySQL. Wykonuje zapytania do bazy danych z poziomu PHP. Tworzy prostą księgę gości: tworzy tabelę na wpisy z księgi gości, dodaje wpisy do księgi gości za pomocą instrukcji INSERT, tworzy formularz dodający wpisy do bazy danych, odczytuje dane z bazy za pomocą instrukcji SELECT.	Potrafi samodzielnie zapoznać się z nowym programem do tworzenia stron internetowych. Potrafi posługiwać się językiem skryptowym PHP (lub JavaScript) do tworzenia stron dynamicznych. Tworzy samodzielnie rozbudowaną witrynę internetową opartą na bazach danych.