**Część 1,2, 3 – Informatyka rozszerzona**

**Egzamin teoretyczny i praktyczny: zadania do wykonania z zakresu: analiza algorytmów, tworzenie algorytmów, test z ogólnej wiedzy informatycznej, symulacja, przetwarzanie i tworzenie informacji, bazy danych.**

**Link do ćwiczeń: https://cke.gov.pl/egzamin-maturalny/egzamin-maturalny-w-formule-2015/materialy-dodatkowe/materialy-dla-uczniow-i-nauczycieli/zbiory-zadan/**

**Zadania do wykonania:**

* + - 1. Przygotuj fotomontaż na dowolny temat w programie Gimp. Pamiętaj o tworzeniu warstw. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
			2. Przygotuj obrazek na dowolny temat z wykorzystaniem grafiki wektorowej w programie Inkscape. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
			3. Przygotuj stronę internetową w języku HTML na dowolny temat. Ma ona zawierać: formatowanie tekstu, kolory, tło strony, odnośniki do innej własnej podstrony oraz innej strony w Internecie, tabelkę, obrazki, listę wypunktowaną lub numerowaną, kaskadowe arkusze stylów CSS. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
			4. Przygotuj program w języku C++ wyszukujący największą liczbę w ciągu liczb. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.
			5. Przygotuj program w języku C++ z zastosowaniem algorytmu konwersji między systemami liczbowymi binarnym i dziesiętnym. Im więcej wykorzystasz funkcji i możliwości tym wyższa punktacja.

Programy nie mogą być identyczne jak w podręczniku!

Do programowania w języku C++ pobierz program Dev C++ Lub Code Blocks.

**Prace należy przesłać co najmniej tydzień przed egzaminem z pośrednictwem platformy Teams. Zostanie tam utworzone specjalne zadanie**

**Plan wynikowy (propozycja) cz.1**

|  |  |  |
| --- | --- | --- |
| Lp. | Temat | Osiągnięcia uczniów |
| **Wymagania podstawowe. Uczeń:** | **Wymagania** **ponadpodstawowe. Uczeń:** |
| 1 | Systemy operacyjne w środowisku sieciowym | * wymienia systemy operacyjne oraz ich zadania
* rozumie kwestie związane z bezpieczeństwem w przestrzeni cyfrowej
* zna zasady tworzenia mocnych haseł
* rozumie potrzebę stosowania kont użytkownika w systemie operacyjnym
* stosuje zasady bezpieczeństwa obowiązujące w pracowni komputerowej
* instaluje i aktualizuje oprogramowanie
* zakłada i usuwa konto w środowisku aplikacji Google
* pracuje w środowisku sieciowym
 | * wyjaśnia, w jakim trybie (jądra czy użytkownika) powinien pracować program sterownika urządzenia w większości systemów operacyjnych
* zna procedurę wykonania kopii zapasowej dla systemu operacyjnego i wszystkich danych użytkownika komputera
* tworzy nośnik awaryjny uruchamiający komputer, gdy zainstalowany na nim system operacyjny nie działa prawidłowo
* wie, czym są fragmentacja i defragmentacja dysku
* sprawdza poziom fragmentacji dysku komputera i ocenia, czy wymagana jest jego defragmentacja
* wie, jaka jest rola systemu plików jako części systemu operacyjnego
* sprawdza, jaki system plików został przypisany do danego dysku
* wie, w jaki sposób uruchomić tryb awaryjny w systemie Windows (od wersji Windows 7), zna poszczególne opcje dostępne dla trybu awaryjnego i wie, do czego służą
* zna polecenia w trybie tekstowym Windows i posługuje się nimi
 |
| 2 | Nowe technologie i oprogramo-wanie | * rozumie pojęcia takie jak: sztuczna inteligencja, chmura obliczeniowa i posługuje się nimi
* wymienia zastosowania automatyki i robotyki w życiu codziennym
* wskazuje zalety i sposoby wykorzystania druku 3D
 | * proponuje własne, dotąd nieznane, sposoby na wykorzystanie nowych technologii
* wyjaśnia zastosowanie nowych rozwiązań technologicznych w różnych dziedzinach życia
* posługując się darmowymi aplikacjami do tworzenia
* rozszerzonej rzeczywistości, tworzy filmy, artykuły i infografiki
 |
| 3 | Projektowane zestawu komputerowego | * projektuje rozbudowę nowego zestawu komputerowego oraz oprogramowania
 | * projektując zestaw komputerowy, kieruje się podanymi kryteriami (cena, przeznaczenie) i uzasadnia swój wybór
 |
| 4 | Sieci komputerowe – budowa i usługi | * rozumie pojęcia: sieć, protokół sieciowy, topologia sieci
* rozróżnia i poprawnie nazywa sieci komputerowe ze względu na ich zasięg i topologię
* opisuje budowę sieci lokalnej i sieci Internet
* rozumie pojęcia takie jak adres IP, host, router, maska podsieci, brama, DNS oraz omawia zasadę adresowania urządzeń w sieci Internet
* wymienia różne usługi internetowe
* potrafi opisać warstwowy model działania Internetu oraz wymienić zadania poszczególnych warstw
 | testuje prędkość połączenia z siecią Internet na wybranym urządzeniu i interpretuje otrzymany wynik zna polecenia tekstowe służące do diagnostyki sieci i korzysta z nichoblicza liczbę możliwych do zaadresowania hostów na podstawie adresów IP i masek podsiecirozumie, czym jest model warstwowy TCP/IPwyjaśnia sposoby działania usługi NAT |
| 5 | Model warstwowy sieci komputerowych | * opisuje warstwowy model sieci komputerowej oraz model sieci Internet
* opisuje podstawowe funkcje urządzeń i protokoły stosowane w przepływie informacji i w zarządzaniu siecią
 |  |
| 6 | Protokoły IPv4 i IPv6 | * konfiguruje przykładową lokalną sieć komputerową oraz bezprzewodowy dostęp do sieci Internet
 |  |
| 7 | Konfiguracja sieci komputerowej | * wyjaśnia, od czego zależą sprawne funkcjonowanie sieci komputerowej oraz szybki dostęp do jej usług i zasobów
* określa parametry wybranego osprzętu sieciowego, szerokość pasma oraz możliwości serwera
 |  |
| 8 | E-usługi | * poprawnie definiuje pojęcie e-usługi
* wymienia różne zastosowania usług elektronicznych
* charakteryzuje problemy oraz wymienia zalety związane z wykorzystaniem e-usług
 | * opisuje zabezpieczenia wybranych e-usług (w tym systemu ePUAP)
* określa możliwości rozwoju dla wybranych e-usług, z których korzysta
* wymienia narzędzia dostępne w sieci, które umożliwiają utworzenie wybranych e-usług
 |
| 9 | Korzystanie z e-zasobów i współpraca zdalna | * rozumie pojęcie informacji
* korzysta z zasobów internetowych, wyszukując potrzebne informacje
* wymienia etapy rozwoju technologii komputerowych
* korzysta z różnych wyszukiwarek internetowych
* wykorzystuje zasoby sieciowe do poszerzania własnej wiedzy (e-learning)
* zna podstawy prawa autorskiego
* stosuje zasady netykiety i korzysta z niej w komunikacji zdalnej
 | * wie, czym jest pozycjonowanie serwisów internetowych
* wyjaśnia sposób tworzenia wybranych e-zasobów oraz wskazuje zalety i wady poszczególnych rozwiązań
* zna i stosuje zapisy ustawy o prawie autorskim i prawach pokrewnych
 |
| 10 | Rozbudowane dokumenty tekstowe | * korzysta z programu Microsoft Word
* stosuje style nagłówkowe (korzysta z gotowych, tworzy własne i modyfikuje je)
* stosuje numeracje i wypunktowania, dostosowując ich styl
* formatuje elementy dokumentu odpowiedzialne za automatyczne spisy (treści, tabel, ilustracji)
* wstawia w dokumencie spisy treści, tabel, ilustracji
* poprawnie operuje nagłówkiem i stopką dokumentu
* tworzy strony tytułowe
* współpracuje przy edycji dokumentu z innymi użytkownikami, korzystając z opcji recenzji dokumentu
 | * tworzy styl według wzoru
* pracuje nad dokumentem wspólnie z innymi osobami w trybie śledzenia zmian
 |
| 11 | Sztuka prezentacji | * korzysta z programu Microsoft PowerPoint
* zna zasady zachowania się podczas wystąpień publicznych
* opracowuje plan prezentacji
* zna narzędzia i pomoce wizualne wykorzystywane podczas prelekcji
* prezentuje poprawnie sformatowaną treść slajdów
* stosuje efekty i multimedia w prezentacji
 | * dodaje do slajdów swój komentarz głosowy i zapisuje prezentację jako film
 |
| P1 | Nowoczesne technologie w służbie człowiekowi – projekt zespołowy | * aktywnie uczestniczy w realizacji projektów informatycznych
* przyjmuje różne role w zespole realizującym projekt
* prezentuje efekty wspólnej pracy
* uzupełnia swoją wiedzę, korzystając z zasobów udostępnionych na platformie do e-nauczania
 | * przyjmuje rolę lidera odpowiedzialnego za zespół i projekt
 |
| 12 | Moja cyfrowa tożsamość | * definiuje pojęcie cyfrowej tożsamości
* zna problemy zarządzania zasobami cyfrowymi
* bezpiecznie kreuje swój wizerunek w przestrzeni medialnej
* rozumie pojęcie wirtualnej komunikacji i komunikuje się z innymi w środowisku wirtualnym
* dostrzega zalety i wady komunikacji wirtualnej oraz posługiwania się cyfrową tożsamością
* rozumie pojęcie hejtu i dostrzega jego destrukcyjny wpływ
* rozumie zagrożenia wynikające z upraszczania komunikacji za pośrednictwem sieci
* zna narzędzia wirtualnej komunikacji
 | * wie, czym jest zautomatyzowane profilowanie i przetwarzanie danych
* zna prawa przysługujące osobom, których dane są wykorzystywane
 |
| 13 | Przemiany społeczne a technologie | * rozumie i wymienia czynniki przemian społecznych
* dostrzega możliwości wynikające z przemian gospodarczych
* określa obszary w społeczeństwie, na które wpływa rozwój technologii
* rozumie potrzebę stosowania regulacji prawnych i norm etycznych
* wskazuje pozytywne i negatywne skutki rozwoju technologii informacyjnej
* zna wyzwania, przed którymi stoi edukacja
* operuje pojęciami: e-zasoby, e-usługi, e-learning
* rozumie pojęcie mediów i przestrzeni medialnej w kontekście IT
* wskazuje możliwości zapobiegania negatywnym skutkom rozwoju technologii
 | * wymyśla rozwiązania technologiczne, których nie ma jeszcze na rynku
 |
| 14 | Cyber-bezpieczeń-stwo | * rozpoznaje zagrożenia związane z oprogramowaniem komputerowym
* dba o przestrzeganie podstawowych zasad bezpieczeństwa, korzystając z urządzeń mobilnych czy komputera
* bezpiecznie korzysta z bankowości elektronicznej
* umiejętnie i w bezpieczny sposób weryfikuje własną tożsamość, korzystając z e-usług
* rozumie związek ochrony danych osobowych z cyberbezpieczeństwem
* właściwie zachowuje się w sytuacji cyberprzemocy
* stosuje pojęcia związane z bezpieczeństwem w internecie
 | * wymienia symptomy wskazujące na zainfekowanie komputera złośliwym oprogramowaniem
* wie, czym jest infrastruktura krytyczna i jak się ją chroni
 |
| 15 | Podstawy kryptografii | * objaśnia rolę technik uwierzytelniania, kryptografii i podpisu elektronicznego
 | * wyjaśnia, czym są: steganografia, kryptologia, kryptografia, kryptogram, kryptoanaliza
* zna zasadę Kerckhoffsa
* wyjaśnia, jak kolejne certyfikaty protokołu SSL (DV, OV i EV) zwiększają poziom bezpieczeństwa usług sieciowych
* wyjaśnia znaczenie problemu bezkolizyjności funkcji ha-szujących i ich znaczenie dla kryptoanalizy
* prezentuje przykład zaawansowanej kryptoanalizy
 |
| 16 | Tworzenie stron internetowych | * korzysta z różnych przeglądarek internetowych
* zna strukturę strony WWW
* definiuje podstawowe znaczniki HTML
* korzysta z atrybutów znaczników
* zna reguły stosowania arkuszy stylów w połączeniu z kodem HTML
* stosuje narzędzia wspierające pisanie kodu źródłowego
* wyszukuje informacje w sieci i korzysta z zasobów witryn internetowych na temat tworzenia stron WWW
 | * tworzy rozbudowaną stronę WWW z podstronami, tabelą, elementami graficznymi, formatując jej wygląd za pomocą stylów CSS
 |
| 17 | Grafika 2D i 3D | * rozróżnia pojęcia grafiki rastrowej i wektorowej
* stosuje właściwe narzędzia do edycji zdjęć w wybranym programie graficznym
* wykonuje różne operacje na obrazie w grafice rastrowej
* zna różne formaty graficzne dla plików i korzysta z nich
* modeluje proste obiekty w grafice 3D za pomocą wybranego oprogramowania
* rysuje za pomocą narzędzi grafiki wektorowej
 | * tworzy bryły obrotowe i kompozycje obiektów na scenie
* tworzy wektorowe modele sfotografowanego przez siebie wybranego obiektu
* projektuje bardziej skomplikowane trójwymiarowe modele
* tworzy trójwymiarowe modele dowolnego budynku
 |
| 18 | JavaScript na stronach WWW | * projektuje modele dwuwymiarowe i trójwymiarowe
* wykorzystuje różne formaty obrazów
* przekształca pliki graficzne, uwzględniając wielkość i jakość obrazów
* tworzy stronę internetową zgodnie ze standardami, wzbogaconą listami i elementami dynamicznymi
* korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron
* publikuje własną stronę w internecie
* programuje elementy strony internetowej współpracujące z sieciową bazą danych
 | * wie, czym jest hosting
* posługuje się systemem CMS
* sprawdza dostępność nazw dla domeny
* wie, czym są WordPress, Joomla!, Drupal
 |
| 19 | Animacja komputerowa | * tworzy i edytuje dwuwymiarowe oraz trójwymiarowe wizualizacje i animacje
* stosuje właściwe formaty plików graficznych
 |  |
| P2 | Responsywna strona WWW w systemie CMS – projekt zespołowy | * aktywnie uczestniczy w realizacji projektów informatycznych
* przyjmuje różne role w zespole realizującym projekt
* prezentuje efekty wspólnej pracy
* publikuje własną stronę w internecie
 | * przyjmuje rolę lidera odpowiedzialnego za zespół i projekt
 |
| 20 | Jak pobierać dane do arkusza kalkulacyjnego | * gromadzi w tabeli arkusza kalkulacyjnego dane pochodzące z różnych źródeł
* korzysta z różnorodnych funkcji arkusza w zależności od rodzaju danych
* filtruje dane według kilku kryteriów
 |  |
| 21 | Wyciągamy wiedzę z danych | * dobiera wykresy do rodzaju danych
* analizuje dane, korzystając z dodatkowych narzędzi, w tym z tabel i wykresów przestawnych
 |  |
| 22 | Zaawansowane formuły | * stosuje zaawansowane funkcje arkusza kalkulacyjnego w zależności od rodzaju danych
 |  |
| 23 | Konsekwencje zaokrąglania liczb | * wyjaśnia, jakie może być źródło błędów pojawiających się w obliczeniach komputerowych: błąd zaokrąglenia, błąd przybliżenia
 | * wykonuje zadania w arkuszu kalkulacyjnym z wykorzystaniem funkcji służących do zaokrąglania liczb
* zna właściwości formatu walutowego
* projektuje obliczenia walutowe
 |
| 24 | Korespondencja seryjna | * do rozwiązania problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki
* definiuje korespondencję seryjną
* wyszukuje informacje, korzystając z bazy danych opartej na co najmniej dwóch tabelach
* definiuje relacje
* stosuje filtrowanie
* formułuje kwerendy
* tworzy i modyfikuje formularze
* drukuje raporty
 |  |
| 25 | Makropolecenia VBA | * definiuje makropolecenia
* zna możliwości wbudowanego języka programowania
 |  |
| 26 | Programowanie własnych funkcji | * stosuje zasady programowania strukturalnego i obiektowego do rozwiązywania problemów
* wykorzystuje wbudowany język programowania
 |  |

**Rozkładu materiału nauczania informatyki w zakresie rozszerzonym dla liceum ogólnokształcącego i technikum – *Informatyka na czasie*, część 2(propozycja)**

**Plan wynikowy (propozycja) cz.2**

|  |  |  |
| --- | --- | --- |
| Lp. | Temat | Osiągnięcia uczniów |
| **Wymagania podstawowe. Uczeń:** | **Wymagania** **ponadpodstawowe. Uczeń:** |
| 1 | Od problemu do programu | * definiuje pojęcie specyfikacja algorytmu, określa dane i wyniki
* planuje kolejne kroki rozwiązania problemu
* omawia różne sposoby przedstawiania algorytmów (opis słowny, lista kroków, schemat blokowy, pseudokod)
* programuje i testuje rozwiązanie problemu
* sprawdza działanie algorytmów dla różnych danych
* tworzy algorytmy działania na liczbach całkowitych
* stosuje w języku C++ podstawowe konstrukcje programistyczne (operacje wejścia i wyjścia, instrukcja warunkowa, operatory matematyczne i logiczne)
* tworzy w języku C++ programy wykonujące działania na liczbach całkowitych
 | * dobiera struktury i typy danych do rodzaju problemu
* wyszukuje optymalne rozwiązania problemów
* ocenia efektywność algorytmu
* objaśnia dobrany do danego problemu algorytm, uzasadnia jego poprawność i wybór
 |
| 2 | Systemy liczbowe i reprezentacja danych w komputerze | * definiuje pojęcie pozycyjnego systemu liczbowego
* wymienia systemy liczbowe stosowane w informatyce
* definiuje pojęcia bit i bajt
* dokonuje konwersji między pozycyjnymi systemami liczbowymi, wykorzystując przy tym zależności między systemami binarnym i ósemkowym oraz binarnym i heksadecymalnym
* omawia sposób reprezentowania liczb całkowitych w komputerze
* wymienia typy danych służące do zapisu liczb całkowitych (short int, int, long int, long long int, unsigned), stosuje je w pisanych programach
* opisuje, jak w komputerze reprezentowane są znaki i napisy (char, string), odwołuje się do znaku w napisie za pomocą indeksu
* wyjaśnia, czym jest tablica kodów ASCII
* omawia działanie operacji logicznych i reprezentację ich wyników w komputerze (wynik może przyjmować wartość prawda – 1 lub fałsz – 0, co zajmuje 1 bajt pamięci)
* opisuje istotę cyfrowej reprezentacji w komputerze obrazów, dźwięków i animacji
 | * wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych z lat poprzednich lub konkursów i olimpiad informatycznych
 |
| 3 | Algorytmy zamiany reprezentacji liczb między systemami liczbowymi | * tworzy programy do konwersji między liczbami w systemach binarnym i decymalnym
* pisze programy konwertujące liczbę dziesiętną na liczbę w podanym systemie pozycyjnym
* posługuje się środowiskiem programistycznym, strukturami danych oraz językiem programowania w stopniu umożliwiającym implementację omawianych algorytmów
* stosuje binarną reprezentację liczby w algorytmie szybkiego podnoszenia do potęgi
 | * pisze programy zamieniające liczby z systemu decymalnego na system heksadecymalny
* pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zamiany: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
* uzasadnia poprawność zaproponowanego rozwiązania
* korzysta z dostępnych bibliotek w tworzonych przez siebie programach
* tworzy własne funkcje rozwiązujące problemy
 |
| 4 | Czy to jest palindrom? | * definiuje pojęcie palindromu
* określa, czy dany napis lub liczba są palindromami
* wykonuje operacje na napisach (wczytywanie napisów ze spacjami, sprawdzanie długości napisu, zamiana liter dużych na małe i odwrotnie, porównywanie znaków, znajdowanie oraz usuwanie fragmentów napisów)
* definiuje własne funkcje w języku C++, wyjaśnia celowość ich stosowania, rozróżnia parametry formalne i aktualne
* realizuje w języku C++ algorytmy sprawdzające, czy dany napis jest palindromem, oraz wyszukujące palindromy w zdaniach
* opisuje popularne funkcje oraz metody stosowane dla zmiennych typu string (toupper, tolower, size, substr, erase)
 | * pisze programy dotyczące palindromów o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* optymalizuje algorytmy i ocenia ich efektywność
 |
| 5 | Czy ta liczba jest pierwsza? | * wymienia podstawowe własności liczb pierwszych
* sprawdza, czy dana liczba jest pierwsza, stosując algorytm naiwny
* rozkłada liczbę złożoną na czynniki pierwsze
* wyznacza liczby bliźniacze
 | * implementuje algorytmy dotyczące liczb pierwszych o podwyższonym stopniu trudności: z zadań oznaczonych trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* stosuje optymalny algorytm sprawdzający, czy liczba jest pierwsza, wykorzystując funkcję logiczną; uzasadnia jego efektywność
* pisze program rozkładający liczbę złożoną na sumę dwóch liczb pierwszych (hipoteza Goldbacha)
 |
| 6 | Działania na liczbach w systemach innych niż dziesiętny | * wykonuje działania arytmetyczne na liczbach w różnych systemach pozycyjnych
* wykonuje obliczenia na dowolnie dużych liczbach, wykorzystując napisy
* wyjaśnia różnicę między operacjami na liczbach o podstawie od 1 do 9 i większej od 10
* stosuje odejmowanie w dzieleniu pisemnym liczb binarnych
* stosuje dodawanie liczby przeciwnej zapisanej w kodzie U2 przy odejmowaniu liczby binarnej
 | * wykonuje działania o podwyższonym stopniu trudności
* pisze programy wykonujące operacje arytmetyczne na liczbach w różnych systemach pozycyjnych
* optymalizuje programy, szacuje ich efektywność
 |
| 7 | Algorytm Euklidesa i działania na ułamkach | * opisuje geometryczną interpretację algorytmu Euklidesa
* pisze program realizujący algorytm Euklidesa w wersjach z dzieleniem i odejmowaniem, stosując funkcję typu void
* stosuje strukturę do reprezentacji liczb wymiernych
* wykorzystuje algorytm Euklidesa do działań na ułamkach
* stosuje zmienne lokalne i globalne, a także przekazywanie parametrów przez wartość
 | * pisze programy o podwyższonym stopniu trudności prezentujące zastosowanie algorytmu Euklidesa
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* stosuje funkcje i dobiera sposób przekazywania parametrów, jednocześnie go uzasadniając
 |
| 8 | Szyfr Cezara i inne szyfry podstawieniowe | * definiuje szyfry: podstawieniowy, monoalfabetyczny i permutacyjny, wymienia przykłady takich szyfrów
* pisze program szyfrujący informację szyfrem Cezara z wykorzystaniem liter z polskimi znakami diakrytycznymi
* omawia szyfr Vigenere’a
* stosuje w swoich programach operacje plikowe – wczytywanie danych z pliku dyskowego, zapis wyniku do pliku
 | * pisze programy szyfrujące o podwyższonym stopniu trudności
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* wykorzystuje odpowiednio dobrane struktury danych
* korzysta z funkcji bibliotecznych
* tworzy własne funkcje, dobierając sposób przekazywania parametrów
 |
| 9 | Łamiemy szyfr Cezara | * wyjaśnia, na czym polega łamanie szyfru (kryptoanaliza)
* łamie szyfr Cezara, stosując analizę częstości
* stosuje algorytmy zliczające liczbę wystąpień znaków w tekście z zastosowaniem strukturalnego typu danych – tablic
* pisze program znajdujący maksimum w tablicy i wypisujący jego pozycję (algorytm „dziel i zwyciężaj”)
 | * opisuje różne sposoby łamania szyfrów i implementuje je w języku C++
* pisze programy deszyfrujące o podwyższonym poziomie trudności
 |
| 10 | Poszukujemy liczby | * znajduje wartość w zbiorach uporządkowanym i nieuporządkowanym, stosując odpowiednio algorytmy wyszukiwania liniowego, liniowego z wartownikiem i binarnego
* pisze programy wykorzystujące przekazywanie parametru do funkcji przez wskaźnik i referencję
* stosuje algorytm „dziel i zwyciężaj” do jednoczesnego znajdowania maksimum i minimum w zbiorze
 | * pisze programy o podwyższonym stopniu trudności
* szacuje złożoność czasową zastosowanych algorytmów wyszukiwania
* wyjaśnia na przykładach różnice między różnymi sposobami przekazywania parametrów do funkcji
* podaje wzór na liczbę wykonywanych operacji w algorytmie „dziel i zwyciężaj”
 |
| 11 | Jak ocenić złożoność obliczeniową algorytmu? | * definiuje złożoność obliczeniową algorytmu
* szacuje złożoność czasową i pamięciową
* wyjaśnia, czym jest złożoność oczekiwana (średnia), optymistyczna i pesymistyczna
 | * określa złożoność czasową i pamięciową algorytmów z zastosowaniem odpowiednich wzorów
* rozróżnia pojęcia algorytmu naiwnego i optymalnego
* ocenia efektywność algorytmów
 |
| 12 | Metody sortowania prostego | * definiuje pojęcie sortowania, prawidłowo określając klucz i porządek sortowania
* definiuje pojęcia sortowania *in situ* i stabilnego
* stosuje metody sortowania prostego do sortowania liczb w zbiorze – bąbelkowe i przez wybieranie
* szacuje złożoność obliczeniową stosowanych algorytmów
* definiuje operacje kluczowe (dominujące) w algorytmach sortowania
* pisze programy realizujące poznane algorytmy sortowania
 | * pisze programy sortujące o podwyższonym stopniu trudności: sortowanie danych w plikach tekstowych, sortowanie struktur
* podaje przykłady sortowania prostego w życiu codziennym
* dobiera właściwe struktury danych
* definiuje własne funkcje do rozwiązywania problemów z wykorzystaniem algorytmów sortowania
* ocenia wpływ pierwotnego ułożenia danych w zbiorze na liczbę wykonywanych operacji
 |
| 13 | Szyfry przestawieniowe, anagramy | * omawia zasadę działania szyfrów przestawieniowych, wymienia przykłady takich szyfrów
* sprawdza, czy słowa (napisy) są anagramami
* pisze funkcje sprawdzające
* wykorzystuje poznane wcześniej algorytmy sortowania i zliczania w rozwiązywaniu problemów
 | * pisze programy szyfrujące o podwyższonym stopniu trudności
* wyszukuje anagramy w plikach tekstowych
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
 |
| 14 | Sito Eratostenesa | * opisuje algorytmy sprawdzające, czy liczba jest pierwsza
* omawia i stosuje algorytm sita Eratostenesa do wyszukiwania liczb pierwszych w określonym przedziale liczbowym
* określa złożoność obliczeniową algorytmu
 | * pisze programy o podwyższonym stopniu trudności wykorzystujące sito Eratostenesa
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* optymalizuje algorytm, dążąc do minimalnej złożoności obliczeniowej
 |
| 15 | Szukamy różnych podciągów | * definiuje pojęcia podciągu oraz podciągu spójnego
* znajduje w zbiorze podciągi o różnych własnościach
* oblicza długość najdłuższego niemalejącego spójnego podciągu oraz liczbę jego elementów
* wymienia i stosuje różne algorytmy znajdowania maksymalnej sumy elementów spójnych podciągów, oceniając ich złożoność obliczeniową
* znajduje w zbiorze spójny podciąg o maksymalnej sumie i wypisuje jego elementy
 | * pisze programy o podwyższonym stopniu trudności wyszukujące spójne podciągi
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
 |
| 16 | W poszukiwaniu lidera i idola | * definiuje pojęcia idola w grupie i lidera w zbiorze
* znajduje idola w grupie lub stwierdza jego brak
* określa, czy w zbiorze jest lider
* omawia i implementuje w języku C++ algorytmy szukania idola oraz lidera
* ocenia złożoność obliczeniową stosowanych algorytmów i ich efektywność
* stosuje tablice dwuwymiarowe w pisanych programach
* stosuje funkcję sort z biblioteki STL do wyszukiwania lidera
 | * pisze programy o podwyższonym stopniu trudności
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
 |
| 17 | Iteracja a rekurencja | * opisuje zasadę działania rekurencji
* implementuje w języku C++ algorytmy rekurencyjne, określa warunki brzegowe
* porównuje iteracyjne i rekurencyjne wersje algorytmów
* opisuje zasadę złotego podziału
* oblicza *n*-ty wyraz ciągu Fibonacciego metodami iteracyjną i rekurencyjną
* wyjaśnia, na czym polega rozszerzony algorytm Euklidesa, oraz implementuje go w języku C++
 | * pisze programy o podwyższonym stopniu trudności, np. sprawdzanie hipotezy Collatza
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
* uzasadnia wybór iteracji lub rekurencji do rozwiązania problemu
* szacuje złożoność czasową stosowanych algorytmów
* oblicza liczbę wykonywanych operacji w algorytmach rekurencyjnych
 |
| 18 | Metoda zachłanna | * wyjaśnia, na czym polega metoda zachłanna, i wymienia przykłady jej stosowania
* implementuje następujące algorytmy zachłanne: problem kasjera (wydawania reszty minimalną liczbą nominałów), problem telewidza/kinomana (optymalny harmonogram wykorzystania sali), pakowanie plecaka, wyszukiwanie optymalnej drogi
* ocenia przydatność zastosowanych algorytmów
* stosuje własne kryterium porównania w funkcji sort z biblioteki STL
 | * pisze programy o podwyższonym stopniu trudności z wykorzystaniem algorytmów zachłannych, stosując rekurencję i algorytmy z nawrotami
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
* objaśnia algorytm wybrany do rozwiązania problemu oraz ocenia jego efektywność i niedoskonałość
 |
| 19 | Programowanie dynamiczne | * wyjaśnia, na czym polega metoda dynamiczna
* implementuje optymalne algorytmy dotyczące problemu kasjera, telewidza, znajdowania drogi oraz pakowania plecaka
* stosuje metodę dynamiczną do znajdowania najdłuższego wspólnego podciągu
* porównuje metody zachłanną i dynamiczną
 | * pisze programy o podwyższonym stopniu trudności wykorzystujące algorytmy dynamiczne
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych
 |
| 20 | Dziel i zwyciężaj, czyli sortujemy sprawniej | * omawia metodę „dziel i zwyciężaj” oraz rekurencję
* wyjaśnia, na czym polega algorytm sortowania szybkiego oraz przez scalanie i implementuje je
* ocenia i porównuje złożoność czasową i obliczeniową algorytmów
 | * pisze programy o podwyższonym stopniu trudności wykorzystujące metodę „dziel i zwyciężaj” oraz algorytmy sortowania
* posługuje się środowiskiem programistycznym oraz językiem programowania w stopniu zaawansowanym
* do rozwiązania problemu dobiera optymalny algorytm i struktury danych, a także korzysta z funkcji bibliotecznych
 |

**Plan wynikowy (propozycja) cz.3**

|  |  |  |
| --- | --- | --- |
| Lp. | Temat | Osiągnięcia uczniów |
| **Wymagania podstawowe. Uczeń:** | **Wymagania** **ponadpodstawowe. Uczeń:** |
| Rozdział 1. Rozwiązywanie problemów z wykorzystaniem dynamicznych struktur danych |
| 1 | Odwrotna notacja polska (ONP) | * definiuje pojęcia: notacja infiksowa, sufiksowa, prefiksowa, drzewo wyrażenia algebraicznego
* zapisuje wyrażenia algebraiczne bez użycia nawiasów, w tym w odwrotnej notacji polskiej (ONP)
* zamienia zapis wyrażenia algebraicznego z notacji tradycyjnej na ONP i odwrotnie
* definiuje pojęcie dynamicznej struktury danych oraz stosu jako przykładu takiej struktury
* wymienia przykłady operacji, jakie można wykonać na stosie, używa struktury stos w programach
* omawia i implementuje algorytm zamiany wyrażenia algebraicznego z notacji tradycyjnej na ONP
* omawia i implementuje algorytm obliczania wartości wyrażenia arytmetycznego zapisanego w ONP z wykorzystaniem rekurencji oraz stosu
 | * wykonuje zadania o podwyższonym stopniu trudności: oznaczone trzema gwiazdkami w podręczniku, z arkuszy maturalnych, z konkursów i olimpiad informatycznych oraz ze zbioru zadań CKE
* optymalizuje programy, szacuje ich efektywność
 |
| 2 | Znajdowanie drogi wyjścia z labiryntu | * omawia kolejkę jako kolejny przykład dynamicznej struktury danych i ją deklaruje
* wymienia przykładowe operacje na kolejce i je stosuje
* wyjaśnia algorytm przeszukiwania z nawrotami
* definiuje pojęcie manipulator strumienia
* omawia algorytm znajdowania wyjścia z labiryntu z wykorzystaniem rekurencji
* stosuje kolejkę w iteracyjnym algorytmie znajdującym najkrótszą drogę wyjścia z labiryntu
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* wskazuje różnice między algorytmami znajdowania wyjścia z labiryntu z wykorzystaniem rekurencji i iteracji
* uzasadnia użycie kolejki w algorytmie wyznaczania najkrótszej drogi wyjścia z labiryntu
 |
| 3 | Wykorzystanie list w rozwiązywaniu problemów | * definiuje pojęcie listy, w tym listy jednokierunkowej, dwukierunkowej, cyklicznej, wyjaśnia, co oznacza sekwencyjny dostęp do danych na liście
* wyróżnia przykładowe operacje na liście i je stosuje
* omawia problem Flawiusza, przeprowadza jego symulację
* definiuje pojęcia porządek leksykograficzny, sortowanie leksykograficzne, sortowanie kubełkowe, sortowanie stabilne
* omawia i implementuje algorytm porządkujący słowa leksykograficznie
* wykorzystuje pliki tekstowe do wczytywania danych do programu i zapisywania wyników
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* szacuje złożoność czasową poznanych algorytmów, optymalizuje je
 |
| 4 | Grafy. Znajdowanie najkrótszej drogi | * definiuje graf, wymienia elementy składowe grafu i rodzaje grafów
* zna sposoby reprezentacji grafu: macierz sąsiedztwa, listy sąsiedztwa
* stosuje typ **vector** do reprezentacji grafu w postaci list sąsiedztwa
* stosuje metody dla klasy vector
* wczytuje dane do programu z plików tekstowych
* definiuje własne nazwy typów zmiennych
* omawia algorytm przeszukiwania grafu w głąb (DFS)
* omawia algorytm przeszukiwania grafu wszerz (BFS)
* wyjaśnia, do czego służy algorytm Dijkstry
 | * implementuje algorytmy przeszukiwania grafu w głąb (DFS) oraz przeszukiwania grafu wszerz (BFS)
* omawia i implementuje algorytm Dijkstry
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* szacuje czasową złożoność algorytmów
 |
| Rozdział 2. Algorytmy numeryczne |
| 5 | Reprezentacja liczb rzeczywistych w komputerze | * znajduje rozwinięcie binarne nieskracalnego ułamka właściwego
* wyjaśnia, które ułamki właściwe mają skończone rozwinięcie binarne, a które okresowe
* omawia różnice między stałoprzecinkową a zmiennoprzecinkową reprezentacją liczb rzeczywistych w komputerze
* wyjaśnia pojęcia: cecha, mantysa, postać znormalizowana
* zapisuje liczby w postaci znormalizowanej
* definiuje pojęcia: liczba pojedynczej precyzji, liczba podwójnej precyzji, kod z nadmiarem
* wykonuje działania na liczbach zmiennoprzecinkowych
 | * znajduje reprezentację liczby zapisanej w systemie dziesiętnym jako liczby pojedynczej i podwójnej precyzji
* świadomie używa typów **float** i **double** w zadaniach
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* interpretuje wpływ zastosowanych typów na uzyskane wyniki
 |
| 6 | Błędy w obliczeniach | * rozróżnia pojęcia błąd względny i błąd bezwzględny
* omawia przyczyny i rodzaje błędów w obliczeniach komputerowych – błąd reprezentacji, zaokrąglenia, przybliżenia, obcięcia
* wskazuje różnice między algorytmem stabilnym i niestabilnym
* znajduje pierwiastki równania kwadratowego algorytmem stabilnym i algorytmem niestabilnym
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* optymalizuje programy, stosując algorytmy stabilne oraz unikając błędów w obliczeniach
 |
| 7 | Obliczanie wartości wielomianu | * wyjaśnia schemat Hornera
* implementuje algorytm obliczający wartość wielomianu algorytmem naiwnym oraz algorytm obliczający wartość wielomianu z zastosowaniem schematu Hornera
 | * stosuje schemat Hornera do zamiany liczby w systemie pozycyjnym o wybranej podstawie na liczbę dziesiętną oraz do szybkiego podnoszenia do potęgi
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 8 | Metody obliczeń przybliżonych | * wyjaśnia, na czym polegają metody obliczeń przybliżonych
* znajduje metodą bisekcji miejsce zerowe funkcji w zadanym przedziale
* oblicza pierwiastek kwadratowy metodą bisekcji i metodą Newtona–Raphsona, porównuje obie metody
* oblicza pola obszarów zamkniętych metodą prostokątów oraz metodą trapezów
 | * implementuje algorytmy numeryczne: znajdowania miejsc zerowych funkcji oraz obliczania pierwiastka kwadratowego metodą bisekcji, obliczania pierwiastka kwadratowego metodą Newtona−Raphsona, obliczania pola obszaru zamkniętego metodą prostokątów i metodą trapezów, znajdowania przybliżenia liczby pi oraz symulacji ruchów Browna metodą Monte Carlo
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 9 | Algorytmy badające własności geometryczne | * bada położenie punktów względem prostej i odcinka
* sprawdza, czy dwa odcinki się przecinają
* bada przynależność punktu do figury
 | * implementuje algorytmy badające własności geometryczne
* w algorytmach badających własności geometryczne wykorzystuje macierz oraz regułę Sarrusa do obliczania wyznacznika macierzy
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 10 | Fraktale | * wyjaśnia, czym jest fraktal
* wskazuje przykłady struktur fraktalnych występujących w przyrodzie
* podaje przykłady fraktali (zbiór Cantora, drzewo binarne, płatek Kocha, dywan Sierpińskiego) i wyjaśnia sposób tworzenia tych fraktali
 | * implementuje w języku JavaScript algorytmy generujące fraktale danego stopnia
* stosuje metodę IFS do tworzenia fraktali w arkuszu kalkulacyjnym
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| Rozdział 3. Zaawansowane algorytmy i techniki programistyczne |
| 11 | Wyszukiwanie wzorca w tekście | * definiuje problem wyszukiwania wzorca w tekście
* wyszukuje wzorzec w tekście algorytmem naiwnym
* wyjaśnia, na czym polega metoda haszowania, i czym są: funkcja haszująca, klucz, hasz, kolizja
 | * omawia algorytm Karpa−Rabina do wyszukiwania wzorca w tekście z zastosowaniem funkcji haszującej
* stosuje funkcję haszującą oraz algorytm Karpa−Rabina w programach wyszukujących wzorzec w tekście
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* ocenia czasową złożoność obliczeniową algorytmów
 |
| 12 | Szyfrowanie z kluczem publicznym. Algorytm RSA | * wskazuje różnice między kryptografią symetryczną i kryptografią asymetryczną, definiuje pojęcia klucz prywatny i klucz publiczny
* wyjaśnia, do czego służy algorytm RSA, i wyróżnia główne etapy tego algorytmu (generowanie kluczy, szyfrowanie z kluczem publicznym oraz deszyfrowanie z kluczem prywatnym)
* wyjaśnia, jak generuje się klucze publiczny i prywatny oraz jak wykorzystuje się je do szyfrowania i deszyfrowania informacji w algorytmie RSA
 | * pisze program generujący klucz prywatny i klucz publiczny w algorytmie RSA
* pisze programy szyfrujące i deszyfrujące informacje w algorytmie RSA
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 13 | Programowanie obiektowe | * wyjaśnia, na czym polega programowanie strukturalne, czym charakteryzują się metoda zstępująca i metoda wstępująca
* definiuje programowanie obiektowe i podstawowe pojęcia z nim związane
* definiuje własne klasy korzystając ze specyfikatorów dostępu
* deklaruje konstruktory w klasach
* wyjaśnia, na czym polega przeciążanie metod
 | * definiuje i implementuje własne klasy, obiekty, atrybuty i metody
* wyjaśnia, na czym polega polimorfizm i czym są metody wirtualne
* stosuje hierarchię klas
* wyjaśnia, na czym polega hermetyzacja danych i jakie jest zastosowanie operatora zasięgu
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| Rozdział 4. Relacyjne bazy danych |
| 14 | Wprowadzenie do relacyjnych baz danych | * wyjaśnia, czym jest baza danych, oraz definiuje podstawowe pojęcia z nią związane: rekord, pole rekordu, tabela bazy danych, atrybut, relacja, klucz główny, klucz obcy, redundancja, formularz, kwerenda, raport, system zarządzania bazą danych
* rozróżnia typy relacji w bazach danych
* wprowadza i modyfikuje dane w tabelach
* wyszukuje informacje w bazach danych, stosując filtrowanie i kwerendy
* stosuje formularze do wprowadzania i modyfikowania danych
* eksportuje dane, wykorzystując raporty
* importuje dane do tabel
 | * projektuje i tworzy relacyjne bazy danych
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 15 | Wykorzystanie danych pochodzących z kwerend | * definiuje pojęcia: tabela pomostowa, klucz złożony, kwerenda wybierająca, kwerenda krzyżowa
* łączy dane w bazach za pomocą tabeli pomostowej
* stosuje kwerendy krzyżowe i wybierające
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 16 | Podstawy języka SQL | * opisuje przeznaczenie języka SQL, wymienia podstawowe klauzule języka
* wykorzystuje język SQL do tworzenia i usuwania baz danych, dodawania tabel do baz danych, usuwania tabel z baz, dodawania rekordów do tabel, importowania danych do tabel, edycji rekordów
* wyjaśnia działanie baz danych w architekturze klient–serwer
* tworzy konta użytkowników i przydziela im uprawnienia do wybranej bazy
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 17 | Zapytania w języku SQL | * zna zasady tworzenia zapytań do bazy
* formułuje zapytania zwracające określone dane
* sortuje wyniki zapytań
* eksportuje wyniki zapytania do pliku tekstowego
 | * formułuje zapytania w języku SQL, stosując selekcję, sortowanie, projekcję oraz agregowanie danych
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| 18 | Tworzenie aplikacji korzystającej z sieciowej bazy danych | * wyróżnia etapy pracy nad aplikacją internetową
* wyróżnia technologie back-end i front-end
* wymienia kluczowe zadania w procesie tworzenia aplikacji
* określa założenia projektu, jego funkcjonalność
* instaluje i konfiguruje niezbędne oprogramowanie
* przygotowuje bazę danych na potrzeby projektu
 | * tworzy aplikacje internetowe z przejrzystym interfejsem użytkownika korzystające z sieciowej bazy danych, samodzielnie je testuje i wprowadza poprawki
* wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
 |
| P1 | Pułapki cyfrowego świata | * wyjaśnia, czym jest dokumentacja projektu, bierze czynny udział w jej tworzeniu
* definiuje cel projektu
* wie, czym jest dyskusja panelowa
* aktywnie uczestniczy w realizacji projektu, wykorzystując specjalistyczne narzędzia do gromadzenia, opracowania i prezentacji danych oraz prowadzenia spotkań online
 | * przyjmuje rolę lidera odpowiedzialnego za zespół i projekt
* przydziela zadania, nadzoruje pracę innych
* przyjmuje funkcję eksperta lub moderatora
 |
| Rozdział 5. Rozwiązywanie różnych problemów z wykorzystaniem komputera |
| 19 | Sterujemy robotem | * definiuje pojęcie robota
* omawia budowę oraz wybrane parametry robotów (serwomotor, magnetometr, akcelerometr, diody, czujniki, wyświetlacz)
* programuje roboty wykorzystując specjalistyczne narzędzia (aplikacje), w tym symulatory online
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* wykazuje się kreatywnością przy projektowaniu własnych projektów, takich jak np.: stacja pogodowa, gry logiczne i zręcznościowe, mierzenie odległości od przeszkód, loty synchroniczne (drony)
* stosuje aplikacje mobilne do sterowania robotami
 |
| 20 | Sztuka publikowania w sieci | * opracowuje interesujące treści internetowe dostosowane do potrzeb potencjalnych odbiorców, wykorzystując zasadę 5W
* dba o identyfikację wizualną i dźwiękową
* stosuje elementy przyciągające uwagę użytkowników, np. lidy, hastagi, właściwie dobiera słowa
* korzysta z narzędzi graficznych i multimedialnych do wzbogacania treści
* rozróżnia pojęcia webcast, webinarium, screencast i podcast
* montuje materiały, wykorzystując specjalistyczne oprogramowanie (np. Stream z pakietu Office 365)
* dba o właściwy rozmiar materiałów, stosując różne rodzaje kompresji oraz właściwy format plików
* występuje przed kamerą i mikrofonem, przekazując treści w sposób atrakcyjny dla odbiorców, utrzymujący ich uwagę
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* tworzy podcasty i publikacje wideo na wybrane tematy wymagające dużego nakładu pracy (np. promocja czy jubileusz szkoły, szkolny festiwal kultury lub nauki), korzysta z zaawansowanych narzędzi
 |
| 21 | Grafiki informacyjne | * wymienia różne sposoby przedstawiania informacji
* definiuje pojęcie grafiki informacyjnej, wymienia przykłady grafiki narracyjnej i wizualizacji danych
* tworzy infografikę z wykorzystaniem języka piktogramów Isotype
* poprawnie projektuje proste infografiki zawierające uporządkowane informacje (chronologicznie, według kategorii, przestrzenne czy hierarchiczne), umiejętnie stosuje tekst i obraz
* wymienia, co powoduje wrażenie chaosu na infografice
* przy projektowaniu świadomie dobiera barwy i wykorzystuje funkcje kolorów
* w swoich projektach zwraca uwagę na dostosowanie treści do odbiorców
 | * wykonuje zadania o podwyższonym stopniu trudności, oznaczone trzema gwiazdkami w podręczniku
* wykazuje się kreatywnością, tworząc infografiki dotyczące globalnych problemów współczesnego świata, lokalnych, szkolnej społeczności czy też środowisk młodzieżowych
 |