

Teraz bajty. Informatyka dla szkoły ponadpodstawowej. Zakres podstawowy. Klasa I

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne dla klasy I

Autor: Grażyna Koba

MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne dla klasy I, uwzględniające zmiany wynikające z zawężenia podstawy programowej dla szkoły ponadpodstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera

Komputer i urządzenia peryferyjne				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia i omawia podstawowe elementy komputera; podaje przykłady urządzeń peryferyjnych; wymienia urządzenia peryferyjne	klasyfikuje środki technologii informacyjnej ze względu na przeznaczenie; charakteryzuje przykładowe urządzenia peryferyjne; określa własności i przeznaczenie dysku twardego	potrafi określić podstawowe elementy komputera (wartości podstawowych parametrów, ich wzajemne współdziałanie); wie, czym jest RAM i BIOS, określa ich funkcje; omawia dodatkowe urządzenia pamięci masowej, m.in.: napędy optyczne, pamięci flash, pamięci taśmowe (streamery)	wymienia podstawowe układy mieszczące się na płycie głównej i charakteryzuje ich parametry; wyjaśnia, czym jest karta rozszerzenia; wie, w jakim celu tworzy się partycje na dysku twardym; wyjaśnia pojęcia: <i>partycja dyskowa, formatowanie dysku</i>	potrafi dobrać pełną konfigurację sprzętu i oprogramowania do danego zastosowania; dba o prawidłowe funkcjonowanie komputera, przeprowadzając wszystkie niezbędne testy

Systemy operacyjne i inne oprogramowanie				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, co to jest system operacyjny;</p> <p>omawia podstawowy zestaw oprogramowania, który może być zainstalowany na komputerze</p>	<p>zna funkcje systemu operacyjnego;</p> <p>wymienia popularne systemy operacyjne;</p> <p>omawia rodzaje programów komputerowych i potrafi określić ich przeznaczenie</p>	<p>podaje podstawowe cechy systemu Windows;</p> <p>charakteryzuje narzędzia TI, w tym: oprogramowanie użytkowe, języki programowania, programy narzędziowe;</p> <p>zna podstawowe typy plików</p>	<p>omawia ogólną strukturę systemu operacyjnego;</p> <p>potrafi scharakteryzować różne systemy operacyjne (Windows, Linux, Unix);</p> <p>omawia zawartość plików w zależności od ich rozszerzenia</p>	<p>omawia drogę rozwoju systemu Windows;</p> <p>dokonuje analizy porównawczej różnych systemów operacyjnych</p>

Moduł B. Wokół dokumentów komputerowych

Opracowywanie dokumentów tekstowych o rozbudowanej strukturze				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie do czego służy nagłówek i stopka dokumentu;</p> <p>rozdziela style tekstu;</p> <p>zapisuje dokument w pliku we wskazanym folderze;</p> <p>zna i stosuje podstawowe zasady redagowania i formatowania tekstu;</p> <p>zna podstawowe zasady pracy z dokumentem wielostronicowym;</p> <p>wstawia tabelę i wykonuje podstawowe operacje na komórkach tabeli;</p> <p>stosuje numerację i wypunktowanie;</p> <p>dzieli dokument na strony</p>	<p>redaguje nagłówek i stopkę, wstawia numery stron;</p> <p>wie, w jakim celu stosuje się style tekstu;</p> <p>stosuje wbudowane style nagłówkowe;</p> <p>stosuje przypisy;</p> <p>właściwie dzieli tekst na akapity;</p> <p>poprawia tekst, wykorzystując możliwości wyszukiwania i zamiany znaków oraz słowniki: ortograficzny i synonimów;</p> <p>stosuje tabulację i wcięcia;</p> <p>wykorzystuje indeksy górny i dolny oraz symbole do pisania prostych wzorów i tekstów w języku obcym;</p> <p>wie do czego służy podział dokumentu na sekcje;</p> <p>rozmieszcza tekst w kolumnach</p>	<p>redaguje inną stopkę i inny nagłówek dla stron parzystych i nieparzystych;</p> <p>stosuje różne wbudowane style tekstu;</p> <p>wie, czym są odwołania w tekście;</p> <p>umieszcza podpisy pod rysunkami, tabelami i wykresami;</p> <p>tworzy spis treści;</p> <p>zmienia ustawienia strony – wielkość marginesów, orientację strony, rozmiar papieru;</p> <p>stosuje różne typy tabulatorów, potrafi zmienić ich ustawienia w całym tekście;</p> <p>dzieli dokument na sekcje;</p> <p>pracuje z dokumentem trybie recenzji;</p> <p>korzysta z opcji śledzenia zmian, wstawia komentarze</p>	<p>tworzy spis ilustracji, tabel i wykresów;</p> <p>stosuje różne sposoby wyświetlania dokumentu;</p> <p>przygotowuje poprawnie zredagowany i sformatowany tekst, dostosowując formę tekstu do jego przeznaczenia;</p> <p>redaguje złożone wzory matematyczne korzystając z edytora równań;</p> <p>samodzielnie wyszukuje opcje menu potrzebne do rozwiązania dowolnego problemu;</p> <p>wykonuje konwersję tekstu na tabelę i odwrotnie;</p> <p>korzysta z podziału tekstu na sekcje;</p> <p>pracuje z dokumentem trybie recenzji i porównuje dokumenty;</p>	<p>samodzielnie odkrywa nowe możliwości edytora tekstu, przygotowując dokumenty tekstowe;</p> <p>tworzy dokumenty tekstowe, stosując poprawnie wszystkie poznane zasady redagowania i formatowania tekstu;</p> <p>przygotowuje profesjonalny tekst – pismo, sprawozdanie, z zachowaniem wszystkich zasad redagowania i formatowania tekstów</p>

Opracowywanie grafiki rastrowej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>z pomocą nauczyciela korzysta z wybranego programu do tworzenia grafiki rastrowej;</p> <p>wyszukuje potrzebne funkcje w menu programu;</p> <p>wymienia rodzaje grafiki komputerowej</p>	<p>zna formaty plików graficznych;</p> <p>opracowuje grafikę rastrową: stosuje warstwy i narzędzia selekcji, zmianę kontrastu i nasycenia kolorów, kadrowanie i skalowanie;</p> <p>wykonuje proste projekty w grafice wektorowej, korzystając z możliwości wstawiania Kształtów w edytorze tekstu</p>	<p>sprawne korzysta z Pomocy wbudowanej do programów w celu znalezienia szczegółowych sposobów rozwiązania danego problemu;</p> <p>podaje różnice między grafiką rastrową i wektorową;</p> <p>opracowuje grafikę rastrową: uzyskuje efekty specjalne dzięki zastosowaniu tzw. filtrów;</p> <p>tworzy proste kompozycje, korzystając z wybranego programu do tworzenia grafiki wektorowej;</p> <p>podaje różnice między grafiką 2D i 3D</p>	<p>rozumie znaczenie zapisu pliku graficznego w danym formacie – zależnie od przeznaczenia;</p> <p>omawia zalety, wady i zastosowanie wybranych formatów plików grafiki rastrowej;</p> <p>potrafi zastosować odpowiedni format pliku graficznego;</p> <p>zapisuje pliki w różnych formatach;</p> <p>opracowuje grafikę wektorową: przekształca obraz (pochyla, obraca), grupuje obiekty</p>	<p>samodzielnie zapoznaje się z możliwościami wybranego programu graficznego, przygotowując złożone projekty z różnych dziedzin</p>

Tworzenie prezentacji multimedialnej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna podstawowe typy i zasady tworzenia prezentacji multimedialnej;</p> <p>tworzy prezentację składającą się z kilku slajdów z zastosowaniem animacji niestandardowych;</p> <p>korzysta z szablonów slajdów;</p> <p>umieszcza na slajdach teksty i obrazy;</p> <p>zapisuje prezentację we wskazanym folderze docelowym;</p> <p>uruchamia pokaz slajdów</p>	<p>zna etapy tworzenia prezentacji multimedialnej;</p> <p>przygotowuje prezentację na zadany temat;</p> <p>wie, do czego służą poszczególne widoki slajdów;</p> <p>potrafi ustawić jednakowe tło dla wszystkich slajdów oraz zmienić tło dla wybranego slajdu;</p> <p>wstawia do slajdu wykresy, tabele, równania matematyczne, efekty dźwiękowe</p>	<p>potrafi właściwie zaplanować prezentację na zadany temat;</p> <p>wstawia dźwięki z plików spoza listy standardowej;</p> <p>zmienia tło, wstawia obiekty i hiperłącza;</p> <p>umieszcza przyciski akcji;</p> <p>dodaje animacje i efekty dźwiękowe do obiektów;</p> <p>dodaje narrację do prezentacji;</p> <p>prezentuje swoje prace przed klasą</p>	<p>wstawia podkład muzyczny odtwarzany podczas całej prezentacji;</p> <p>przygotowuje materiały informacyjne dla uczestników pokazu i przeprowadza pokaz;</p> <p>konwertuje przygotowaną prezentację do formatu umożliwiającego publikację w Internecie i otwiera ją lokalnie w przeglądarce internetowej</p>	<p>potrafi samodzielnie zaprojektować i przygotować multimedialną prezentację na wybrany temat, cechującą się ciekawym ujęciem zagadnienia, interesującym układem slajdów</p>

Moduł C. Wokół algorytmiki i programowania

Programowanie i rozwiązywanie problemów z wykorzystaniem komputera				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wyjaśnia co to jest algorytm; podaje przykłady sytuacji problemowych; wyjaśnia pojęcie <i>specyfikacja problemu</i>; wie, na czym polega programowanie; analizuje gotowe proste programy zapisane w wybranym języku programowania</p>	<p>wyjaśnia pojęcie algorytmu oraz zależności między problemem, algorytmem i programem; dobiera algorytm do rozwiązania problemu; formułuje specyfikację zadania; określa dane do zadania oraz wyniki; zna klasyfikację języków programowania; klasyfikuje języki programowania</p>	<p>omawia etapy rozwiązywania problemu (zadania); testuje rozwiązania; wyjaśnia, na czym polega prezentacja algorytmu w postaci programu; zna pojęcia: <i>program źródłowy, program wynikowy, implementacja, kompilacja, interpretacja, translacja</i>; porównuje gotowe, proste programy zapisane w różnych językach programowania (wizualnych i tekstowych)</p>	<p>analizuje i porównuje gotowe, proste programy zapisane w języku C++ i/lub Python; odróżnia kompilację od interpretacji; wymienia i charakteryzuje języki programowania</p>	<p>potrafi samodzielnie napisać specyfikację określonego zadania; samodzielnie określa algorytm i narzędzia właściwe do rozwiązania danego problemu</p>

Tworzenie programów w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>charakteryzuje środowisko programistyczne wybranego tekstowego języka programowania;</p> <p>analizuje gotowe proste programy zapisane w wybranym języku programowania</p>	<p>omawia etapy programowania w wybranym tekstowym języku programowania;</p> <p>wie, na czym polega iteracja;</p> <p>zna kryteria, jakie powinien spełniać poprawny program;</p> <p>wyjaśnia, co to jest iteracja</p>	<p>zna zasady stosowania zmiennych i wykonywania obliczeń w wybranym tekstowym języku programowania;</p> <p>realizuje prostą sytuację warunkową w wybranym języku programowania, stosuje proste warunki logiczne;</p> <p>sprawdza poprawność danych;</p> <p>zapisuje proste algorytmy iteracyjne w postaci listy kroków</p>	<p>wyprowadza komunikaty i wyniki na ekran w wybranym tekstowym języku programowania;</p> <p>zapisuje rozwiązanie problemu w wybranym tekstowym języku programowania;</p> <p>realizuje sytuację warunkową w wybranym języku programowania;</p> <p>stosuje złożone warunki logiczne;</p> <p>zapisuje rozwiązanie problemu iteracyjnego w postaci programu</p>	<p>zapisuje złożony algorytm w wybranym tekstowym języku programowania;</p> <p>samodzielnie pisze program realizujący algorytm z warunkami zagnieżdżonymi;</p> <p>stosuje zagnieżdżone instrukcje iteracyjne;</p> <p>uczestniczy w konkursach i olimpiadach informatycznych</p>

Moduł D. Wokół Internetu i projektów

Internet i wyszukiwanie informacji w Internecie				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wyszukuje strony WWW poprzez proste hasło wpisywane do wyszukiwarki internetowej; zna zasady nawigacji po stronie WWW, poruszając się po wybranych stronach internetowych	wie, czym są Internet i strona WWW oraz zna genezę powstania Internetu; wymienia wybrane usługi Internetowe; podaje opisy i zastosowania wyszukiwarki internetowej; szuka informacji w Internecie, konstruując złożone hasło	omawia rozwój usług internetowych, wskazując najważniejsze fakty; wyjaśnia, na czym polega przeglądanie strony internetowej; potrafi właściwie zawęzić obszar poszukiwań, aby szybko odszukać informacje; korzysta z encyklopedii i słowników w wersji elektronicznej; wyszukuje informacje zapisane w innych językach; korzysta z serwisu mapowego	omawia organizację informacji w WWW; wyjaśnia postać adresu URL; potrafi zastosować różne narzędzia do wyszukiwania informacji, usprawniając szukanie informacji; właściwie porządkuje informacje o stronach WWW; potrafi odpowiednio ocenić przydatność i wiarygodność informacji	potrafi formułować własne wnioski i spostrzeżenia dotyczące rozwoju Internetu, jego znaczenia dla różnych dziedzin gospodarki i dla własnego rozwoju; wyszukuje, gromadzi i właściwie selekcjonuje informacje, tworząc złożone projekty z różnych dziedzin

Usługi internetowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia przykładowe e-usługi, np. e-nauczanie, e-banki, e-sklepy, e-aukcje, e-podpis; wie, na czym polegają nauczanie i praca na odległość	omawia przykładowe e-usługi; korzysta z wybranych e-usług, np. e-learningu; jest świadomy istnienia zagrożeń wynikających z korzystania z e-usług	omawia zalety i wady poszczególnych e-usług; zna i stosuje zasady bezpiecznego korzystania z poszczególnych e-usług	wyjaśnia działanie e-banku; podaje metody zabezpieczeń; podaje zasady korzystania z poszczególnych e-usług; wie, czym jest podpis elektroniczny	potrafi przedstawić własne wnioski z analizy zalet i wad poszczególnych e-usług; korzystając z dodatkowych źródeł, znajduje najnowsze informacje na temat e-usług

Zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
omawia etapy tworzenia projektu grupowego; przestrzega zasad korzystania z cudzych materiałów	wyjaśnia, jak przeprowadza się debatę za i przeciw; wyjaśnia pojęcie; <i>prawo autorskie, domena publiczna</i> ; stosuje poznane metody wyszukiwania informacji	prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu	pełni rolę koordynatora projektu grupowego; przydziela zadania szczegółowe; scala dokumenty wykonane przez członków grupy	proponuje tematykę własnego projektu, samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji; koordynuje realizację projektu

Teraz bajty. Informatyka dla szkoły ponadpodstawowej. Zakres podstawowy. Klasa II

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne dla klasy II

Autor: Grażyna Koba

MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne dla klasy II, uwzględniające zmiany wynikające z zawężenia podstawy programowej dla szkoły ponadpodstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera i sieci komputerowych

Praca w sieci komputerowej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, czym jest sieć komputerów i dlaczego komputery łączą się w sieć. korzysta z podstawowych usług sieci	wymienia podstawowe klasy sieci; rozumie pojęcie logowania się do sieci; omawia podstawowe sposoby łączenia komputerów w sieć; wymienia korzyści płynące z połączenia komputerów w sieć	zna podstawy konfiguracji sieci (protokoły sieciowe, identyfikacja sieciowa); wymienia elementy niezbędne do budowy sieci; potrafi udostępniać zasoby komputera; omawia korzyści płynące z połączenia komputerów w sieć	omawia przykładowe schematy sieci: domowej i szkolnej; udostępnia zasoby w sieci	potrafi mapować zasoby komputera; wie, czym jest maska podsieci

Bezpieczeństwo i ochrona danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia sposoby ochrony danych w komputerach i sieciach komputerowych	zna zasady ochrony danych w komputerach i sieciach komputerowych	rozumie potrzebę wykonywania podstawowych operacji porządkujących zasoby komputera oraz stosowania podstawowych zasad ochrony własnych dokumentów i zasobów komputera; zna sposoby ochrony przed utratą danych	podając przykłady, dyskutuje na temat odmian złośliwego oprogramowania i oprogramowania zabezpieczającego komputer	dzieli się własnymi doświadczeniami w zakresie bezpieczeństwa i ochrony danych w komputerach

Moduł B. Wokół dokumentów komputerowych

Formuły, funkcje i wykresy w arkuszu kalkulacyjnym				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna podstawowe zastosowania arkusza kalkulacyjnego;</p> <p>potrafi zaznaczyć zadany blok komórek;</p> <p>ustawia liczbowy format danych;</p> <p>samodzielnie pisze formułę wykonującą jedno z czterech podstawowych działań arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie);</p> <p>potrafi zastosować kopiowanie i wklejanie formuł;</p> <p>tworzy prosty wykres;</p> <p>zapisuje utworzony skoroszyt we wskazanym folderze docelowym;</p> <p>zna i stosuje podstawowe funkcje arkusza kalkulacyjnego: SUMA, ŚREDNIA</p>	<p>rozdziela zasady adresowania w arkuszu kalkulacyjnym;</p> <p>stosuje adresowanie bezwzględne wtedy, gdy jest to uzasadnione;</p> <p>potrafi tworzyć formuły wykonujące bardziej zaawansowane obliczenia (potęgowanie, pierwiastkowanie, z zastosowaniem nawiasów);</p> <p>tworzy wykres składający się z wielu serii danych, dodając do niego odpowiednie opisy;</p> <p>ustawia inne formaty danych poza liczbowym;</p> <p>formatuje tabelę;</p> <p>korzysta z możliwości wstawiania funkcji;</p> <p>potrafi zastosować funkcję JEŻELI</p>	<p>poprawnie planuje tabelę w arkuszu kalkulacyjnym, umieszczając w niej dane liczbowe i opisy;</p> <p>stosuje adresowanie mieszane;</p> <p>stosuje formatowanie warunkowe tabeli arkusza;</p> <p>potrafi stosować wybrane funkcje arkusza kalkulacyjnego do rozwiązywania zadań z różnych przedmiotów;</p> <p>zna zastosowania różnych typów wykresów;</p> <p>potrafi narysować wykres wybranej funkcji matematycznej;</p> <p>tworzy wykres funkcji trygonometrycznej;</p> <p>wie, na czym polega myślenie komputacyjne</p>	<p>tworzy rozbudowane formuły z zastosowaniem funkcji arkusza kalkulacyjnego;</p> <p>stosuje wybrane funkcje arkusza kalkulacyjnego: statystyczne, logiczne, matematyczne, daty i czasu;</p> <p>panuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego;</p> <p>przedstawia dane i wyniki w tabelach i na wykresach;</p> <p>dopasowuje wygląd arkusza kalkulacyjnego po wydruku - dobiera ustawienia strony, ustawia podział stron i obszar wydruku</p>	<p>potrafi przeprowadzić analizę przykładowego problemu i opracować właściwy algorytm obliczeń;</p> <p>potrafi samodzielnie planować kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego;</p> <p>potrafi samodzielnie zrealizować rozwiązanie danego problemu;</p> <p>zna działanie i zastosowanie większości funkcji dostępnych w arkuszu kalkulacyjnym;</p> <p>samodzielnie opracowuje problemy zgodnie z kolejnymi krokami myślenia komputacyjnego</p>

Filtry oraz tabele i wykresy przestawne w arkuszu kalkulacyjnym				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wie, do czego służą filtry; potrafi, z pomocą opisu w podręczniku i nauczyciela, wyświetlić dane według prostego kryterium	korzysta z możliwości ustawiania niestandardowych filtrów do filtrowania danych w arkuszu kalkulacyjnym	potrafi stosować filtry i selekcjonować dane na podstawie zaawansowanych kryteriów; wie, do czego służą tabele przestawne; tworzy tabele i wykresy przestawne, korzystając z przykładów z podręcznika	tworzy tabele i wykresy przestawne, analizując dane zgromadzone w arkuszu kalkulacyjnym; stosuje filtry w tabeli przestawnej	odszukuje w Pomocy informacje na temat tabel i wykresów przestawnych; potrafi samodzielnie określić dane, jakie można przedstawić i poddać analizie z wykorzystaniem tablic i wykresów przestawnych

Opracowywanie grafiki wektorowej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>korzysta z podstawowych możliwości wybranego programu do tworzenia grafiki wektorowej;</p> <p>wykonuje proste projekty w grafice wektorowej, korzystając z możliwości wstawiania Kształtów w edytorze tekstu;</p> <p>tworzy i edytuje prosty rysunek w wybranym programie graficznym, korzystając z podstawowych narzędzi do rysowania figur</p>	<p>wykonuje projekty w grafice wektorowej, korzystając z możliwości wstawiania Kształtów w edytorze tekstu;</p> <p>grupuje obiekty;</p> <p>tworzy proste kompozycje, korzystając z wybranego programu do tworzenia grafiki wektorowej;</p> <p>zna podstawowe możliwości wybranego programu do edycji obrazu wektorowego;</p> <p>korzysta z narzędzi do rysowania figur i ścieżek;</p> <p>przekształca obraz – przeciąga, skaluje, obraca</p>	<p>wyszukuje potrzebne funkcje programu w menu programu graficznego;</p> <p>tworzy kompozycje z figur w grafice wektorowej;</p> <p>rysuje bryły, korzystając z możliwości rysowania przestrzennego</p>	<p>dostrzega różnice między grafiką rastrową i wektorową.</p> <p>opracowuje grafikę wektorową, rysując ciekawe kompozycje z figur, przy tym potrafi zmienić właściwości wybranego narzędzia;</p> <p>wyjaśnia, czym są ścieżki i rysuje je, używając odpowiednich narzędzi;</p> <p>wyjaśnia, czym są punkty węzłowe;</p> <p>wie, na czym polega praca z warstwami;</p> <p>wykonuje rysunki, korzystając z warstw</p>	<p>zapoznaje się z możliwościami wybranego programu graficznego, korzystając z Pomocy i innych źródeł;</p> <p>poznaje możliwości programów graficznych;</p> <p>przygotowuje złożone projekty z różnych dziedzin;</p> <p>tworzy obrazy, wykorzystując różne możliwości programu;</p> <p>przygotowuje grafikę do własnej strony internetowej lub prezentacji multimedialnej;</p> <p>uczestniczy w konkursach dotyczących grafiki komputerowej</p>

Moduł C. Wokół algorytmiki i programowania

Tworzenie programów w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>tworzy prosty program w języku wysokiego poziomu, np. wyświetlający napis na ekranie, wykonujący proste obliczenia;</p> <p>stosuje w programach zmienne i wykonuje proste obliczenia, np. oblicza sumę dwóch liczb, średnią z dwóch liczb;</p> <p>potrafi uruchomić utworzony program i wyprowadzić wyniki na ekran;</p> <p>analizuje i omawia działanie gotowych prostych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli <code>for</code> i/lub instrukcję warunkową <code>if</code></p>	<p>zna postać i działanie instrukcji iteracyjnej <code>while</code> w wybranym języku programowania i stosuje ją w tworzonych programach komputerowych;</p> <p>analizuje (wspólnie z nauczycielem) programy, w których zastosowano funkcje;</p> <p>zapisuje je, uruchamia i wyjaśnia ich działanie; wie, co to są podprogramy i zna ich zastosowanie;</p> <p>definiuje (korzystając z podręcznika) funkcje bez parametrów i stosuje je w programach;</p> <p>korzystając z podręcznika, deklaruje tablice, wczytuje i wyprowadza elementy tablicy i/lub listy na ekran</p>	<p>zna postać i działanie instrukcji iteracyjnej <code>do ... while</code> w języku C++ i stosuje ją w tworzonych programach komputerowych;</p> <p>wyjaśnia pojęcia: <i>parametr formalny</i>, <i>parametr aktualny</i>;</p> <p>definiuje funkcje z parametrami w wybranym języku wysokiego poziomu;</p> <p>zna sposób definiowania funkcji zwracającej wartość i niezwracającej wartości;</p> <p>zna pojęcia: <i>tablica</i>, <i>zmienna indeksowana</i>;</p> <p>na bazie przykładów z podręcznika, deklaruje tablicę i/lub listę, wczytuje i wyprowadza elementy tablicy i/lub listy;</p> <p>definiując odpowiednie funkcje w wybranym języku programowania</p>	<p>wyjaśnia różnicę pomiędzy funkcją zwracającą wartość i niezwracającą wartości;</p> <p>stosuje funkcje bez parametrów i z parametrami w programach;</p> <p>potrafi zastosować tablicę i/lub listę w zadaniach;</p> <p>potrafi odwoływać się do dowolnego elementu tablicy i/lub listy;</p> <p>wykonuje operacje na elementach tablicy i/lub listy;</p> <p>potrafi modyfikować program, znaleźć błędy i je poprawić</p>	<p>potrafi samodzielnie zastosować odpowiedni rodzaj instrukcji pętli w tworzonym programie;</p> <p>omawia podobieństwa i różnice w działaniu wszystkich omówionych instrukcji pętli w dwóch różnych językach programowania;</p> <p>omawia podobieństwa i różnice w definiowaniu tablic i/lub list w dwóch różnych językach programowania;</p> <p>stosuje w programach tablice i/lub listy, odpowiednio dobierając określoną strukturę danych do algorytmu;</p> <p>pisze trudniejsze programy, w których stosuje funkcje i tablice</p>

Programowanie wybranych algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje opis jednego z wybranych algorytmów, np. porządkowania metodą bąbelkową;</p> <p>potrafi wyjaśnić na czym polega, zademonstrować z użyciem pomocy dydaktycznych, przetestować</p>	<p>analizuje opisane w podręczniku gotowe programy realizujące jeden wybrany algorytm z podanych w podręczniku, np. pierwszości liczby;</p> <p>przepisuje wybrane kody programów, uruchamia programy i wyjaśnia ich działanie</p>	<p>omawia dwa wybrane algorytmy, np. obliczania elementów ciągu liczb Fibonacciego i porządkowania przez wstawianie, demonstruje je przy użyciu pomocy dydaktycznych;</p> <p>korzystając z opisu w podręczniku zapisuje je w wybranym języku programowania i potrafi wyjaśnić ich działanie</p>	<p>omawia algorytmy badania pierwszości liczby i obliczania elementów ciągu liczb Fibonacciego, porządkowania metodą bąbelkową i przez wstawianie;</p> <p>zapisuje je w postaci programów, korzystając z podręcznika, rozumie działanie tych programów;</p> <p>stosuje funkcje i tablice w zapisie w/w algorytmów w postaci programów</p>	<p>samodzielnie zapisuje w postaci programów algorytmy badania pierwszości liczby i obliczania elementów ciągu liczb Fibonacciego, porządkowania metodą bąbelkową i przez wstawianie;</p> <p>definiuje odpowiednie i funkcje;</p> <p>rozwiązuje przykładowe zadania z konkursów informatycznych;</p> <p>bierze udział w konkursach</p>

Moduł D. Wokół Internetu i projektów

Wybrane przepisy prawa				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna podstawowe przepisy prawa dotyczące korzystania z cudzych materiałów i stosuje je w praktyce;</p> <p>zna podstawowe zasady korzystania z programów komputerowych;</p> <p>rozumie konieczność posiadania licencji na programy komputerowe;</p> <p>jest świadomy istnienia przestępstw komputerowych</p>	<p>wie, co jest przedmiotem prawa autorskiego i co jemu nie podlega;</p> <p>zna pojęcie licencji;</p> <p>wymienia przykładowe rodzaje licencji;</p> <p>wymienia przykładowe rodzaje przestępstw komputerowych</p>	<p>wyjaśnia wybrane przepisy prawa autorskiego, m.in.: „dozwolony użytek utworów”, zasady korzystania z cudzego utworu bez pytania o zgodę, ochrona wizerunku;</p> <p>omawia przykładowe rodzaje licencji na programy komputerowe;</p> <p>omawia wybrane przykłady przestępstw komputerowych</p>	<p>potrafi uzasadnić zastosowanie wybranego przepisu prawa w konkretnym przypadku;</p> <p>podaje przykłady łamania wybranych przepisów prawa;</p> <p>omawia różnice pomiędzy różnymi rodzajami licencji;</p> <p>sprawdza, na podstawie jakiej licencji jest rozpowszechniany dany program i wyjaśnia zasady tej licencji</p>	<p>potrafi samodzielnie interpretować ważniejsze przepisy prawa autorskiego dotyczące korzystania z różnych źródeł informacji i ochrony programów komputerowych;</p> <p>wyszukuje dodatkowe informacje na temat przestępstw komputerowych</p>

Komunikacja i wymiana informacji w Internecie				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wymienia podstawowe zasady pisania listów elektronicznych;</p> <p>podaje przykładowe sposoby komunikacji i wymiany informacji z wykorzystaniem Internetu;</p> <p>zna i stosuje zasady netykiety;</p> <p>podaje przynajmniej dwie korzyści wynikające z rozwoju technologii informacyjno-komunikacyjnych (TIK);</p> <p>jest świadomy istnienia zagrożeń wynikających z rozwoju TIK</p>	<p>poprawnie redaguje listy elektroniczne, dbając o ich formę i treść;</p> <p>omawia wybrane formy komunikacji i wymiany informacji;</p> <p>zna i stosuje zasady netykiety dotyczące form komunikacji, z których korzysta;</p> <p>wyjaśnia, na czym polega komunikacja w czasie rzeczywistym;</p> <p>wie, czym jest chmura obliczeniowa;</p> <p>wymienia przynajmniej dwie możliwości korzystania z chmury;</p> <p>podaje zalety korzystania z komunikacji za pomocą Internetu;</p> <p>wymienia podstawowe zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnej</p>	<p>rozdziela poszczególne formy komunikowania się przez Internet;</p> <p>rozdziela poszczególne sposoby komunikacji i wymiany informacji;</p> <p>omawia działanie poczty elektronicznej;</p> <p>zna i stosuje zasady netykiety dotyczące wszystkich form komunikacji;</p> <p>zna i omawia możliwości korzystania z chmury obliczeniowej;</p> <p>omawia szczegółowo zagrożenia wynikające z rozwoju technologii informacyjno-komunikacyjnej (m.in.: cyberprzemoc, anonimowość kontaktów, uzależnienie od komputera);</p> <p>wie, czym jest fake news;</p> <p>omawia korzyści i zagrożenia dotyczące korzystania z różnych form komunikacji i wymiany informacji z wykorzystaniem Internetu</p>	<p>potrafi dokonać analizy porównawczej różnych form komunikacji i wymiany informacji, podając opis poszczególnych form i niezbędne wymagania;</p> <p>porównuje metody dostępu do poczty elektronicznej;</p> <p>wyjaśnia, na czym polega korzystanie z oprogramowania w chmurze i na czym polega współdzielenie dokumentów umieszczonych w chmurze;</p> <p>wyjaśnia, jakie korzyści daje rozwój technologii informacyjno-komunikacyjnej osobom z niepełnosprawnościami;</p> <p>bierze aktywny udział w debacie na temat szans i zagrożeń wynikających z rozwoju TIK.</p>	<p>samodzielnie wyszukuje w różnych źródłach informacje na temat nowoczesnych możliwości korzystania z Internetu, np. za pomocą urządzeń mobilnych;</p> <p>wyszukuje w Internecie informacji na temat łączenia się różnych urządzeń znajdujących się w domach czy samochodach z chmurą;</p> <p>potrafi samodzielnie ocenić znaczenie technologii informacyjno-komunikacyjnej w komunikacji i wymianie informacji oraz zna najnowsze osiągnięcia w tej dziedzinie</p>

Tworzenie stron internetowych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, w jaki sposób zbudowane są strony WWW; zna najważniejsze narzędzia do tworzenia stron internetowych;</p> <p>wie, na czym polega tworzenie strony internetowej; zapoznaje się z przykładowym źródłem strony internetowej, przeglądając strukturę pliku</p>	<p>potrafi przygotować prostą stronę internetową, używając dowolnego edytora tekstu;</p> <p>umie tworzyć akapity i wymuszać podział wiersza, dodawać nagłówki do tekstu, zmieniać krój i wielkość czcionki;</p> <p>wie, jak wstawiać linie rozdzielające;</p> <p>umie wstawiać hiperłącza, korzystać z kotwic;</p> <p>rozumie strukturę plików HTML;</p> <p>podaje przykłady stosowania stylów CSS</p>	<p>potrafi tworzyć proste strony w języku HTML, używając edytora tekstowego;</p> <p>zna funkcje i zastosowanie najważniejszych znaczników HTML;</p> <p>potrafi wstawiać grafikę do utworzonych stron;</p> <p>umie tworzyć listy wypunktowane i numerowane;</p> <p>zna nazewnictwo kolorów;</p> <p>zna najczęściej wykorzystywane atrybuty CSS i sposoby określania ich wartości;</p> <p>omawia sposoby publikowania strony w Internecie</p>	<p>umie wstawiać tabele do tworzonych stron i je formatować;</p> <p>koduje polskie znaki;</p> <p>tworzy podstronę;</p> <p>umieszcza łącza hipertekstowe, m.in. łącząc stronę główną z podstroną;</p> <p>stosuje kolory;</p> <p>analizuje wady i zalety różnych sposobów publikowania i promowania stron w Internecie;</p> <p>przygotowuje stronę do publikacji w Internecie i ją publikuje</p>	<p>zna zagadnienia dotyczące promowania stron WWW;</p> <p>potrafi stworzyć własny, rozbudowany serwis WWW i przygotować go w taki sposób, żeby wyglądał estetycznie i zachęcał do odwiedzin;</p> <p>zna większość znaczników HTML</p>

Zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
omawia etapy tworzenia projektu grupowego; wykonuje proste zadania szczegółowe z projektu grupowego; przestrzega zasad korzystania z cudzych materiałów	wyjaśnia, jak przeprowadza się debatę za i przeciw; wyjaśnia pojęcie; <i>prawo autorskie, domena publiczna</i> ; stosuje poznane metody wyszukiwania informacji	prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu	pełni rolę koordynatora projektu grupowego; przydziela zadania szczegółowe; scala dokumenty wykonane przez członków grupy	proponuje tematykę własnego projektu; samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji; koordynuje realizację projektu

Teraz bajty. Informatyka dla szkoły ponadpodstawowej. Zakres podstawowy. Klasa III

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne dla klasy III

Autor: Grażyna Koba

MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne dla klasy III, uwzględniające zmiany wynikające z zawężenia podstawy programowej dla szkoły ponadpodstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera i sieci komputerowych

Reprezentacja danych w komputerze				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcie systemu pozycyjnego;</p> <p>wie, co to jest system binarny;</p> <p>analizuje gotowy przykład z podręcznika obliczający wartość dziesiętną liczby zapisanej w systemie dwójkowym i na tej podstawie wykonuje podobne ćwiczenie</p>	<p>definiuje pojęcie systemu pozycyjnego;</p> <p>wie na czym polega działanie procesora;</p> <p>potrafi dokonać konwersji liczby między systemem dziesiętnym a dwójkowym oraz dwójkowym a dziesiętnym</p>	<p>zna podwójne nazwy pozycyjnych systemów liczbowych tj. dwójkowy – binarny, dziesiętny – decymalny, szesnastkowy - heksadecymalny;</p> <p>wyjaśnia, w jaki sposób procesor dodaje liczby;</p> <p>dokonuje konwersji liczb między systemem dziesiętnym i szesnastkowym oraz szesnastkowym i dziesiętnym</p>	<p>zna elementy uproszczonego modelu komputera zgodny z ideą von Neumanna;</p> <p>dokonuje konwersji liczb między systemem szesnastkowym i binarnym</p>	<p>potrafi narysować uproszczony model komputera zgodny z ideą von Neumanna;</p> <p>samodzielnie potrafi dokonać zamiany między trzema systemami pozycyjnymi (dwójkowym, dziesiętnym i szesnastkowym) w jednym zadaniu</p>

Wybrane urządzenia cyfrowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia urządzenia cyfrowe wykorzystywane w szkole podczas zajęć (np. drukarka, drukarka 3D, tablica interaktywna, monitor, kamera); podaje nazwy urządzeń cyfrowych wykorzystywane w domu i poza nim (np. płyta grzejna, okap kuchenny, odtwarzacze audio, system multiroom, system nawigacji, smartwatch)	omawia funkcje poznanych urządzeń używanych w szkole oraz w domu i poza nim; potrafi zaprezentować w klasie wybrane urządzenie cyfrowe i omówić jego działanie	korzysta z wyszukiwarki internetowej celem opracowania informacji na temat wybranego urządzenia cyfrowego; z pomocą nauczyciela przygotowuje model 3D do druku 3D, korzystając z odpowiedniego oprogramowania; uruchamia drukarkę 3D i wykonuje przykładowy wydruk (lub omawia sposób drukowania – w przypadku braku drukarki w szkole)	objaśnia funkcje poznanych urządzeń używanych w domu i poza nim; zna podstawowe możliwościami oprogramowania towarzyszącego wybranemu urządzeniu, np. drukarce 3D i przygotowuje model 3D do wydruku; samodzielnie potrafi uruchomić drukarkę 3D i wykonać przykładowy wydruk (w przypadku, gdy szkoła ma takie możliwości)	wymienia parametry techniczne urządzeń cyfrowych podanych w specyfikacji technicznej; potrafi posługiwać się instrukcją obsługi urządzeń cyfrowych i poznawać samodzielnie możliwości towarzyszącego im oprogramowania

Moduł B. Wokół dokumentów komputerowych

Tworzenie baz danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wyjaśnia, na czym polega przetwarzanie danych;</p> <p>definiuje pojęcie baza danych;</p> <p>na przykładzie gotowego pliku bazy danych potrafi omówić jej strukturę – określić, jakie informacje są w niej pamiętane</p>	<p>podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych;</p> <p>wyjaśnia pojęcia: <i>baza danych</i>, <i>rekord</i> i <i>pole</i>;</p> <p>rozumie organizację danych w relacyjnych bazach danych;</p> <p>potrafi przygotować schemat prostej relacyjnej bazy danych;</p> <p>tworzy prostą bazę danych, składającą się z dwóch tabel;</p> <p>planuje zawartość tabel;</p> <p>stosuje zasady tworzenia tabel;</p> <p>wie do czego służy import danych w bazie danych</p>	<p>omawia etapy przygotowania bazy danych;</p> <p>określa odpowiednio typy danych;</p> <p>rozumie pojęcia relacji i klucza podstawowego;</p> <p>przygotowuje projekt formularza i raportu;</p> <p>tworzy tabele i korzysta z Widoku projektu;</p> <p>umie zaimportować dane z arkusza kalkulacyjnego do bazy danych</p>	<p>rozumie, co oznacza przetwarzanie danych w bazach danych;</p> <p>definiuje relacje między tabelami;</p> <p>potrafi uzasadnić, dlaczego warto umieszczać dane w kilku tabelach połączonych relacją;</p> <p>podczas rozwiązywania nowego problemu korzysta z doświadczeń zdobytych przy rozwiązaniu innego, podobnego problemu</p>	<p>analizuje problemy występujące w utworzonej bazie danych i znajduje ich rozwiązanie;</p> <p>samodzielnie przygotowuje projekt bazy danych (składającej z trzech tabel) i potrafi ją wykonać w programie do tworzenia baz danych</p>

Wykonywanie podstawowych operacji na relacyjnej bazie danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna zastosowanie filtrów do wyszukiwania danych; potrafi wyświetlić wynik gotowego zapytania i omówić, czego zapytanie dotyczy; omawia zastosowanie korespondencji seryjnej	definiuje pojęcie kwerendy; tworzy kwerendę wybierającą w Widoku projektu ; zna sposób przygotowania korespondencji seryjnej z wykorzystaniem danych z bazy danych	modyfikuje gotowe zapytania; tworzy kwerendę parametryczną; umieszcza w korespondencji seryjnej pola z tabeli bazy danych; korzysta z gotowych szablonów listów seryjnych	samodzielnie modyfikuje i tworzy kwerendy; wie, jaka jest korelacja między edytorem tekstu a bazą danych podczas tworzenia korespondencji seryjnej; w edytorze tekstu przygotowuje listy seryjne i etykiety adresowe, korzystając z danych zapisanych w bazie danych	potrafi zaprojektować samodzielnie relacyjną bazę danych (składającą się z trzech tabel), ustala typy pól; potrafi budować złożone kwerendy z dwóch lub więcej tabel połączonych; opracowuje własny szablon listu seryjnego

Projektowanie modeli dwuwymiarowych i trójwymiarowych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wyjaśnia różnice między grafiką 2D i 3D;</p> <p>wymienia nazwy programów do tworzenia grafiki 3D;</p> <p>planuje kroki wykonania projektu ogrodu;</p> <p>korzysta z podstawowych narzędzi programu SketchUp</p>	<p>potrafi zainstalować i skonfigurować program SketchUp;</p> <p>realizuje przekształcenie modelu 2D w 3D;</p> <p>korzysta z dodatkowych narzędzi programu SketchUp;</p> <p>umieszcza gotowe elementy z biblioteki</p>	<p>tworzy obiekty z zachowaniem odpowiedniej skali;</p> <p>przekształca pliki graficzne;</p> <p>korzysta z dodatkowych narzędzi programu SketchUp jak Offset, Pull/Push, Orbit;</p> <p>przekształca pliki graficzne, uwzględniając wielkość i jakość obrazów</p>	<p>wykonuje modele trójwymiarowe z zachowaniem skali i wytycznych;</p> <p>tworzy modele, wykorzystując różne możliwości programu</p>	<p>zapoznaje się z możliwościami wybranego programu graficznego, korzystając z Pomocy i innych źródeł;</p> <p>przygotowuje złożone projekty z różnych dziedzin;</p> <p>uczestniczy w konkursach dotyczących grafiki komputerowej</p>

Moduł C. Wokół algorytmiki i programowania

Algorytmy na tekstach w językach C++ i Python				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, że w edytorach tekstu wykorzystywane są algorytmy na tekstach – pokazuje przykłady wyszukiwania znaków w tekście, porównywania tekstów;</p> <p>potrafi omówić, posługując się przykładami i pomocami dydaktycznymi, wybrany algorytm na tekstach;</p> <p>analizuje gotowy program wykorzystujący dane tekstowe i objaśnia stosowanie zmiennych tekstowych;</p> <p>testuje gotowe programy dla różnych danych</p>	<p>planuje kolejne kroki rozwiązania problemu porównywania tekstów, szukając rozwiązania;</p> <p>deklaruje zmienne typu tekstowego <code>char</code> i <code>string</code>;</p> <p>realizuje algorytm porównania dwóch tekstów;</p> <p>wie, na czym polega algorytm szukania wzorca w tekście;</p> <p>wie, jak odwoływać się do pojedynczego znaku łańcucha,</p> <p>wie, jak wyznaczyć długość łańcucha – potrafi zastosować funkcję <code>length()</code> oraz <code>strlen()</code> (C++) i funkcję <code>len()</code> (Python);</p> <p>analizuje funkcję realizującą np. algorytm porównywania tekstów i omawia działanie funkcji;</p> <p>tworzy program wykorzystujący tę funkcję</p>	<p>przeprowadza analizę rozwiązania algorytmu szukania znaku w tekście;</p> <p>szuka wystąpień wzorca w tekście metodą naiwną, analizując i uzupełniając kolejne kroki algorytmu z wykorzystaniem podręcznika;</p> <p>definiuje pojęcie konkatencji;</p> <p>formułuje treść zadania do przedstawionego kodu źródłowego;</p> <p>definiuje funkcję szukającą i zliczającą wystąpienia znaków lub ciągów znaków w tekście;</p> <p>korzysta z instrukcji <code>if...elif</code> (Python) do utworzenia menu programu</p>	<p>opracowuje program zliczający wystąpienie znaku w tekście;</p> <p>korzysta z funkcji wyboru <code>switch()</code> (C++) do utworzenia menu programu;</p> <p>umieszcza w kodzie źródłowym funkcję <code>getline()</code> do wprowadzania napisów składających się z wyrazów oddzielonych spacjami</p>	<p>tworzy zaawansowane programy wykorzystujące dane tekstowe i poznane funkcje;</p> <p>potrafi samodzielnie utworzyć algorytm i program realizujący porównywanie tekstów;</p> <p>tworzy rozbudowane menu wyboru z wykorzystaniem funkcji <code>switch()</code> (C++);</p> <p>tworzy rozbudowane menu wyboru z wykorzystaniem z instrukcji <code>if ... elif</code> (Python)</p>

Algorytmy szyfrowania w językach C++ i Python				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym jest szyfrowanie danych i w jakim celu się je stosuje;</p> <p>potrafi, korzystając z przykładu z podręcznika, przeanalizować prosty przykład szyfrowania</p>	<p>zna pojęcia: <i>szyfr</i>, <i>szyfrowanie</i>, <i>deszyfrowanie</i>;</p> <p>omawia schemat procesu szyfrowania i deszyfrowania;</p> <p>zna przynajmniej jeden algorytm szyfrowania danych – szyfr Cezara;</p> <p>korzystając z podręcznika szyfruje i deszyfruje wiadomość, korzystając z szyfru Cezara;</p> <p>analizuje i rozumie działanie funkcji (podanych w podręczniku) realizujących wybrany algorytm szyfrowania</p>	<p>wie, czym zajmuje się kryptologia i kryptoanaliza;</p> <p>stosuje szyfrowanie tekstu metodą Cezara;</p> <p>tworzy program realizujący algorytm szyfrowania szyfrem Cezara;</p> <p>formułuje algorytm deszyfrowania podstawieniowego i pisze funkcję realizującą ten algorytm</p>	<p>pisze program realizujący algorytm szyfrowania podstawieniowego (opracowuje odpowiednie funkcje pomocnicze);</p> <p>pisze funkcję deszyfrowania podstawieniowego;</p> <p>objaśnia szyfrowanie symetryczne i asymetryczne</p>	<p>omawia dziedziny gospodarki, w których wykorzystywane jest szyfrowanie danych;</p> <p>wyszukuje dodatkowe informacje na temat szyfrowania danych;</p> <p>omawia, czym się zajmuje stenografia, samodzielnie wyszukując informacje na ten temat</p>

Algorytmy na liczbach w językach C++ i Python				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>przeprowadza konwersję liczby z systemu dwójkowego na dziesiętny oraz z dziesiętnego na dwójkowy (bazując na przykładach z tematu A1); wyjaśnia, na czym polega wyznaczanie NWD i NWW</p>	<p>korzystając z fragmentu programów z podręcznika, pisze programy w wybranym języku programowania (C++, Python): obliczające wartość dziesiętną liczby dwójkowej i wyznaczające rozwinięcie dwójkowe liczby dziesiętnej; testuje programy dla różnych danych; opracowuje funkcję w języku programowania wyznaczającą NWD; opracowuje funkcję w języku programowania wyznaczającą NWW</p>	<p>analizuje algorytm zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie wyróżniając podproblemy; testuje działanie algorytmów: obliczania wartości dziesiętnej liczby zapisanej w dowolnym systemie i zapisywania liczby dziesiętnej w systemie liczbowym o określonej podstawie, korzystając z odpowiednich list kroków (z podręcznika); wie, jak można reprezentować ułamek zwykły w języku C++ (deklaruje rekordy za pomocą słowa kluczowego <code>struct</code>) lub w języku Python; zna wzory na obliczanie sumy ułamków zwykłych oraz mnożenie i dzielenie ułamków zwykłych</p>	<p>na podstawie list kroków (podanych w podręczniku) tworzy programy: obliczania wartości dziesiętnej liczby zapisanej w dowolnym systemie i zapisujący liczbę dziesiętną w systemie liczbowym o określonej podstawie; potrafi zastosować schemat Hornera do obliczenia wartości wielomianu; analizuje funkcje skracającą ułamki zwykłe i dodająca ułamki zwykłe oraz stosuje je do napisania programów: skracającego ułamki zwykłe oraz dodającego ułamki zwykłe; definiuje funkcje: odejmującą, mnożącą oraz dzielącą ułamki zwykłe; wywołuje funkcje w programie głównym i testuje programy dla różnych danych</p>	<p>pisze program wykonujący konwersję liczb zapisanych w dowolnych podstawach; bierze udział w konkursach informatycznych i/lub olimpiadzie informatycznej</p>

Moduł D. Wokół Internetu i projektów

Elementy robotyki – projekty				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie czym zajmuje się robotyka;</p> <p>potrafi scharakteryzować funkcje mikrokontrolera Arduino;</p> <p>zna etapy tworzenia projektu grupowego i wykonuje proste zadania szczegółowe z projektu grupowego</p>	<p>zna podstawową strukturę programu dla mikrokontrolera Arduino;</p> <p>potrafi nazwać moduły składające się na system sterowania nawadnianiem ogrodu;</p> <p>wykonuje przydzielone zadania szczegółowe</p>	<p>na podstawie wytycznych wykonuje układ elektroniczny oraz pisze program obsługujący system „podlewania ogrodu”;</p> <p>prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu;</p> <p>prezentuje efekty wspólnej pracy</p>	<p>rozszerza system „podlewania ogrodu”, zgodnie z opisem w podręczniku;</p> <p>konstruuje robota, stosując elementy elektroniczne oraz programuje jego funkcje;</p> <p>pełni rolę koordynatora projektu grupowego, m.in.: określa i przydziela zadania szczegółowe;</p> <p>scala elementy projektu wykonane przez członków grupy w jeden projekt</p>	<p>samodzielnie tworzy programy sterujące mikrokontrolerem;</p> <p>wykonuje wybrany inny system, np. system „Wycisz radio” po otwarciu drzwi do pokoju;</p> <p>proponuje tematykę własnego projektu, samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji</p>

Więcej na temat tworzenia stron internetowych – projekt				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym są blogi; potrafi znaleźć blog o wybranej tematyce;</p> <p>zna najważniejsze narzędzia do tworzenia stron internetowych;</p> <p>wie na czym polega tworzenie strony internetowej;</p> <p>wykonuje proste zadania szczegółowe z projektu grupowego;</p> <p>przestrzega zasad korzystania z cudzych materiałów</p>	<p>wie, jak założyć blog;</p> <p>wie, czym jest system zarządzania treścią;</p> <p>omawia etapy tworzenia strony internetowej;</p> <p>uczestniczy w przygotowaniu projektu graficznego strony internetowej;</p> <p>wie, jak ustalić tło strony internetowej;</p> <p>wykonuje przydzielone zadania szczegółowe;</p> <p>prezentuje efekty wspólnej pracy</p>	<p>potrafi założyć prosty blog o wybranej tematyce;</p> <p>korzysta z szablonów do tworzenia stron;</p> <p>przygotowuje projekt graficzny strony internetowej;</p> <p>potrafi ustawić listy w wierszach i kolumnach na stronie internetowej;</p> <p>omawia wybrane atrybuty CSS i podaje przykłady ich stosowania;</p> <p>prawidłowo zapisuje, przechowuje i udostępnia dokumenty potrzebne do realizacji projektu</p>	<p>tworzy stronę internetową wzbogaconą o dodatkowe elementy;</p> <p>potrafi tworzyć przyciski na stronie internetowej z elementów listy poprzez dodanie obramowania i innych atrybutów;</p> <p>stosuje wybrane atrybuty CSS;</p> <p>testuje stronę internetową, określając czy, projekt został wykonany zgodnie ze specyfikacją;</p> <p>pełni rolę koordynatora projektu grupowego;</p> <p>przydziela zadania szczegółowe;</p> <p>scala dokumenty wykonane przez członków grupy</p>	<p>wykonuje samodzielnie projekt graficzny strony internetowej na wybrany przez siebie temat;</p> <p>tworzy własną stronę internetową wzbogaconą o dodatkowe elementy, w tym tabelami, listami, elementy dynamiczne;</p> <p>posługuje się arkuszem stylów;</p> <p>publikuje stronę w Internecie;</p> <p>proponuje tematykę własnego projektu, samodzielnie wyznacza zadania szczegółowe i sposób ich realizacji</p>

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne

opracowany na podstawie materiału edukacyjnego:

Grażyna Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres rozszerzony. Materiał edukacyjny. Część 1,*

dostępnego na stronie internetowej *ir.migra.pl*

Autor: Grażyna Koba
MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne na pierwsze 60 godzin lekcyjnych realizacji materiału edukacyjnego do zakresu rozszerzonego, zgodnie z opublikowaną odpowiednio propozycją rozkładu materiału.

Wymagania uwzględniają zmiany wynikające z zawężenia podstawy programowej dla szkoły ponadpodstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera i sieci komputerowych

Kompresja danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
rozumie, na czym polega kompresja danych i w jakim celu się ją stosuje; zna wzór na obliczenie współczynnika kompresji	wymienia i omawia rodzaje kompresji – bezstratną i stratną; potrafi obliczyć współczynnik kompresji; zna pojęcie <i>przepływności</i>	omawia algorytmy kompresji stratnej; omawia algorytmy kompresji bezstratnej (statyczne, słownikowe)	dokonyuje kompresji informacji, m.in. stosuje w przykładowych ćwiczeniach algorytmy kompresji bezstratnej; objaśnia różnice między kompresją stratną i bezstratną tekstów, obrazów, dźwięków i filmów	wykonuje samodzielnie zadania dla zainteresowanych

Sztuczna inteligencja				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie czym jest sztuczna inteligencja; potrafi wymienić obszary zastosowania sztucznej inteligencji (SI);</p>	<p>umie przeanalizować prosty system SI w arkuszu kalkulacyjnym; potrafi wytłumaczyć schemat sieci neuronowej; konstruuje zapytania w systemie ChatGPT (lub podobnym); tworzy proste grafiki w systemie tworzenia obrazu</p>	<p>konstruuje rozwinięte zapytania w systemie ChatGPT (lub podobnym); samodzielnie korzysta z systemów tworzenia obrazów</p>	<p>wskazuje zalety i wady z korzystania z systemów wykorzystujących SI; korzysta z różnych systemów tworzenia obrazów</p>	<p>wyszukuje w Internecie inne systemy wykorzystujące sztuczną inteligencję oraz ocenia ich przydatność w różnych dziedzinach; prowadzi dyskusje o zastosowaniu SI i potrafi prawidłowo uargumentować swoje zdanie</p>

Moduł B. Wokół dokumentów komputerowych

Tworzenie i edytowanie obrazów rastrowych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna podstawowe możliwości programu do edycji obrazu rastrowego (programu GIMP); potrafi utworzyć prosty rysunek, stosując różne narzędzia malarskie i korekcyjne oraz wybrać odpowiedni tryb ich pracy	wykonuje operacje na obszarach selekcji, m.in.: wycinanie, kopiowanie, kadrowanie, obrysowywanie; stosuje poznane możliwości programu GIMP do edycji zdjęć	zna przeznaczenie i odpowiednio stosuje wybrane klawisze modyfikujące (i kombinacje klawiszy); potrafi wykonać proste modyfikacje barw: zmianę jasności, odwracanie kolorów, zmianę nasycenia, odcieni kolorów	zna pojęcia: <i>histogram</i> , <i>krzywa barw</i> ; tworzy własny pędzel; wykorzystuje tryb Kolor ; operuje na kanałach barw; używa krzywej jasności i funkcji histogramu; korzysta z narzędzia klonowania; w razie konieczności wyszukuje informacje w Pomocy	samodzielnie zapoznaje się z dodatkowymi możliwościami programu GIMP, w tym opcjami tworzenia pędzli; rozwiązuje trudniejsze zadanie, korzystając z narzędzia klonowania i innych; wyszukuje w Internecie informacje na temat tworzenia zdjęć panoramicznych; uczestniczy w konkursach dotyczących grafiki komputerowej

Praca z warstwami obrazu i animacje				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
poprawia jakość zdjęcia, stosując wybrane filtry; wie, na czym polega praca z warstwami	wykonuje przekształcenia obrazu (obroty, odbicia), tworząc obrazy w grafice rastrowej; stosuje filtry, m.in. poprawia ostrość obrazu, wykonuje proste fotomontaże, korzystając z warstw	stosuje filtry, m.in. wykonuje efekt zamglenia, usuwa szumy; wykonuje fotomontaże, korzystając z kilku warstw; wie, czym jest oderwane zaznaczenie i jak zakotwiczyć warstwę; przygotowuje animacje składające się z kilkunastu klatek, stosując pracę na warstwach; zapisuje animację w formacie GIF	samodzielnie tworzy rysunki, powiela warstwy i stosuje możliwości przekształceń, aby utworzyć kolejną klatkę animacji; wykorzystując dodatkowe możliwości programu GIMP, rozwiązuje różne zadania, edytując obrazy, tworząc fotomontaże korzystając z poznanych możliwości programu GIMP, projektuje dwuwymiarową wizualizację lub animację	wyszukuje w Internecie dodatkowe informacje na temat korzystania z filtrów w programie GIMP; wymyśla temat fotomontażu i przygotowuje animacje według własnych pomysłów, korzystając z różnych możliwości programu do tworzenia animacji; projektuje dwuwymiarową wizualizację lub animację, wykorzystując dodatkowe możliwości programu GIMP

Moduł C. Wokół algorytmiki i programowania

Prezentacja algorytmu liniowego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
zna sposoby prezentacji algorytmów; testuje działanie algorytmu liniowego zapisanego w postaci listy kroków; zapisuje prosty algorytm liniowy w wybranym języku programowania	zna pojęcie specyfikacji zadania i potrafi zapisać specyfikację zadania; zna i stosuje zasady tworzenia listy kroków; pisze programy komputerowe realizujące dane algorytmy na podstawie ich list kroków	przedstawia dokładną specyfikację dowolnego zadania; objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych; realizuje przykładowy algorytm liniowy w wybranym języku programowania na podstawie ich list kroków; testuje program dla różnych danych	potrafi przeprowadzić szczegółową analizę poprawności listy kroków; analizuje działanie algorytmu dla przykładowych danych; potrafi wskazać i poprawić błędy w liście kroków; samodzielnie pisze program realizujący algorytm liniowy	zapisuje specyfikację zadania i przedstawia w postaci listy kroków rozwiązanie trudniejszego zadania; w wybranym języku programowania (C++ lub Python) pisze trudniejsze programy realizujące algorytmy przedstawione w postaci list kroków; uczestniczy w konkursach/olimpiadach informatycznych

Prezentacja algorytmu z warunkami w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>określa sytuacje warunkowe; podaje przykłady zadań, w których występują sytuacje warunkowe; potrafi zapisać w liście kroków nietrudny algorytm z warunkiem prostym</p>	<p>analizuje działanie algorytmu z warunkami; zna i stosuje zasady tworzenia listy kroków algorytmu z warunkami; pisze program w wybranym języku programowania realizujący algorytm z warunkami prostymi; zna i stosuje instrukcję warunkową</p>	<p>przedstawia algorytm z warunkami w postaci listy kroków; analizuje listę kroków i schemat blokowy algorytmu z warunkami, testując go dla wybranych danych; potrafi zapisać warunek złożony; korzystając z przykładu, zapisuje w postaci programu algorytm z warunkami złożonymi; zna i omawia warunek istnienia trójkąta</p>	<p>zapisuje algorytmy z pętlą zagnieżdżoną; testuje działanie algorytmu z warunkami zagnieżdżonymi zapisanego w postaci listy kroków na wybranych przykładach danych; zapisuje w postaci programu algorytm z warunkami zagnieżdżonymi</p>	<p>wymyśla samodzielnie problem z warunkami zagnieżdżonymi, zapisuje jego specyfikację, listę kroków i program w wybranym języku programowania; korzystając z dodatkowych źródeł, znajduje inny, niż podany w podręczniku, sposób sprawdzenia, czy z danych trzech odcinków można zbudować trójkąt i zapisuje ten algorytm w postaci programu komputerowego; w wybranym języku programowania pisze trudniejsze programy realizujące algorytmy z warunkami (w tym zagnieżdżonymi) przedstawione w postaci list kroków</p>

Prezentacja algorytmu iteracyjnego w wybranej notacji				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje listę kroków algorytmu iteracyjnego, testując go dla wybranych danych</p>	<p>zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego;</p> <p>podaje przykłady algorytmów iteracyjnych;</p> <p>tworzy algorytm z warunkiem prostym i pętlą;</p> <p>testuje rozwiązanie dla wybranych danych;</p> <p>zna i stosuje instrukcję iteracyjną for w wybranym języku programowania;</p> <p>zapisuje prosty algorytm iteracyjny w postaci programu w wybranym języku programowania</p>	<p>analizuje algorytmy, w których występują powtórzenia (iteracje);</p> <p>ocenia zgodność algorytmu ze specyfikacją;</p> <p>analizuje listę kroków algorytmu z pętlą zagnieżdżoną, testując go dla wybranych danych;</p> <p>zapisuje algorytm iteracyjny w postaci programu w wybranym języku programowania</p>	<p>zapisuje w postaci listy kroków algorytmów z warunkami i iteracyjne;</p> <p>pisze listę kroków algorytmu z pętlą zagnieżdżoną;</p> <p>ocenia zgodność algorytmu ze specyfikacją problemu;</p> <p>wie, kiedy należy zastosować pętlę zagnieżdżoną;</p> <p>zapisuje w postaci programu wybrany algorytm z pętlą zagnieżdżoną;</p> <p>testuje program dla wybranych danych</p>	<p>przestrzega zasad zapisu algorytmów w zadanej postaci (notacji);</p> <p>stosuje listy kroków w opisie zadań (problemów) z innych przedmiotów szkolnych oraz różnych dziedzin życia;</p> <p>wskazuje podobieństwa i różnice dotyczące działania instrukcji warunkowych i iteracyjnej for w różnych językach programowania;</p> <p>zapisuje trudniejsze algorytmy iteracyjne w wybranym języku programowania;</p> <p>zapisuje programy w czytelnej postaci – stosuje wcięcia, komentarze;</p> <p>uczestniczy w konkursach/olimpiadach informatycznych</p>

Funkcje w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym są podprogramy; w wybranym języku programowania definiuje prostą funkcję niezwracającą wartości bez parametrów.</p> <p>wie, jak wywołać funkcję niezwracającą wartości bez parametrów w programie; wywołuje taką funkcję w programie</p>	<p>wie, na czym polega programowanie strukturalne; pisze programy, stosując funkcje; definiuje funkcje niezwracające wartości bez parametrów i z parametrami; stosuje te funkcje w programach; wyjaśnia pojęcia zmienna lokalna i zmienna globalna</p>	<p>wymienia modele programowania; rozumie i stosuje zasady programowania strukturalnego; wyjaśnia, czym są parametry funkcji; wyjaśnia, czym jest funkcja zwracająca wartość, porównuje do funkcji niezwracającej wartości i wskazuje różnice; wie, na czym polega wywołanie funkcji z parametrami i wywołuje taką funkcję w programach; definiuje funkcje zwracające wartości bez parametrów i z parametrami oraz stosuje je w programach; wie, co to jest zasięg zmiennej i jakie ma znaczenie; deklaruje odpowiednio zmienne lokalne i globalne w programach</p>	<p>omawia modele programowania; rozumie zasady postępowania przy rozwiązywaniu problemu metodą zstępującą; wyjaśnia, czym różni się programowanie zstępujące od wstępującego; definiuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami oraz stosuje je w programach; wyjaśnia, na czym polega przesłanianie zmiennych globalnych; wyjaśnia, na czym polega przesłanianie parametrów funkcji; modyfikuje programy, stosuje zmienne globalne i lokalne, objaśnia otrzymane wyniki</p>	<p>porównuje modele programowania, wskazując różnice; sprawnie definiuje i stosuje funkcje niezwracające wartości i zwracające wartość bez parametrów i z parametrami w programach; rozwiązuje przykładowe zadania z matury i olimpiady informatycznej; potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić różnice w stosowaniu zmiennych lokalnych i globalnych, omówić zasięg zmiennych i przesłanianie zmiennych; potrafi, na przykładzie programu utworzonego według własnego pomysłu, wyjaśnić przesłanianie parametrów funkcji</p>

Tablice w wybranym języku programowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje i omawia gotowe przykłady programów, w których są użyte tablice lub listy;</p> <p>potrafi wyjaśnić, dlaczego do rozwiązania niektórych zadań należy użyć tablic (list)</p>	<p>zna pojęcia: <i>tablica</i>, <i>lista</i>, <i>zmienna indeksowana</i>;</p> <p>na bazie przykładów, deklaruje tablicę i/lub listę, wczytuje i wyprowadza elementy tablicy i/lub listy, definiując odpowiednie funkcje w wybranym języku programowania</p>	<p>rozdziela struktury danych proste i złożone oraz podaje ich przykłady;</p> <p>wczytuje i wyprowadza elementy tablicy i/lub listy; definiuje odpowiednie funkcje; wie, jak odwołać się do elementu tablicy i/lub listy; potrafi zastosować tablicę (listę) w zadaniach oraz modyfikować program, znaleźć błędy i je poprawić</p>	<p>rozumie, na czym polega dobór struktur danych do algorytmu i tworzy programy, dobierając odpowiednie struktury danych do programu;</p> <p>deklaruje tablicę dwuwymiarową w języku C++; definiuje odpowiednie funkcje; definiuje różne listy (w tym dwuwymiarowe) w języku Python;</p> <p>tworzy programy, dobierając odpowiednie struktury danych do programu</p>	<p>omawia podobieństwa i różnice w definiowaniu tablic w języku C++ i list w języku Python;</p> <p>stosuje w programach listy jednowymiarowe i dwuwymiarowe, odpowiednio dobierając określoną strukturę danych (tu: rodzaj listy) do algorytmu;</p> <p>rozwiązuje przykładowe zadania z konkursów i olimpiady informatycznej, w których należy zastosować tablice i/lub listy i funkcje; dobiera najlepszy algorytm i odpowiednie struktury danych do rozwiązania postawionego problemu</p>

Stosowanie instrukcji iteracyjnych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli for;</p> <p>analizuje i omawia działanie gotowych programów zapisanych w wybranym języku programowania, zawierających instrukcję pętli while</p>	<p>zna pojęcie iteracji i rozumie pojęcie algorytmu iteracyjnego, podaje ich przykłady;</p> <p>zna postać i działanie instrukcji iteracyjnej while w językach C++ i/lub Python i stosuje ją w tworzonych programach komputerowych</p>	<p>zna sposoby zakończenia iteracji, określa kroki iteracji;</p> <p>stosuje instrukcję while w programach komputerowych;</p> <p>w języku C++ stosuje instrukcję do...while w programach komputerowych;</p> <p>zna różne sposoby określania liczby iteracji w języku Python, np. poprzez powtarzanie poleceń zawartych wewnątrz pętli dla konkretnych wartości lub poprzez podanie liczby powtórzeń z zastosowaniem funkcji range ()</p>	<p>rozumie różnicę pomiędzy instrukcją for a instrukcją while w wybranym języku programowania;</p> <p>modyfikuje programy, zamieniając pętlę for na while i odwrotnie;</p> <p>ocenia program po zmianie (styl, czytelność);</p> <p>pisze programy, stosując poznane instrukcje iteracyjne;</p> <p>dobiera odpowiednią technikę algorytmiczną i odpowiednie struktury danych do rozwiązywanego problemu</p>	<p>potrafi samodzielnie zastosować odpowiedni rodzaj instrukcji pętli w tworzonym programie;</p> <p>potrafi samodzielnie dobrać odpowiednią instrukcję while lub do...while;</p> <p>omawia podobieństwa i różnice w działaniu wszystkich omówionych instrukcji pętli w języku C++;</p> <p>sprawnie pisze trudniejsze programy, stosując poznane instrukcje iteracyjne;</p> <p>wymyśla samodzielnie problem iteracyjny, formułuje zadanie, pisze jego specyfikację i listę kroków oraz zapisuje w wybranym języku programowania</p>

Iteracyjna realizacja wybranych algorytmów, w tym algorytmów wyszukiwania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>potrafi odróżnić algorytm liniowy od algorytmu iteracyjnego;</p> <p>omawia algorytm znajdowania elementu najmniejszego oraz analizuje listę kroków;</p> <p>analizuje i omawia gotową listę kroków algorytmu Euklidesa w jednej z wersji, testuje algorytm na podstawie listy</p>	<p>zna przykładowe algorytmy na liczbach naturalnych;</p> <p>wie, na czym polega metoda wyszukiwania liniowego i przez połowienie;</p> <p>zapisuje algorytm znajdowania najmniejszego (największego) elementu w postaci programu;</p> <p>zna iteracyjną postać algorytmu Euklidesa w wersji z odejmowaniem;</p> <p>potrafi napisać listy kroków algorytmu Euklidesa w wersji z odejmowaniem;</p> <p>zna metodę „dziel i zwyciężaj”;</p> <p>zapisuje algorytm Euklidesa w wersji z odejmowaniem w postaci programu w wybranym języku programowania</p>	<p>potrafi omówić algorytm naiwny i optymalny jednoczesnego znajdowania największego i najmniejszego elementu w zbiorze;</p> <p>zna iteracyjną postać algorytmu Euklidesa z resztą z dzielenia, pisze listę kroków tego algorytmu;</p> <p>zapisuje algorytm Euklidesa w wersji z resztą z dzielenia w postaci programu w wybranym języku programowania;</p> <p>wyjaśnia na przykładzie różnicę między wersją algorytmu Euklidesa z odejmowaniem a wersją z resztą z dzielenia</p>	<p>określa liczbę porównań w algorytmie naiwnym i optymalnym znajdowania największego i najmniejszego elementu w zbiorze, porównuje otrzymane wyniki;</p> <p>pisze listę kroków algorytmu jednoczesnego znajdowania minimum i maksimum z wykorzystaniem metody dziel i zwyciężaj, zapisuje ten algorytm w postaci programu w języku C++ i/lub Python;</p> <p>omawia zastosowanie schematu Hornera do obliczania wartości wielomianu, pisze listę kroków tego algorytmu i pisze program realizujący algorytm obliczania wartości wielomianu według schematu Hornera;</p> <p>programując w/w algorytmy, definiuje odpowiednie funkcje, dobiera struktury danych, dba o stosowanie podstawowych zasad programowania</p>	<p>podaje przykłady problemów, w których można zastosować wyszukiwanie liniowe lub przez połowienie;</p> <p>pisze trudniejsze programy komputerowe, w których wykorzystuje poznane algorytmy;</p> <p>korzystając z dodatkowych źródeł, wyszukuje informacje o zastosowaniu metody „dziel i zwyciężaj” oraz pisze program według własnego pomysłu pokazujący zastosowanie tej metody;</p> <p>samodzielnie zapoznaje się ze schematem Hornera i zapisuje go w postaci programu, dobierając poprawne struktury danych;</p> <p>korzystając z dodatkowych źródeł, omawia przykłady zastosowań algorytmu Euklidesa i poznaje trudniejsze algorytmy, np. trwałego małżeństwa, problem ośmiu hetmanów, algorytm znajdowania liczb bliźniaczych, potrafi zapisać je w języku programowania</p>

Elementy analizy algorytmów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
wymienia własności algorytmów; potrafi przeanalizować przebieg prostego algorytmu zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego poprawność	zna i omawia własności algorytmów; wie, kiedy algorytm jest poprawny; potrafi przeanalizować przebieg algorytmu (np. obliczania silni) zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego poprawność; analizuje program wybranego algorytmu (np. obliczania silni) i ocenia jego poprawność	wie, jak sprawdzić, czy algorytm jest skończony; potrafi ocenić poprawność działania algorytmu i jego zgodność ze specyfikacją; określa liczbę prostych działań zawartych w algorytmie; określa liczbę prostych działań zawartych w algorytmie; potrafi poprawić program, który jest niepoprawny	wie, kiedy algorytm jest skończony; potrafi przeanalizować przebieg algorytmu zapisanego w postaci listy kroków dla przykładowych danych i ocenić w ten sposób jego skończoność; analizuje program realizujący wybrany algorytm i ocenia jego skończoność; modyfikuje program, aby działał poprawnie; sprawdza poznane własności algorytmów, rozwiązując zadania, m.in. uzasadnia skończoność algorytmu znajdowania największego wspólnego dzielnika (NWD) dwóch liczb naturalnych; oblicza liczbę operacji porównania w algorytmie wyboru minimum z tablicy zawierającej n losowo uporządkowanych liczb	potrafi samodzielnie ocenić poprawność i skończoność wybranych algorytmów; potrafi samodzielnie wymyśleć zadanie (problem), napisać specyfikację zadania, listę kroków i program realizujący to zadanie; korzysta samodzielnie z dodatkowej literatury fachowej; poznane dodatkowe możliwości wybranego języka programowania (np. standardowe funkcje) i stosuje w programach; rozwiązuje przykładowe zadania z olimpiady informatycznej

Moduł D. Wokół Internetu i projektów

Praca w chmurze i 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; uczestniczy czynnie w projekcie grupowym, wykonując proste zadania, np. wprowadza dane do bazy i je aktualizuje	planuje temat projektu; bierze aktywny udział w pracy grupowej jako członek zespołu, gromadząc i selekcjonując materiały do projektu; bierze udział w testowaniu projektu; uczestniczy w przygotowaniu dokumentacji projektu, korzystając z edytora tekstu, ewentualnie z wybranego szablonu	korzystając z chmury, potrafi udostępnić pliki, linki do folderu, umożliwić współdzielenie danego folderu; realizuje projekt na zadany (lub samodzielnie wybrany) temat zgodnie z etapami projektowania, przygotowuje dokumentację projektu; współpracuje w grupie, wykonując projekt na temat projektowania zakupu nowego zestawu komputerowego oraz oprogramowania dla ucznia szkoły ponadpodstawowej oraz wpływu trendów w historycznym rozwoju pojęć i metod informatyki oraz technologii na możliwości rozwoju grafiki komputerowej	korzysta z oprogramowania dostępnego w chmurze, m.in. tworzy dokumenty w edytorze tekstu i umieszcza w chmurze, współdzieli je z innymi współużytkownikami; przygotowuje wybrane zadanie szczegółowe zgodnie z etapami przygotowania projektu; gromadzi materiały i inne pomoce, opracowuje dokumentację projektu, wykorzystując m.in. możliwości pracy w chmurze; udostępnia pliki, linki do folderu; wyszukuje informacje na e-platformach do e-nauczania; prezentuje wykonane zadanie projektowe i inicjuje dyskusje	sprawnie posługuje się środowiskiem przeznaczonym do współpracy i realizacji projektów zespołowych, w tym środowiskiem w chmurze; potrafi pełnić funkcję koordynatora grupy; koordynuje wykonywanie zadań szczegółowych na poszczególnych etapach; zarządza folderami (współdzieleniem) i pracą nad dokumentami, w tym dokumentacją projektu; ustala sposób prezentacji projektu i wyznacza osobę (osoby) do prezentacji; w miarę możliwości współtworzy zasoby udostępniane na platformach do e-nauczania; przygotowuje projekt na wybrany przez siebie temat

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne

opracowany na podstawie materiału edukacyjnego:

Grażyna Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres rozszerzony. Materiał edukacyjny. Część 2,*

dostępnego na stronie internetowej ir.migra.pl

Autorzy: Grażyna Koba, Paweł Rogoziński
MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne dla *Materiału edukacyjnego* (ir.migra.pl) do zakresu rozszerzonego, uwzględniające zmiany wynikające z zawężenia podstawy programowej dla szkoły podstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera i sieci komputerowych

Sieci komputerowe – funkcje urządzeń i protokoły				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna terminologię sieciową (<i>sieć komputerowa, zasoby sieciowe, klient, serwer, switch, router, brama, konfiguracja sieciowa, protokoły sieciowe</i>);</p> <p>potrafi zamienić oktety zapisane w systemie dziesiętnym na system dwójkowy</p>	<p>potrafi wyznaczyć adresy hostów i maskę w podsieci;</p> <p>przedstawia strukturę adresu domenowego;</p> <p>zna podstawowe funkcje urządzeń sieciowych;</p> <p>wie do czego służy szafa rackowa</p>	<p>opisuje podstawowe funkcje urządzeń i protokoły stosowane w przepływie informacji, w tym DNS, DHCP;</p> <p>potrafi odnaleźć w systemie Windows plik <i>hosts</i> oraz dokonać jego modyfikacji;</p> <p>przeprowadza konfigurację routera wg instrukcji;</p> <p>potrafi dokonać montażu urządzeń w szafie rackowej</p>	<p>określa ustawienia sieciowe danego komputera i jego lokalizację w sieci (adres sieciowy, podsieci IPv4, protokoły DHCP, DNS i TCP, określa ustawienia sieciowe komputera);</p> <p>zna standardowe numery portów przypisane wybranym usługom;</p> <p>analizuje wydajność sieci komputerowej w tym bezprzewodowej korzystając z odpowiedniej do tego celu aplikacji</p>	<p>przygotowuje prezentację multimedialną z opisem podstawowych protokołów stosowanych w przepływie informacji i w zarządzaniu siecią;</p> <p>samodzielnie konfiguruje urządzenia sieciowe;</p> <p>zabezpiecza urządzenia sieciowe przed nieuprawnionym dostępem;</p> <p>samodzielnie wykonuje zadania dla zainteresowanych</p>

Moduł B. Wokół dokumentów komputerowych

Stosowanie wybranych funkcji w arkuszu kalkulacyjnym				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>analizuje i realizuje algorytm liniowy w arkuszu kalkulacyjnym, stosując odpowiednie funkcje</p> <p>zna zastosowanie funkcji JEŻELI w arkuszu kalkulacyjnym;</p> <p>wymienia grupy funkcji arkusza kalkulacyjnego np. statystyczne, matematyczne, tekstowe;</p> <p>wie, czym są zależności funkcyjne w arkuszu kalkulacyjnym</p>	<p>potrafi podać i omówić przykłady prostych algorytmów liniowych;</p> <p>zapisuje prostą sytuację warunkową w arkuszu kalkulacyjnym;</p> <p>tworzy formułę w arkuszu kalkulacyjnym sprawdzając czy wprowadzona liczba jest liczbą pierwszą;</p> <p>stosuje wybrane funkcje arkusza (matematyczne, logiczne, statystyczne, tekstowe) do realizacji algorytmów;</p> <p>analizuje przykłady zastosowania funkcji SUMA.JEŻELI, SILNIA, PRAWY,FRAGMENT.TEKST U, DŁ i stosuje je w zadaniach</p>	<p>analizuje realizację algorytmu z warunkami w arkuszu kalkulacyjnym;</p> <p>stosuje i analizuje formuły, sprawdzające czy dana liczba jest liczbą pierwszą;</p> <p>wymienia przykłady fraktali jak: śnieżynka Kocha, dywan i trójkąt Sierpińskiego, paprotka Bransleya;</p> <p>umie przeprowadzić kilka kroków „gry w chaos”;</p> <p>stosuje w zadaniach funkcje SUMA.JEŻELI i SILNIA, PRAWY,FRAGMENT.TEKST U, DŁ w arkuszu kalkulacyjnym;</p> <p>tworzy formuły na obliczenie funkcji liniowej, kwadratowej, trygonometrycznej i logarytmicznej i tworzy odpowiednio wykresy tych funkcji</p>	<p>zapisuje warunki zagnieżdżone w arkuszu kalkulacyjnym;</p> <p>dobiera funkcje arkusza kalkulacyjnego w zależności od rozwiązywanego problemu, w tym rodzaju danych;</p> <p>samodzielnie szkicuje trójkąt Sierpińskiego;</p> <p>omawia algorytm „gry w chaos” w arkuszu kalkulacyjnym;</p> <p>samodzielnie realizuje ćwiczenia z wykorzystaniem wybranych wzorów matematycznych, np. silni i symbolu Newtona;</p> <p>wykorzystuje w rozwiązaniach funkcje matematyczne, tekstowe i statystyczne i sprawdza poprawność rozwiązania</p>	<p>stosuje zaawansowane funkcje arkusza do realizacji algorytmów liniowych, z warunkami i iteracyjnymi i rysowania fraktali;</p> <p>samodzielnie analizuje algorytm konstruowania paprotki Barnsleya;</p> <p>samodzielnie tworzy arkusze kalkulacyjne i stosuje poznane funkcje zgodnie z ich przeznaczeniem;</p> <p>wykonuje zadania dla zainteresowanych</p>

Więcej o tworzeniu i edytowaniu grafiki wektorowej				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna cechy grafiki wektorowej; wie, czym różni się grafika wektorowa od grafiki rastrowej;</p> <p>potrafi wymienić nazwy programów do tworzenia grafiki wektorowej;</p> <p>wyjaśnia czym jest ścieżka; tworzy proste kształty; zapisuje plik do podanego formatu</p>	<p>potrafi korzystać z narzędzi <i>Kaligrafia, Narzędzie węzłów</i>; używa podstawowych skrótów klawiszowych;</p> <p>stosuje właściwe formaty do zapisu plików graficznych; wie, czym jest efekt ścieżki;</p> <p>wymienia znane operacje logiczne</p>	<p>korzystając z poznanych narzędzi, tworzy rysunek według podanej specyfikacji; zmienia kształty korzystając z tzw. węzłów edycyjnych; zamienia kształt w ścieżkę; stosuje efekty ścieżki do zmiany wyglądu obiektu;</p> <p>tworzy projekt graficzny według podanych założeń</p>	<p>samodzielnie modyfikuje kształt i zmienia jego właściwości;</p> <p>potrafi łączyć działanie kilku narzędzi w tworzeniu i modyfikacji obiektów; wykorzystuje efekty przekształceń do uzyskania kształtów według swojego pomysłu;</p> <p>wykonuje na obiektach operacje logiczne; samodzielnie tworzy projekt graficzny</p>	<p>tworzy i edytuje obrazy wektorowe z wykorzystaniem zaawansowanych funkcji programu;</p> <p>stosuje wybrane narzędzia edycji obrazu i narzędzia pomocnicze, m.in. wybrane operacje na ścieżkach, klonowanie i inne możliwości zaawansowanego modyfikowania obiektów;</p> <p>wykorzystuje program Inkscape i Tinkercad do drukowania modelu 3D</p>

Moduł C. Wokół algorytmiki i programowania

Wybrane struktury danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym są <i>typ danych</i> i <i>format danych</i>;</p> <p>wymienia podstawowe proste i złożone typy danych;</p> <p>zna sposoby wprowadzania i wyprowadzania danych w języku C++ i/lub Python;</p> <p>potrafi wytłumaczyć, czym są dane losowe;</p> <p>wyjaśnia, czym jest sprawdzanie poprawności danych w programie komputerowym</p>	<p>korzysta z wybranych zbiorów i podzbiorów typów w językach C++ i Python;</p> <p>potrafi określić do jakiego typu należą zadeklarowane zmienne w kodzie źródłowym;</p> <p>zna wybrane elementy formatowania wykorzystywane przez funkcję <code>printf()</code> i <code>print</code>;</p> <p>wymienia funkcje do generowania danych losowych w języku C++ i/lub Python;</p> <p>zna funkcje standardowe, które można użyć do sprawdzenia poprawności danych</p>	<p>stosuje operator <code>sizeof()</code> w języku C++ do określania liczby bajtów pamięci zajmowanych przez zmienne;</p> <p>stosuje funkcje <code>getsizeof()</code> i <code>type()</code> w języku Python;</p> <p>używa wbudowanych funkcji;</p> <p>objaśnia sposób formatowania poszczególnych wyników wykorzystanych przez funkcję <code>printf()</code> i <code>print</code>;</p> <p>wie, do czego służą f-łańcuchy w języku Python;</p> <p>analizuje przykłady stosowania danych losowych;</p> <p>analizuje przykłady sprawdzające poprawność danych liczbowych w języku C++ i/lub Python</p>	<p>oblicza i wyświetla wielkości pamięci zajmowane przez zmienne wybranych typów;</p> <p>oblicza i wypisuje minimalne i maksymalne wartości dające się zapamiętać w zmiennej;</p> <p>samodzielnie tworzy program z wykorzystaniem formatowania wyprowadzanych wyników w języku C++ i/lub Python;</p> <p>korzysta z f-łańcuchów w języku Python;</p> <p>tworzy program symulacji rzutów sześcienną kostką do gry;</p> <p>tworzy programy sprawdzające poprawność wprowadzonych danych liczbowych w języku C++ i/lub Python</p>	<p>wykonuje zadania dla zainteresowanych;</p> <p>potrafi wybrać odpowiednie struktury danych do wykonania zadań;</p> <p>samodzielnie bez pomocy nauczyciela rozwiązuje ćwiczenia i zadania;</p> <p>przygotowuje program według własnego pomysłu wykorzystujący poznane struktury danych</p>

Przetwarzanie plików				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym nazywamy strumień danych;</p> <p>wie, czym jest kod ASCII, zna liczbę pozycji w tabeli;</p> <p>wymienia nazwy innych stron kodowych;</p> <p>omawia przykłady błędów w programach komputerowych</p>	<p>zna cel wykorzystania metod <code>seekp()</code>, <code>seekg()</code> (w języku C++) oraz/lub metody <code>seek()</code> (w języku Python);</p> <p>zna funkcje do obsługi błędów w języku C++;</p> <p>zna konstrukcje do obsługi błędów w języku Python</p>	<p>wykorzystuje klasy i funkcje (zależnie od języka programowania) do odczytu i zapisu danych;</p> <p>analizuje gotowy program odczytujący dane z pliku i wyświetlający je na ekranie;</p> <p>analizuje gotowy program zapisujący dane do pliku;</p> <p>stosuje funkcje do obsługi błędów w języku C++ i/lub stosuje konstrukcje do obsługi błędów w języku Python</p>	<p>używa funkcji obsługi plików w programach komputerowych;</p> <p>analizuje gotowy program z obsługą błędów;</p> <p>samodzielnie modyfikuje programy zgodnie z treścią ćwiczeń</p>	<p>wykonuje zadania dla zainteresowanych;</p> <p>potrafi wybrać odpowiednie struktury danych do wykonania zadań;</p> <p>samodzielnie bez pomocy nauczyciela rozwiązuje ćwiczenia i zadania</p>

Sito Eratostenesa, rozkład liczby na czynniki pierwsze oraz znajdowanie w ciągu podciągów				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna i potrafi wytłumaczyć algorytm badania pierwszościci liczby metodą sita Eratostenesa;</p> <p>omawia rozkład liczby na czynniki pierwsze;</p> <p>zna definicję podciągu, podciągu spójnego i podaje przykłady</p>	<p>wskazuje i nazywa konstrukcje programistyczne wykorzystane w przykładach programów generowania liczb pierwszych metodą sita Eratostenesa;</p> <p>analizuje opis algorytmu rozkładu liczby na czynniki pierwsze;</p> <p>podaje przykłady podciągów spójnych;</p> <p>zna definicję liczby doskonałej</p>	<p>analizuje realizację algorytmu generowania liczb pierwszych metodą sita Eratostenesa w języku C++ i/lub Python;</p> <p>potrafi wytłumaczyć korzystając z listy kroków algorytm wyszukiwania liczby doskonałej;</p> <p>planuje znajdowanie najdłuższego spójnego podciągu niemalejącego;</p> <p>planuje znajdowanie spójnego podciągu o największej sumie</p>	<p>tworzy programy komputerowe w językach C++ i/lub Python wyznaczające liczby pierwsze metodą sita Eratostenesa;</p> <p>tworzy programy komputerowe rozkładu liczby na czynniki pierwsze w języku C++ i/lub Python;</p> <p>samodzielnie sprawdza, czy dana liczba jest liczbą doskonałą;</p> <p>pisze i modyfikuje programy znajdujące najdłuższy spójny podciąg niemalejący oraz o największej sumie</p>	<p>wykonuje zadania dla zainteresowanych, w tym znajdowanie liczb bliźniaczych;</p> <p>potrafi wybrać odpowiednie struktury danych do wykonania zadań;</p> <p>przygotowuje program według własnego pomysłu wykorzystujący poznane struktury danych</p>

Stosowanie metody zachłannej i metody połowienia				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
omawia definicję algorytmu zachłannego; podaje przykłady wykorzystania algorytmu zachłannego	tłumaczy algorytm wydawania reszty metodą zachłanną; wie, na czym polega algorytm połowienia i demonstrowuje ten algorytm na przykładzie	analizuje na konkretnym przykładzie algorytm wydawania reszty metodą zachłanną; analizuje funkcje realizujące algorytm wydawania reszty metodą zachłanną i wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia	analizuje programy realizujące algorytm zachłanny wydawania reszty i wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia w języku C++ i/lub Python; objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych zastosowane w w/w/ programach	pisze program realizujący algorytm wyszukiwania elementu w zbiorze uporządkowanym metodą połowienia w języku C++ i/lub Python; znajduje błędy w wykonanym programie i je poprawia wykonuje zadania dla zainteresowanych; przygotowuje program według własnego pomysłu wykorzystujący poznane techniki programistyczne

Szyfrowanie				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>wie, czym jest szyfrowanie z kluczem;</p> <p>zna różnice między algorytmem szyfrowania symetrycznego a algorytmem szyfrowania asymetrycznego</p>	<p>omawia schemat szyfrowania z kluczem;</p> <p>potrafi scharakteryzować szyfr Vigenère'a;</p> <p>wymienia zastosowanie szyfrowania w życiu codziennym</p>	<p>analizuje przykład stosowania szyfru Vigenère'a;</p> <p>określa wady i zalety szyfrowania z kluczem jawnym</p>	<p>dokonuje szyfrowania szyfrem Vigenere'a;</p> <p>opisuje wykorzystanie algorytmów szyfrowania w podpisie elektronicznym;</p> <p>samodzielnie wykonuje zadania</p>	<p>przygotowuje prezentację multimedialną dotyczącą algorytmu RSA;</p> <p>wykonuje zadania dla zainteresowanych</p>

Programowanie w języku Java				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>potrafi zainstalować środowisko programistyczne języka Java;</p> <p>tworzy prosty program;</p> <p>potrafi wyjaśnić różnice między kompilacją a interpretacją kodu;</p> <p>zna definicję zmiennej w ujęciu informatycznym i matematycznym;</p> <p>potrafi przedstawić konstrukcję instrukcji warunkowej;</p> <p>potrafi wymienić operatory porównania;</p> <p>wymienia nazwy instrukcji iteracyjnych stosowanych w języku Java;</p> <p>wie, do czego służy tablica w języku programowania</p>	<p>opisuje elementy programu w języku Java;</p> <p>wie, czym jest <i>pakiet</i>;</p> <p>zna etapy tworzenia programu;</p> <p>wymienia podstawowe typy zmiennych;</p> <p>porównuje instrukcję warunkową z postacią uproszczoną;</p> <p>potrafi wymienić operatory logiczne;</p> <p>zna strukturę instrukcji iteracyjnych;</p> <p>tworzy i inicjalizuje tablice</p>	<p>analizuje błędy kompilacji i rozwiązuje problemy;</p> <p>deklaruje zmienne różnych typów i inicjalizuje te zmienne;</p> <p>wprowadza dane z klawiatury;</p> <p>wyprowadza komunikaty i wyniki działania programu na ekran monitora;</p> <p>stosuje instrukcję warunkową z warunkiem wykorzystującym operator porównania;</p> <p>tworzy proste programy wykorzystujące instrukcje iteracyjne;</p> <p>używa metod;</p> <p>ustawia dostęp do metod, pól i klas;</p> <p>potrafi uzyskać dostęp do elementów tablicy</p>	<p>używa metod klasy <code>Scanner</code>;</p> <p>stosuje zmienne w programie;</p> <p>samodzielnie modyfikuje programy;</p> <p>zapisuje algorytm z warunkiem logicznym złożonym;</p> <p>tworzy programy wykorzystujące instrukcje iteracyjne;</p> <p>dba o czystość kodu w programach komputerowych;</p> <p>deklaruje metody ze zmienną liczbą parametrów;</p> <p>deklaruje tablice wielowymiarowe</p>	<p>tworzy programy według własnego pomysłu, wykorzystujące poznane techniki programistyczne;</p> <p>samodzielnie poprawia ewentualne błędy w swoich programach;</p> <p>samodzielnie realizuje algorytm z warunkiem logicznym złożonym;</p> <p>potrafi zastosować kilka instrukcji iteracyjnych w jednym programie;</p> <p>potrafi wykorzystać techniki programowania obiektowego w zadaniach;</p> <p>wykonuje zadania dla zainteresowanych</p>

Moduł D. Wokół Internetu i projektów

Zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>potrafi wyjaśnić cel projektu;</p> <p>określa propozycję zadań szczegółowych;</p> <p>wykonuje proste elementy projektu</p>	<p>potrafi wyszukać informacje potrzebne do wykorzystania w projekcie;</p> <p>wykonuje poszczególne zadania szczegółowe;</p> <p>tworzy slajdy w prezentacji i uzupełnia treścią</p>	<p>dba o sprawdzenie części merytorycznej i technicznej projektu;</p> <p>wybiera odpowiednie narzędzia;</p> <p>rozszerza slajdy prezentacji o elementy multimedialne</p>	<p>koordynuje prace zespołu projektowego;</p> <p>zarządza czasem wykonania poszczególnych etapów projektu;</p> <p>łączy zadania szczegółowe w jeden plik;</p> <p>prezentuje fragment projektu</p>	<p>samodzielnie prezentuje cały projekt;</p> <p>odpowiada na pytania;</p> <p>prowadzi dyskusję</p>

Opis założonych osiągnięć ucznia – wymagania na poszczególne oceny szkolne

opracowany na podstawie materiału edukacyjnego:

Grażyna Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres rozszerzony. Materiał edukacyjny. Część 3,*

dostępnego na stronie internetowej ir.migra.pl

Autorzy: Grażyna Koba, Paweł Rogoziński
MIGRA 2024

Przedstawiamy wymagania na poszczególne oceny szkolne dla *Materiału edukacyjnego* (ir.migra.pl) do zakresu rozszerzonego, uwzględniające zmiany wynikające z zawężenia podstawy programowej dla szkoły podstawowej na podstawie rozporządzenia MEN z 2024 roku: *Rozporządzenie Ministra Edukacji zmieniającego rozporządzenie w sprawie podstawy programowej kształcenia ogólnego dla liceum ogólnokształcącego, technikum oraz branżowej szkoły II stopnia.*

Moduł A. Wokół komputera i sieci komputerowych

Reprezentacja danych w komputerze				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcia <i>wartość logiczna</i> i <i>standard kodu</i>;</p> <p>wymienia nazwy kodów (np. ASCII, Windows-1250 i ISO-8859-2);</p> <p>zna podstawowe pozycyjne systemy liczbowe: binarny i dziesiętny;</p> <p>podaje przykład prostej konwersji liczby binarnej na dziesiętną;</p> <p>wie, że komputer wykonuje operacje arytmetyczne na wartościach binarnych;</p> <p>wskazuje w zapisie binarnym najmniej i najbardziej znaczący bit</p>	<p>rozumie reprezentację wartości logicznych 0 i 1 w pamięci komputera;</p> <p>zna liczbę znaków w kodzie ASCII i Unicode;</p> <p>przelicza proste liczby między systemami dziesiętnym i NKB;</p> <p>wie, czym jest reprezentacja stałopozycyjna i zmiennopozycyjna liczb;</p> <p>wykonuje proste działania arytmetyczne na liczbach binarnych;</p> <p>zna oznaczenia operacji logicznych (np. OR, AND, NOT)</p>	<p>definiuje i wyjaśnia pojęcia <i>kodowania znaków</i> i <i>wartości logicznej</i>;</p> <p>porównuje różne systemy kodowania znaków (np. ASCII i Unicode);</p> <p>wyznacza rozwinięcie dwójkowe liczby ujemnej w kodzie U2;</p> <p>zapisuje liczby dziesiętne jako liczby zmiennoprzecinkowej pojedynczej precyzji;</p> <p>wykonuje działania arytmetyczne i logiczne na liczbach binarnych</p>	<p>omawia szczegółowo reprezentację znaków w różnych standardach (ASCII, Unicode);</p> <p>tworzy własne przykłady konwersji znaków;</p> <p>przedstawia praktyczne zastosowanie różnych systemów liczbowych w technice komputerowej;</p> <p>analizuje błędy wynikające z ograniczeń reprezentacji liczb;</p> <p>stosuje arytmetykę komputerową w praktycznych zadaniach;</p> <p>przekształca liczby w różnych formatach i wykonuje na nich działania</p>	<p>planuje i realizuje projekt pokazujący konwersję znaków między standardami;</p> <p>porównuje dokładność różnych metod reprezentacji liczb;</p> <p>przedstawia powiązania między reprezentacją danych a architekturą komputera</p>

Moduł B. Wokół dokumentów komputerowych

Projektowanie relacyjnej bazy danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>podaje obszary zastosowań baz danych – na przykładach z najbliższego otoczenia – szkoły, instytucji naukowych, społecznych i gospodarczych;</p> <p>podaje przykłady programów do tworzenia baz danych;</p> <p>potrafi wykonać podstawowe operacje na bazie danych przygotowanej w jednej tabeli (wprowadzanie, redagowanie, sortowanie, wyszukiwanie, prezentacja);</p> <p>potrafi utworzyć prostą kwerendę</p>	<p>rozumie metody przetwarzania danych na przykładzie gotowej bazy danych;</p> <p>określa podstawowe pojęcia (<i>rekord, pole, typ pola, relacja, klucz podstawowy</i>);</p> <p>tworzy bazę danych składającą się z dwóch tabel, w każdej po kilka pól różnych typów;</p> <p>tworzy proste kwerendy wybierające;</p> <p>potrafi wykonywać operacje przetwarzania danych w bazie składającej się z kilku rekordów;</p> <p>zna zasady przygotowania korespondencji seryjnej</p>	<p>projektuje relacyjną bazę danych (na zadany temat) składającą się z trzech tabel połączonych relacją;</p> <p>omawia typy relacji w bazie danych;</p> <p>zna zasady definiowania kluczy podstawowych;</p> <p>tworzy kwerendy wybierające;</p> <p>importuje dane z tabel arkusza kalkulacyjnego i dokumentu tekstowego do tabel bazy danych;</p> <p>eksportuje dane z tabel bazy danych do tabel arkusza kalkulacyjnego i do dokumentu tekstowego</p>	<p>potrafi wytłumaczyć pojęcie <i>relacji</i>;</p> <p>projektuje relacyjną bazę danych składającą się z trzech lub większej liczby tabel;</p> <p>samodzielnie ustala zawartość bazy (rodzaj informacji);</p> <p>stosuje funkcje standardowe w kwerendach i standardowe operatory w kryteriach wyszukiwania;</p> <p>korzysta z parametrów w kwerendzie</p>	<p>zna dokładnie wybrany program do projektowania baz danych;</p> <p>potrafi samodzielnie zaprojektować bazę danych, korzystając z wybranego narzędzia (programu);</p> <p>projekt bazy opiera na rzeczywistych informacjach, aby można było wykorzystać ją w praktyce, np. w szkole czy w domu;</p> <p>korzysta z dodatkowej, fachowej literatury</p>

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

Trójwymiarowe wizualizacje i animacje				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>korzysta z podstawowych narzędzi programu SketchUp do tworzenia modelu 3D; tworzy prosty model 3D (np. pudełko), korzystając z instrukcji ćwiczenia; zapisuje model w pliku</p>	<p>rozumie zastosowanie narzędzi Follow Me i Pull/Push w prostym modelu; wymienia kroki tworzenia animacji scen w programie SketchUp; wykorzystuje projekt jako punkt wyjścia dla animacji; podaje minimum cztery sceny w animacji modelu 3D zgodnie z ćwiczeniem</p>	<p>definiuje pojęcia bryły obrotowej i toru w kontekście Follow Me; porównuje tworzenie brył zwykłych i obrotowych w SketchUp; stosuje narzędzia do tworzenia brył obrotowych i animacji składającej się z co najmniej czterech scen</p>	<p>omawia funkcję narzędzia Follow Me i jej zastosowanie w zaawansowanych detalach brył; tworzy własny projekt modelu 3D zgodnie z zadaniem, wykazując się indywidualnością; stosuje animację 3D w programie SketchUp, tworząc sekwencję scen z różnymi widokami</p>	<p>planuje i realizuje kompleksowy projekt wizualizacji i animacji 3D; opracowuje dokumentację projektu i prezentuje ją w formie multimedialnej; tworzy model 3D z zaawansowanymi funkcjami SketchUp i eksportuje efekt wideo; analizuje wykonaną wizualizację i animacją 3D oraz przedstawia propozycje usprawnienia projektu</p>

Moduł C. Wokół algorytmiki i programowania

Złożone struktury danych				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcia <i>struktury danych</i> i/lub <i>klasy danych</i>;</p> <p>wymienia przykłady typów strukturalnych w C++ i/lub klas danych w Pythonie;</p> <p>podaje przykład prostego typu strukturalnego;</p> <p>korzysta z gotowego kodu, aby utworzyć zmienną typu struktura lub klasy danych i przypisać do niej wartości</p>	<p>rozumie, jak typ strukturalny w C++ i/lub klasa danych w Pythonie grupuje dane różnych typów w jednym obiekcie;</p> <p>wymienia zastosowanie typu słownikowego i go stosuje w Pythonie;</p> <p>wykorzystuje typ strukturalny i/lub klasę danych w prostym programie, wprowadzając dane i je wyświetlając</p>	<p>definiuje tablicę struktur w C++ i/lub listę obiektów klasy w Pythonie jako odpowiednik tabeli w bazie danych;</p> <p>porównuje implementację elementów bazy danych (<i>rekord, tabela, indeks</i>) ze strukturami w C++ i/lub Pythonie;</p> <p>stosuje tablicę struktur i/lub listę obiektów do przechowania wielu rekordów i wyszukiwania według kryterium;</p> <p>przedstawia ograniczenia prostych struktur danych</p>	<p>omawia zaawansowane aspekty klas danych w Pythonie i/lub struktur zagnieżdżonych w C++;</p> <p>tworzy własną strukturę lub klasę opisującą konkretny obiekt i implementuje listę takich obiektów lub tablicę struktur;</p> <p>stosuje typ słownikowy w Pythonie i/lub strukturę indeksującą w C++ do wyszukiwania danych według klucza;</p> <p>analizuje wpływ struktur danych na efektywność rozwiązania problemu</p>	<p>planuje i realizuje projekt programu, dobierając odpowiednie struktury danych do złożonego problemu;</p> <p>opracowuje dokumentację programu (opis typów danych, struktur, ich relacji i uzasadnienie wyboru);</p> <p>tworzy program wykorzystujący strukturę danych z wieloma rekordami oraz indeksem lub słownikiem</p>

Arytmetyka komputerowa i systemy liczbowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcie <i>systemu liczbowego</i> innego niż dziesiętny (np. binarnego, szesnastkowego);</p> <p>wymienia podstawowe działania arytmetyczne w systemie binarnym;</p> <p>podaje przykład liczby zapisanej w systemie binarnym i przelicza ją na system dziesiętny;</p> <p>zna pojęcia nadmiar i niedomiar w kontekście arytmetyki komputerowej;</p> <p>wymienia przykład sytuacji, gdy dodanie dwóch liczb kończy się błędem z powodu skończonej liczby bitów;</p> <p>podaje, że typy zmiennoprzecinkowe mają ograniczoną liczbę bitów, co wpływa na dokładność obliczeń</p>	<p>rozumie algorytm dodawania dwóch liczb w systemie binarnym;</p> <p>stosuje funkcję konwersji liczby z systemu binarnego na dziesiętny lub odwrotnie;</p> <p>wykorzystuje przykładowy kod (np. w języku C++ lub Python) do wykonania działania dodawania lub odejmowania w systemie innym niż dziesiętny;</p> <p>rozumie, w jaki sposób skończona liczba bitów może powodować błędy zapisu liczb i operacji arytmetycznych;</p> <p>stosuje prosty program lub analizuje przykład, gdzie występuje nadmiar lub niedomiar;</p> <p>wykorzystuje pojęcie błędu przybliżenia lub zaokrąglenia w obliczeniach zmiennoprzecinkowych i podaje jego przyczynę</p>	<p>definiuje funkcje pomocnicze dla algorytmu: np. zamianę cyfry na wartość i odwrotnie;</p> <p>porównuje procedury dodawania i odejmowania w systemie binarnym z odpowiednikami w systemie dziesiętnym;</p> <p>stosuje algorytm dodawania lub mnożenia w systemie binarnym w programie, uwzględniając przeniesienie lub przesunięcie;</p> <p>definiuje pojęcia <i>błąd zaokrąglenia</i> i <i>błąd przybliżenia</i> oraz wyjaśnia, jak one powstają w reprezentacji zmiennoprzecinkowej;</p> <p>porównuje sytuację nadmiaru i niedomiaru w typach całkowitych i rzeczywistych na przykładzie ich zakresów bitowych</p>	<p>omawia programowanie algorytmów mnożenia i dzielenia liczb w systemach innych niż dziesiętny (na przykładzie systemu binarnego);</p> <p>tworzy własny program w języku C++ i/lub Python realizujący działanie arytmetyczne w systemie binarnym;</p> <p>analizuje efektywność użytego algorytmu i przedstawia możliwości jego rozbudowy lub optymalizacji;</p> <p>omawia wpływ właściwości arytmetyki komputerowej na stabilność algorytmów numerycznych;</p> <p>tworzy program, który uwzględni zabezpieczenie przed nadmiarem, niedomiarem lub błędami zmiennoprzecinkowymi</p>	<p>planuje i realizuje projekt, w którym program wykonuje różne działania arytmetyczne w wybranym systemie liczbowym innym niż dziesiętny;</p> <p>opracowuje dokumentację techniczną programu – opis algorytmów, użytych struktur danych i uzasadnienie wyborów;</p> <p>porównuje działanie algorytmów dla różnych podstaw systemów liczbowych;</p> <p>planuje i realizuje projekt lub zadanie programistyczne, w którym dobiera i uzasadnia dobór typów danych oraz algorytmów pod kątem ograniczeń arytmetyki komputerowej.</p> <p>porównuje efektywność i dokładność różnych implementacji algorytmów numerycznych pod kątem wpływu nadmiaru, niedomiaru, zaokrągleń</p>

Algorytmy numeryczne				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcie <i>algorytmu numerycznego</i> i jego zastosowania w programowaniu;</p> <p>wymienia metodę znajdowania miejsca zerowego funkcji;</p> <p>podaje, że algorytmy numeryczne operują na przybliżeniach i wymagają określenia dokładności;</p> <p>korzysta z gotowego kodu lub instrukcji, by uruchomić program realizujący prostą metodę numeryczną</p>	<p>rozumie algorytm metody połowienia przedziału, opisuje podstawowe kroki i warunki zastosowania;</p> <p>stosuje program do realizacji metody połowienia;</p> <p>podaje przykład funkcji, dla której można zastosować metodę połowienia i uzasadnia wybór przedziału</p>	<p>definiuje pojęcie zbieżności algorytmu numerycznego i wyjaśnia, jakie warunki wpływają na jego działanie;</p> <p>porównuje różne kryteria zakończenia algorytmu i wskazuje wady oraz zalety każdego z nich;</p> <p>stosuje w programie numerycznym metodę bisekcji i analizuje otrzymany wynik pod kątem dokładności i liczby iteracji</p>	<p>omawia bardziej zaawansowaną metodę numeryczną (np. metodę Newtona-Raphsona) jako alternatywę dla metody połowienia;</p> <p>tworzy program realizujący metodę Newtona-Raphsona dla wybranej funkcji i analizuje wynik oraz zachowanie algorytmu;</p> <p>stosuje w programie pomiar liczby iteracji oraz czasu działania algorytmu dla różnych wartości startowych i interpretuje różnice;</p> <p>analizuje wpływ doboru parametru dokładności oraz warunków początkowych na efektywność i poprawność działania algorytmu numerycznego</p>	<p>planuje i realizuje projekt, w którym porównuje co najmniej dwie metody numeryczne dla tej samej funkcji i dokumentuje wyniki;</p> <p>opracowuje dokumentację techniczną projektu, opis algorytmów, struktury danych oraz dobór metody i algorytmu</p> <p>porównuje efektywność różnych metod numerycznych dla wielu funkcji – ocenia ich zalety i wady, wskazuje przypadki, w których dana metoda może zawieść lub być mniej efektywna</p>

Zastosowanie rekurencji w programowaniu, w tym do tworzenia fraktali				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcie <i>rekurencji</i> i potrafi wyjaśnić, że funkcja wywołuje samą siebie;</p> <p>wymienia przykłady ciągów liczbowych generowanych rekurencyjnie;</p> <p>zna pojęcie algorytmu Algorytmu Euklidesa i potrafi opisać jego cel;</p> <p>omawia, co oznacza <i>szybkie potęgowanie</i>;</p> <p>zna pojęcie <i>fraktal</i> oraz podaje przykład fraktala;</p> <p>rozumie, że fraktale można generować poprzez rekurencję</p>	<p>rozumie algorytm rekurencyjnego generowania ciągu liczbowego i potrafi opisać jego działanie w uproszczeniu;</p> <p>stosuje prostą funkcję rekurencyjną do wygenerowania n-tego wyrazu ciągu;</p> <p>rozumie rekurencyjną implementację algorytmu Euklidesa i potrafi zapisać jego pseudokod;</p> <p>stosuje w programie funkcję rekurencyjną realizującą szybkie potęgowanie i sprawdza wynik dla prostych danych;</p> <p>omawia zasadę generowania fraktala rekurencyjnie;</p> <p>stosuje podstawowy kod funkcji rekurencyjnej rysującej prostą wersję fraktala</p>	<p>definiuje warunek zakończenia wywołań rekurencyjnych oraz wyjaśnia, dlaczego jest konieczny;</p> <p>porównuje wydajność rekurencyjnego rozwiązania generowania ciągu z jego iteracyjną odpowiednikiem, wskazując główne różnice;</p> <p>definiuje warunek zakończenia w obu algorytmach (Euklidesa i potęgowania) i wyjaśnia jego konieczność w rekurencji;</p> <p>definiuje warunek zakończenia rekurencji w algorytmie tworzącym fraktal i wyjaśnia, dlaczego jest niezbędny;</p> <p>porównuje wizualny efekt dwóch różnych poziomów iteracji fraktala</p>	<p>tworzy własną funkcję rekurencyjną dla wybranego ciągu liczbowego;</p> <p>omawia zalety rekurencyjnej wersji algorytmu Euklidesa oraz potencjalne ograniczenia w zależności od rozmiaru danych;</p> <p>tworzy własny program implementujący rekurencyjnie wyżej wymienione algorytmy;</p> <p>tworzy własną funkcję rekurencyjną w języku programowania generującą fraktal z góry określoną liczbą iteracji lub poziomem zagłębienia;</p> <p>analizuje wpływ liczby iteracji na złożoność obrazu</p>	<p>planuje i realizuje projekt programu, w którego ramach generuje różne typy ciągów liczbowych rekurencyjnie i porównuje je ze sobą;</p> <p>opracowuje dokumentację techniczną, opisuje algorytmy, ich parametry, porównuje ich efektywność i proponuje optymalizacje;</p> <p>planuje i wykonuje projekt graficzny/fraktalny przy użyciu rekurencji – wybiera fraktal, modyfikuje regułę generowania, eksperymentuje z parametrami i dokumentuje zmiany wizualne</p>

Algorytmy sortowania				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcie <i>algorytmu sortowania</i>;</p> <p>wymienia co najmniej dwa algorytmy sortowania (np. przez wstawianie, przez scalanie);</p> <p>omawia na przykładzie działania algorytmu sortowania (np. scalanie dwóch uporządkowanych ciągów)</p>	<p>rozumie zasadę działania <i>algorytmu dziel i zwyciężaj</i> w kontekście sortowania przez scalanie;</p> <p>stosuje w programie algorytm sortowania (np. przez wstawianie) dla niewielkiego zbioru danych;</p> <p>opisuje kroki algorytmu scalania dla dwóch uporządkowanych podciągów w konkretnym przykładzie</p>	<p>definiuje rekurencyjną wersję algorytmu sortowania przez scalanie;</p> <p>porównuje algorytm sortowania przez wstawianie i algorytm scalania pod względem sposobu działania i zakresu zastosowania;</p> <p>stosuje w programie algorytm sortowania przez scalanie, testując go dla nieuporządkowanego ciągu liczb i interpretuje otrzymany wynik</p>	<p>omawia <i>algorytm szybkiego sortowania</i>;</p> <p>tworzy własny program implementujący algorytm szybkiego sortowania i testuje działanie dla różnych danych;</p> <p>analizuje, w jaki sposób wybór <i>pivota</i> i podział danych wpływa na efektywność działania algorytmu sortowania</p>	<p>planuje i realizuje projekt, w którym porównuje co najmniej dwa algorytmy sortowania (np. scalanie i szybkie sortowanie) na tym samym zestawie danych i dokumentuje wyniki;</p> <p>tworzy program, który umożliwi analizę porównań i przesunięć wykonanych przez algorytmy sortowania oraz wizualizuje przebieg działania jednego z algorytmów</p>

Dynamiczne struktury danych i grafy				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna pojęcia <i>tablica dynamiczna</i>, <i>lista jednokierunkowa</i> oraz <i>stos</i> jako przykłady dynamicznych struktur danych;</p> <p>wymienia różnice między statyczną tablicą a tablicą dynamiczną;</p> <p>zna pojęcie <i>programowania dynamicznego</i> i potrafi wymienić przykład jego zastosowania;</p> <p>wymienia pojęcie <i>grafu</i> jako modelu abstrakcyjnego problemu i podaje prosty przykład grafu</p>	<p>rozumie, w jaki sposób tworzy się i zwalnia pamięć dla zmiennej dynamicznej;</p> <p>stosuje prostą implementację listy jednokierunkowej lub stosu;</p> <p>rozumie metodę programowania dynamicznego – dzielenie problemu na podproblemy i wykorzystanie tablicy pomocniczej;</p> <p>stosuje graf do przedstawienia problemu algorytmicznego oraz identyfikuje typ grafu</p>	<p>definiuje operacje w dynamicznych strukturach danych: dodawanie, usuwanie, odczyt elementu – na przykładzie stosu lub kolejki;</p> <p>definiuje zależność między programowaniem dynamicznym i grafami – potrafi wskazać, kiedy obydwa podejścia mogą być stosowane w jednym zadaniu;</p> <p>porównuje zastosowanie programowania dynamicznego i struktur grafowych na przykładzie konkretnego problemu (np. wyszukiwanie ścieżki lub najdłuższego wspólnego podciągu)</p>	<p>tworzy własną implementację dynamicznej struktury danych i dokumentuje kod;</p> <p>analizuje przypadki użycia dynamicznych struktur danych – wskazuje, kiedy statyczna struktura jest niewystarczająca i dlaczego trzeba użyć dynamicznej;</p> <p>tworzy program, w którym wykorzystuje programowanie dynamiczne lub grafy do rozwiązania zadania (np. znajdowanie najdłuższego wspólnego podciągu);</p> <p>analizuje i opisuje, jakie struktury danych są użyte w rozwiązaniu i jakie są ich zalety</p>	<p>realizuje program, który obejmuje co najmniej dwie różne dynamiczne struktury danych;</p> <p>planuje i realizuje projekt, w którym problem modelowany jest zarówno jako graf, jak i poprzez programowanie dynamiczne – wybiera metodę, implementuje ją i dokumentuje</p>

Moduł D. Wokół Internetu i projektów

Zadania projektowe				
2	3	4	5	6
Uczeń:	Uczeń:	Uczeń:	Uczeń:	Uczeń:
<p>zna etapy projektu: określenie zadania, analiza problemu i ustalenie zadań szczegółowych;</p> <p>wymienia formy projektu (np. prezentacja multimedialna) i narzędzia wybrane do realizacji;</p> <p>podaje, że projekt może być wykonywany w grupie i wie, że musi zostać przedstawiona prezentacja końcowa;</p> <p>wykonuje wskazane przez koordynatora zadania szczegółowe całego projektu</p>	<p>rozumie, w jaki sposób wybrać narzędzia do projektu oraz ustalić założenia przechowywania dokumentów (np. w chmurze);</p> <p>stosuje przydział zadań w zespole i wykonuje wybrane zadanie szczegółowe w projekcie;</p> <p>przedstawia prostą formę dokumentu projektu (np. prezentację lub folder z plikami) zgodnie z założeniami projektu</p>	<p>definiuje i opisuje formę końcowego dokumentu projektu (np. prezentacja, raport) oraz normy nazw plików i folderów;</p> <p>porównuje różne narzędzia do współpracy w zespole (np. przechowywanie w chmurze i lokalne foldery) i wskazuje zalety jednego z nich;</p> <p>stosuje w projekcie format dokumentu zgodny z założeniami, łączy materiały i sprawdza, czy wszystkie zadania zostały wykonane według wcześniejszych ustaleń</p>	<p>omawia zalety pracy zespołowej nad projektem – wskazuje korzyści wyboru zadań i współpracy w chmurze;</p> <p>tworzy kompleksowy dokument projektu zawierający wszystkie zadania szczegółowe, spójne nazewnictwo plików i przechowywanie w jednym miejscu;</p> <p>analizuje proces wykonania projektu – sprawdza, czy wyniki odpowiadają analizie i założeniom wstępnym oraz przedstawia wnioski</p>	<p>planuje i realizuje projekt zespołowy lub indywidualny w pełnym zakresie: wybór zadania, narzędzi, wykonanie, współpraca, dokumentacja, prezentacja;</p> <p>opracowuje dokumentację techniczną oraz prezentację końcową – zawierającą analizę problemu, zadania szczegółowe, użyte narzędzia, sposób przechowywania i wnioski</p>