



Zajęcia dodatkowe dla Uczniów Szkoły

III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Tytuł zajęć

„Zajęcia rozwijające pasje i zainteresowanie informatyką.”

Autor/Autorzy opracowania

mgr Anna Jaborkhel

Niniejszy skrypt/scenariusz powstał na potrzeby realizacji Projektu

nr RPWP.08.01.04-30-0005/19 pn.:

„ENIGMA – Wsparcie nauczania matematyki i informatyki

w szkołach podstawowych i ponadpodstawowych

Metropolii Poznań”

Poznań 2021

PROGRAM ZAJĘĆ

L.p.	Temat zajęć	Liczba godzin
1.	Systemy liczbowe i reprezentacja danych w komputerze.	5
2.	Metody sortowania.	3
3.	Szukamy różnych podciągów.	3
4.	Tworzenie stron internetowych - HTML.	3
5.	Tworzenie stron internetowych – CSS (<i>Cascading Style Sheets</i>).	4
6.	Projektowanie stron z wykorzystaniem WordPress.	5
Łączna liczba godzin		23

Spis treści:

Wybrane algorytmy programistyczne.

1. Systemy liczbowe i reprezentacja danych w komputerze.

- 1.1. Systemy liczbowe stosowane w informatyce - reprezentacja liczb całkowitych, znaków i napisów w komputerze .
- 1.2. Algorytm zamiany reprezentacji liczby z dziesiętnej na dwójkową.
- 1.3. Algorytm zamiany reprezentacji liczby dwójkowej na dziesiętną.
- 1.4. Uogólnienie algorytmów zamiany reprezentacji liczb między systemami pozycyjnymi.
- 1.5. Sprawdzamy czy wyraz jest palindromem.
- 1.6. Sprawdzamy czy dwa wyrazy są anagramami.

2. Metody sortowania.

- 2.1. Sortowanie bąbelkowe i przez wybieranie.
- 2.2. Sortowanie szybkie.
- 2.3. Sortowanie przez scalanie.

3. Szukamy różnych podciągów.

- 3.1.1. Szukamy długości najdłuższego spójnego podciągu niemalejącego.
- 3.1.2. Szukamy maksymalnej sumy podciągu spójnego.
- 3.1.3. Szukamy podciągu spójnego o maksymalnej sumie.

Tworzenie stron internetowych.

4. HTML

- 4.1.1. Podstawowe znaczniki HTML – szkielet strony.
- 4.1.2. Formatowanie treści strony za pomocą znaczników HTML.
- 4.1.3. Listy, odsyłacze i grafika na stronach WWW.

5. CSS (*Cascading Style Sheets*).

- 5.1.1. Podstawy CSS.
- 5.1.2. Źródła stylów.
- 5.1.3. Kaskadowość, dziedziczenie, specyficzność
- 5.1.4. Selektory.

6. Projektowanie stron z wykorzystaniem WordPress

- 6.1.1. Instalacja WordPress
- 6.1.2. Obsługa panelu administracyjnego.
- 6.1.3. Obsługa motywów i wtyczek.



I scenariusz zajęć

Systemy liczbowe i reprezentacja danych w komputerze

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 3

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- stosuje algorytmy na liczbach: zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi,
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- przedstawia sposoby reprezentowania w komputerze znaków, liczb, obrazów, dźwięków, animacji.
- objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych,
- stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: bit i bajt, system pozycyjny, kod ASCII, Unicode,
- wymienia systemy liczbowe wykorzystywane w informatyce.

Zrozumienie

Uczeń:

- wyjaśnia, jak reprezentowane są w komputerze liczby całkowite, znaki, napisy, obrazy,

Zastosowanie

Uczeń:



- konwertuje liczby zapisane w systemach: binarnym, szesnastkowym, ósemkowym oraz systemach o podstawie
- zapisuje liczby w kodzie uzupełnień do 2 (U2),

Analizowanie

Uczeń:

- dostrzega zaletę zapisu liczb w kodzie U2,
- sprawdza poprawność podanego zdania logicznego.

Wartościowanie, testowanie, ocenianie

Uczeń:

- określa, ile informacji można przechować na określonej liczbie bitów.

Tworzenie

Uczeń:

- zapisuje kody źródłowe programów opartych na poznanych systemach liczbowych,
- zapisuje kody źródłowe programów wykorzystujących zdania logiczne.

Środki dydaktyczne:

- komputer z zainstalowanym środowiskiem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dlauczyciela.pl: Wydawnictwa Nowa Era
 1. *Kody ASCII i Unicode w języku C++*
 2. *Skrypt z teorią i zadaniami* (

Metody pracy:

- wykład konwersatoryjny (MWK),
- ćwiczenia (MC),
- metoda analizy, opisu i algorytmu (MOA),
- metoda analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji z działu *Systemy liczbowe*.

II. Faza zasadnicza	
1. Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
2. Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami.	
3. Temat proponuje podzielić w następujący sposób: Lekcja 1 : zagadnienie 1.1 skrypt str. 4-28 z ćwiczeniami. Lekcje 2 : zagadnienie 1.2 skrypt str. 11 – 13, 29-30 Lekcja 3 : zagadnienie 1.3 i 1.4 skrypt str. 14 – 18, 32 – 37.	
4. Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
5. Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Podczas omawiania modelu RGB warto wspomnieć o tym, jak zbudowany jest piksel w wyświetlaczach różnych urządzeń. Piksel składa się zwykle z trzech subpikseli, emitujących światło w kolorach czerwonym, zielonym i niebieskim. Stąd model RGB określa natężenie tych właśnie kolorów, a przypisane im wartości są niczym innym, jak informacją dla piksela, z jakim natężeniem na emitować światło każdy z jego subpikseli.

2. Podczas wprowadzania szesnastkowego systemu liczbowego warto pokazać jego praktyczne zastosowanie. System ten jest obecny w programowaniu, na przykład podczas określania kolorów. W wielu językach programowania opisujemy kolory w modelu RGB, jednak natężenia kolorów podawane są w notacji heksadecymalnej. Przykładem wykorzystania systemu szesnastkowego jest też zapis kolorów w kodach HTML stron internetowych.
3. Różne notacje dla kolorów w modelu RGB można zobaczyć, korzystając z narzędzia dostępnego bezpośrednio w wyszukiwarce google.com po wpisaniu frazy “RGB color picker”.
4. Rozmawiając o typach danych, warto podkreślić, że język C++ jest językiem statycznie typowanym. To znaczy, że programista sam określa typ zmiennych, tworząc je. W przypadku niektórych języków istnieje typowanie dynamiczne. Oznacza ono, że typ jest nadawany zmiennej na podstawie przypisanej do niej wartości. Przykładem takiego języka jest Python.
5. Istnienie różnych typów danych to ważny element realizacji tematu. W ramach podkreślenia różnic pomiędzy nimi można zrealizować ćwiczenia polegające na stworzeniu kodu źródłowego programu zawierającego zmienne w różnych typach i próbach przypisania do zmiennych danych innego typu. Na przykład ciągu znaków do zmiennej typu *integer*.
6. Wprowadzając typ danych string, który jest typem obiektowym, dokonujemy wprowadzenia do programowania obiektowego.
7. W ramach podsumowania materiału dotyczącego przechowywania danych, zarówno zmiennych przechowywanych przez program, jak i przechowywania dźwięków, obrazów, czy filmów, można zrealizować ćwiczenie polegające na tym, aby uczniowie na podstawie nazwy typu danych lub formatu wskazali, jaki rodzaj danych jest przez niego przechowywany.
8. Realizując ćwiczenia i zadania polegające na stworzeniu kodu źródłowego o określonym działaniu, warto podkreślić, by uczniowie przed przystąpieniem do pisania kodu źródłowego pamiętali o wcześniejszych etapach rozwiązywania problemów z wykorzystaniem komputera – na przykład sformułowaniu algorytmu.
9. Wykorzystując zmienne typu string w języku C++, do kodu źródłowego dodaje się dyrektywę `#include <string>`. Warto podkreślić, że **string** i **iostream** nie są jedynymi bibliotekami, z jakich można korzystać w tym języku. Biblioteki znacznie zwiększają możliwości języka.

II scenariusz zajęć

Sprawdzamy czy wyraz jest palindromem.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 1

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: palindrom, funkcja, wartownik, parametr formalny, parametr aktualny.

Zrozumienie

Uczeń:

- wyjaśnia, czym jest parametr funkcji,
- opisuje zadanie wartownika w kodzie źródłowym.

Zastosowanie

Uczeń:

- korzysta z metod dostępnych dla typu string w tworzonych programach,
- w kodach źródłowych wykorzystuje wczytywanie wierszy danych wraz ze spacjami.

Analizowanie

Uczeń:

- opisuje zalety wynikające z zapisywania podproblemów w postaci funkcji.



Wartościowanie, testowanie, ocenianie

Uczeń:

- odróżnia parametry formalne i aktualne funkcji.

Tworzenie

Uczeń:

- tworzy i wywołuje funkcje w języku C++.

Środki dydaktyczne:

- komputer z oprogramowaniem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dlanauczyciela.pl:
 1. *Typy zmiennych w języku C++*
 2. *Instrukcja warunkowa w języku C++*
 3. *Pętla for w języku C++*
 4. *Pętla while w języku C++*
 5. *Pętla do while w języku C++*
 6. *Tablice w języku C++*
 7. *Funkcje w języku C++*
 8. *Podstawowe biblioteki języka C++*
- *Skrypt z teorią i zadaniami*

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji.

II. Faza zasadnicza	
Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
Wyjaśnienie, czym jest palindrom. Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami.	
Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzebranych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Wprowadzeniem do lekcji powinno być wyjaśnienie, czym jest palindrom. Można to zrealizować, polecając uczniom, aby samodzielnie znaleźli w internecie słownikową definicję słowa *palindrom*.
2. Podczas omawiania funkcji na podstawie kodów źródłowych języka C++ warto podkreślić, że funkcje występują też w innych językach programowania. Warto również zaznaczyć, że jest to jedna z najważniejszych i najczęściej wykorzystywanych konstrukcji programistycznych.
3. Podczas omawiania metod dla obiektów typu string warto podkreślić, że ich nazwy wywodzą się od angielskich słów opisujących, do czego służą konkretne metody.
4. Omawiając działanie metod `toupper` i `tolower`, warto podkreślić, że zwracają one nie litery, a kody ASCII znaków.
5. Przykłady palindromów można odszukać na portalu palindromy.pl.

6. Warto i poszukać z uczniami innych niż przedstawiony algorytmów sprawdzania, czy słowo jest palindromem. Można zasugerować uczniom implementację rozwiązania sprawdzającego pary liter nie od krańców, a od środka słowa. Dobrze jest wspierać uczniów podczas tworzenia takiego algorytmu oraz jego implementacji.
7. Tematykę gier słownych można rozszerzyć, realizując inne zadania. Przykładem może być kod źródłowy programu, który sprawdza, czy dwa podane słowa rymują się. Warto podjąć z uczniami próbę stworzenia takiego kodu oraz zachęcić ich do poszukiwania i realizacji podobnych pomysłów.

III scenariusz zajęć

Sprawdzamy czy dwa wyrazy są anagramami.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 1

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy poznane w szkole podstawowej oraz algorytmy na tekstach: porównywania tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową,
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych,
- stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcie: anagram.

Zrozumienie

Uczeń:

- wyjaśnia, w jaki sposób sprawdzamy czy dwa wyrazy są anagramami.

Zastosowanie

Uczeń:

- implementuje algorytmy z wykorzystaniem sortowania i zliczania,

Analizowanie

Uczeń:

- sprawdza, czy dwa słowa są anagramami.

Wartościowanie, testowanie, ocenianie

Uczeń:

- rozróżnia algorytmy sprawdzające, czy podane słowa są anagramami, i określa czasową złożoność obliczeniową tych algorytmów.

Tworzenie

Uczeń:

- tworzy i wywołuje w języku C++ funkcje implementujące algorytmy sprawdzające, czy podane słowa są anagramami.

Środki dydaktyczne:

- komputer z oprogramowaniem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dlauczyciela.pl:
 9. Typy zmiennych w języku C++
 10. Instrukcja warunkowa w języku C++
 11. Pętla for w języku C++
 12. Pętla while w języku C++
 13. Pętla do while w języku C++
 14. Tablice w języku C++
 15. Funkcje w języku C++
 16. Podstawowe biblioteki języka C++
- Skrypt z teorią i zadaniami

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji..

II. Faza zasadnicza	
Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none">• sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej• sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego• rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
Wyjaśnienie, czym jest anagram. Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. Skrypt str. 41 - 45	
Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Warto polecić uczniom, aby zastanowili się, czy opisane algorytmy sprawdzania anagramów są jedynymi algorytmami tego typu. Mogą oni spróbować rozwiązać takie zadanie, wykorzystując jeszcze inny algorytm.
2. Istnieją internetowe narzędzia generujące anagramy dla podanych słów według określonych wcześniej założeń. Przykładem takiego narzędzia jest wordlist.eu/anagramy.
3. Dla utrwalenia informacji związanych z anagramami polecam przeprowadzenie dodatkowego ćwiczenia na platformie SPOJ: pl.spoj.com/problems/AL_30_01/. Polega ono na zliczeniu, ile z podanych słów jest anagramami wybranego słowa.
4. Kolejnym dodatkowym zadaniem z platformy SPOJ dotyczącym anagramów, które można zrealizować w ramach rozszerzenia tematu, jest: pl.spoj.com/problems/AL_23_02/. Polega ono na sprawdzeniu, czy dana para słów to anagramy, i ewentualnym określeniu, ile trzeba

str. 14



wykonać operacji, aby para słów stała się anagramami. Warto podkreślić, że zadanie ma wysoki poziom zaawansowania.

IV scenariusz zajęć Metody sortowania – bąbelkowe i wybieranie.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 1

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- stosuje przy rozwiązywaniu problemów z różnych dziedzin algorytmy porządkowania ciągu liczb: przez wybieranie i metodą bąbelkową
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych,
- w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą,
- dyskutuje na temat roli myślenia komputacyjnego i jego metod, takich jak: abstrakcja, reprezentacja danych, dekompozycja problemu, redukcja, myślenie rekurencyjne, podejście heurystyczne w rozwiązywaniu problemów z różnych dziedzin,
- stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: sortowanie, operacja dominująca.

Zrozumienie

Uczeń:

- wyjaśnia, czym jest sortowanie bąbelkowe i sortowanie przez wybieranie.

Zastosowanie

Uczeń:

- stosuje algorytmy sortowania prostego.

Analizowanie

Uczeń:

- szacuje czasową złożoność obliczeniową algorytmów sortowania prostego.

Wartościowanie, testowanie, ocenianie

Uczeń:

- rozróżnia algorytmy sortowania prostego.

Tworzenie

Uczeń:

- tworzy i wywołuje funkcje w języku C++ implementujące algorytmy sortowania prostego.

Środki dydaktyczne:

- komputer z oprogramowaniem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dLANauczyciela.pl:
 - 17. Typy zmiennych w języku C++
 - 18. Instrukcja warunkowa w języku C++
 - 19. Pętla for w języku C++
 - 20. Pętla while w języku C++
 - 21. Pętla do while w języku C++
 - 22. Tablice w języku C++
 - 23. Funkcje w języku C++
 - 24. Podstawowe biblioteki języka C++
- Skrypt z teorią i zadaniami (

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji..

II. Faza zasadnicza	
1. Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
2. Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. a. Skrypt str. 46 - 52	
3. Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
4. Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Sortowanie bąbelkowe jest elementem jednego z zadań w portalu SPOJ – można polecić uczniom rozwiązanie go: pl.spoj.com/problems/FR_01_09. Portal SPOJ to strona przeznaczona do rozwiązywania algorytmicznych zadań programistycznych, gdzie jednym z kryteriów oceny zadania jest odpowiedni czas wykonania algorytmu.
2. W sortowaniu przez wybieranie w kolejnych krokach znajduje się najmniejszy element sortowanego ciągu i przenosi ten element na odpowiednią pozycję ciągu wynikowego (przez zamianę elementów miejscami). Można polecić uczniom odwrócenie tego algorytmu, czyli wyszukiwanie największego elementu i wstawianie go na końcu listy.
3. Warto zadać pytanie, czy w ten sposób liczba porównań będzie taka sama.
4. W ramach rozszerzenia tematu sortowania można polecić uczniom przygotowanie własnego algorytmu sortowania, który działa inaczej niż 3 algorytmy przedstawione podczas lekcji.

5. Podczas realizacji algorytmów sortowania zachodzi konieczność podmiiany miejscami dwóch wartości w tablicy, można to zrobić z wykorzystaniem zmiennej pomocniczej jak przedstawiono to w przykładowych kodach źródłowych. Można zachęcić uczniów do przygotowania swoich programów bez wykorzystywania zmiennej pomocniczej.
6. Warto, aby uczniowie porównali poznane algorytmy sortowania pod względem czasowej złożoności obliczeniowej zarówno podczas sortowania niewielkiej, jak i dużej liczby elementów w zbiorze. W internecie można znaleźć wiele filmów i animacji pomagających lepiej zrozumieć metody sortowania prostego. Polecamy np. sortowanie bąbelkowe wykonane przez węgierską grupę taneczną: [youtube.com/watch?v=lyZQPjUT5B4](https://www.youtube.com/watch?v=lyZQPjUT5B4).

V scenariusz zajęć

Metody sortowania – sortowanie szybkie i sortowanie przez scalanie.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 2

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- wyróżnia w problemie podproblemy i charakteryzuje: metodę połowienia, stosuje podejście zachłanne i rekurencję,
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych,
- w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą,
- dyskutuje na temat roli myślenia komputacyjnego i jego metod, takich jak: abstrakcja, reprezentacja danych, dekompozycja problemu, redukcja, myślenie rekurencyjne, podejście heurystyczne w rozwiązywaniu problemów z różnych dziedzin,
- stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
- sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów,
- zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu i implementuje w wybranym języku programowania algorytmy poznane na wcześniejszych etapach oraz algorytm sortowania ciągu liczb przez scalanie,
- objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności metodę „dziel i zwyciężaj” (jednoczesne znajdowanie minimum i maksimum, sortowanie przez scalanie i szybkie)

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: metoda „dziel i zwyciężaj”, element dzielący.

Zrozumienie

Uczeń:

- stosuje w algorytmach sortowania metodę „dziel i zwyciężaj” oraz rekurencję.

Zastosowanie

Uczeń:

- stosuje algorytmy sortowania prostego.

Wartościowanie, testowanie, ocenianie

Uczeń:

- szacuje złożoność czasową i pamięciową algorytmów sortowania szybkiego i sortowania przez scalanie.

Tworzenie

Uczeń:

- tworzy funkcje scalające ciągi bez używania tablic pomocniczych.

Środki dydaktyczne:

- komputer z oprogramowaniem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dLANauczyciela.pl:
 - 25. Typy zmiennych w języku C++
 - 26. Instrukcja warunkowa w języku C++
 - 27. Pętla for w języku C++
 - 28. Pętla while w języku C++
 - 29. Pętla do while w języku C++
 - 30. Tablice w języku C++
 - 31. Funkcje w języku C++
 - 32. Podstawowe biblioteki języka C++
- Skrypt z teorią i zadaniami (

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji..

II. Faza zasadnicza	
1. Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
2. Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. a. Skrypt str. 53 - 63	
3. Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
4. Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Warto, aby uczniowie w ramach przygotowania do lekcji przejrzyli kody źródłowe programów napisanych podczas realizacji tematu 2.1. Pozwoli im to przypomnieć sobie podstawowe konstrukcje programistyczne pomagające sortować elementy zbioru.
2. W ramach rozszerzenia treści warto, aby uczniowie wykonali zadanie, które będzie polegało na posortowaniu nie liczb, a liter. Wykorzystać powinni jeden z algorytmów sortowania omawianych w temacie.
3. Uczniowie lepiej zrozumieją zagadnienie sortowania, oglądając jedną z wielu dostępnych w internecie animacji przedstawiających konkretne sortowanie. Takie animacje dostępne są w encyklopedii internetowej Wikipedia oraz w serwisie YouTube.



4. Modyfikacją algorytmu sortowania szybkiego może być wyznaczenie mediany dla określenia elementu dzielącego. Warto omówić zagadnienie wyznaczenia mediany i zlecić uczniom napisanie programu realizującego to zagadnienie.

VI scenariusz zajęć Szukamy różnych podciągów.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 3

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
 - sprawdza poprawność działania algorytmów dla przykładowych danych,
 - projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
 - objaśnia sposoby wykonywania przez komputer działań arytmetycznych i operacji logicznych,
 - w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą,
 - dyskutuje na temat roli myślenia komputacyjnego i jego metod, takich jak: abstrakcja, reprezentacja danych, dekompozycja problemu, redukcja, myślenie rekurencyjne, podejście heurystyczne w rozwiązywaniu problemów z różnych dziedzin,
 - stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów,
 - sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów,
 - wykorzystuje znane sobie algorytmy przy rozwiązywaniu i programowaniu rozwiązań następujących problemów: znajdowania w ciągu podciągów o różnorodnych własnościach, np. najdłuższego spójnego podciągu niemalejącego, spójnego podciągu o największej sumie
- Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: podciąg, podciąg spójny, podciąg niemalejący, liczba trójkątna.

Zrozumienie

Uczeń:

- wyjaśnia, na czym polega poszukiwanie podciągów w ciągu.



Zastosowanie

Uczeń:

- znajduje w ciągu podciąg o maksymalnej sumie elementów.

Analizowanie

Uczeń:

- analizuje złożoność czasową algorytmów znajdowania maksymalnej sumy podciągu.

Wartościowanie, testowanie, ocenianie

Uczeń:

- ocenia, czy dany podciąg jest podciągiem spójnym.

Tworzenie

Uczeń:

- tworzy i wywołuje w języku C++ funkcje szukające podciągu o maksymalnej sumie elementów.

Środki dydaktyczne:

- komputer z oprogramowaniem Code::Blocks,
- projektor multimedialny lub tablica multimedialna,
- materiały multimedialne dostępne na stronie dlauczyciela.pl:
 - 33. *Typy zmiennych w języku C++*
 - 34. *Instrukcja warunkowa w języku C++*
 - 35. *Pętla for w języku C++*
 - 36. *Pętla while w języku C++*
 - 37. *Pętla do while w języku C++*
 - 38. *Tablice w języku C++*
 - 39. *Funkcje w języku C++*
 - 40. *Podstawowe biblioteki języka C++*
- *Skrypt z teorią i zadaniami* (

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji.

II. Faza zasadnicza	
Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. Skrypt str. 64 - 72	
Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Temat można zrealizować w czasie trzech godzin lekcyjnych. Proponuje następujący podział:
2. Lekcja 1: zagadnienie 3.1
 - a. Lekcja 2: zagadnienie 3.2
 - b. Lekcja 3: zagadnienie 3.3
3. Aby pomóc uczniom zrozumieć, czym jest podciąg, proponujemy zadanie im kilku pytań typu *prawda/falsz*. Nauczyciel podaje jakiś ciąg, a następnie kilka podciągów. Uczniowie wskazują, które podciągi są podciągami danego ciągu.
4. Zagadnieniem szczególnie ważnym podczas omawiania tego tematu jest przekazywanie danych przez referencję. Warto powtórzyć to zagadnienie przed przystąpieniem do rozwiązywania ćwiczeń.

5. Proponuje następujące ćwiczenie. Polega ono na znalezieniu podanego podciągu spójnego w podanym ciągu składającym się z liter A, G, C i T.
6. W temacie przedstawione są trzy algorytmy znajdowania maksymalnej sumy podciągu spójnego. Uczniów można więc podzielić na trzy zespoły, z których każdy będzie rozwiązywał ćwiczenia dotyczące innego algorytmu. Po opracowaniu i przetestowaniu swojego algorytmu grupa prezentuje wyniki i wyjaśnia działanie algorytmu pozostałym zespołom.
7. W ramach rozszerzenia poruszanych zagadnień warto polecić uczniom rozwiązanie zadań na platformie SPOJ:
 - odnalezienie najdłuższego podciągu w ciągu binarnym:
pl.spoj.com/problems/AL_03_02,
 - odnalezienie najdłuższego podciągu w ciągu różnowartościowym:
pl.spoj.com/problems/FR_02_17,
 - określenie długości najdłuższego wspólnego podciągu dwóch ciągów:
pl.spoj.com/problems/LENLCS,
 - określenie sumy podciągów: pl.spoj.com/problems/FR_10_09,
 - określenie długości najdłuższego spójnego podciągu, który jest ciągiem arytmetycznym:
pl.spoj.com/problems/AL_20_02.

VII scenariusz zajęć Tworzenie stron internetowych HTML i CSS.

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 4

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- sprawdza poprawność działania algorytmów dla przykładowych danych,
- projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu,
- tworzy stronę internetową zgodnie ze standardami, wzbogaconą tabelami, listami, elementami dynamicznymi, posługuje się arkuszem stylów, korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron, potrafi opublikować własną stronę w internecie.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: znacznik, atrybut, selektor,
- wymienia elementy struktury strony internetowej w kodzie HTML.

Zrozumienie

Uczeń:

- omawia różnicę między ścieżką względną a bezwzględną,
- omawia zastosowanie podstawowych znaczników HTML,
- opisuje sposoby stosowania stylów CSS.

Zastosowanie

Uczeń:

- wykorzystuje znaczniki języka HTML do utworzenia prostej strony internetowej,

- używa podczas pracy nad stroną internetową edytorów kolorujących składnię języka HTML,
- stosuje na tworzonej przez siebie stronie style CSS.

Tworzenie

Uczeń:

- opracowuje projekt własnej strony internetowej.

Środki dydaktyczne:

- komputer z dostępem do internetu,
- strony internetowe: pl.lipsum.com, w3schools.com, unsplash.com,
- projektor multimedialny lub tablica multimedialna,
- program do edycji kodu HTML (Notepad++ kolorujący składnię lub systemowy Notatnik),
- tutoriale dostępne na stronie dlauczyciela.pl:
 1. *Struktura strony internetowej w języku HTML*
 2. *Dodawanie zdjęć do strony internetowej*
 3. *Zawartość arkusza stylów CSS*
- *Skrypt z teorią i zadaniami (*

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- analizy, opisu i algorytmu (MOA),
- analizy, uzupełniania i tworzenia kodu programu (MKOD).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji.

II. Faza zasadnicza	
Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. Skrypt str. 73 - 101	
Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

7. Temat można zrealizować w czasie czterech godzin lekcyjnych. Proponuje następujący podział:
 - Lekcja 1: zagadnienie 4.1, 4.2
 - Lekcja 2: zagadnienie 4.3
 - Lekcja 3: zagadnienie 5.1, 5.2
 - Lekcja 4: zagadnienie 5.3, 5.4

1. Na pierwszej lekcji warto przypomnieć wiadomości związane z zapisywaniem plików, rozszerzeniami plików i ich znaczeniem, ścieżkami względnymi i bezwzględnymi. Informacje te są niezwykle istotne i przydają się na każdym etapie tworzenia stron WWW, gdy pojawia się konieczność odwoływania się do plików umieszczonych w różnych lokalizacjach.
2. Warto zwrócić uwagę uczniów na edytory kodu HTML, kolorujące składnię języka i ułatwiające dzięki temu tworzenie strony internetowej.

3. Do nauki znaczników warto wykorzystać stronę w3schools.com, na której można zapisywać i testować kod źródłowy.
4. Podczas pracy w edytorach plików HTML warto podzielić ekran monitora na dwie części, tak aby widoczne były jednocześnie kod źródłowy i plik wyświetlony w przeglądarce. Umożliwi to edycję kodu HTML i bieżące sprawdzanie efektów zmian po odświeżeniu strony w przeglądarce.
5. Uczniom, którzy nie widzą potrzeby nauki tworzenia stron internetowych w języku HTML, warto zaproponować przeczytanie *Z informatyką w przyszłość* (s. 316), aby zdali sobie sprawę, że umiejętność tworzenia stron internetowych jest kompetencją pożądaną na rynku pracy. Warto również zwrócić uwagę na umieszczone w temacie ciekawostki.
6. Przykładowe pliki graficzne do ćwiczeń i zadań uczniowie mogą pobrać ze strony unsplash.com, teksty natomiast można wygenerować lub napisać, dobierając ich tematykę do zainteresowań uczniów.

VIII scenariusz zajęć

Projektowanie stron z wykorzystaniem WordPress

Szkoła: III Liceum Ogólnokształcącego im. św. Jana Kantego w Poznaniu.

Prowadzący: mgr Anna Jaborkhel

Klasa: 4c gr 1 i 2

Liczba jednostek lekcyjnych: 5

Cele ogólne:

Uczeń:

- planuje kolejne kroki rozwiązywania problemu z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania),
- tworzy stronę internetową zgodnie ze standardami, wzbogaconą tabelami, listami, elementami dynamicznymi, posługuje się arkuszem stylów, korzysta z oprogramowania i serwisów przeznaczonych do tworzenia stron, potrafi opublikować własną stronę w internecie,
- wyszukuje w sieci potrzebne informacje i zasoby, ocenia ich przydatność oraz wykorzystuje w rozwiązywanych problemach,
- objaśnia funkcje innych niż komputer urządzeń cyfrowych i korzysta z ich możliwości.

Cele szczegółowe

Zapamiętanie

Uczeń:

- definiuje pojęcia: hosting, responsywność, pozycjonowania stron
- wymienia elementy struktury strony internetowej w kodzie HTML,
- definiuje pojęcie CMS - system zarządzania treścią .

Zrozumienie

Uczeń:

- omawia różnicę między ścieżką względną a bezwzględną,
- omawia zastosowanie podstawowych znaczników HTML,
- opisuje sposoby stosowania stylów CSS.
- omawia różnicę między motywem a wtyczką w CMS.
- omawia uprawnienia kont użytkownika np. administrator, redaktor.

Zastosowanie

Uczeń:

- opracowuje projekt własnej strony internetowej.



- zarządza zawartością strony w WordPress,
- wykorzystuje system CMS na przykładzie WordPress
- potrafi dokonać modyfikacji motywów systemu,
- używa popularne wtyczki rozbudowujące funkcjonalności WordPress
- kreatywnie korzysta z mediów
- publikuje gotowy serwis w Internecie

Tworzenie

Uczeń:

- zakłada konta na stronie wordpress.com
- opracowuje projekt własnej strony internetowej.
- .potrafi założyć konto na stronie wordpress.com
- tworzy stronę internetową z wykorzystaniem CMS
- potrafi samodzielnie stworzyć bloga z użyciem systemu zarządzania treścią WordPress

Środki dydaktyczne:

- komputer z dostępem do internetu,
- konto na stronie wordpress.com
- strony internetowe: w3schools.com, dns.pl, wordpress.com,
- projektor multimedialny lub tablica multimedialna,
- tutoriale dostępne na stronie wordpress.com.
- *Skrypt z teorią i zadaniami.*

Metody pracy:

- wykładu konwersatoryjnego (MWK),
- ćwiczeń (MC),
- projektu edukacyjny (MPR),
- praca z wykorzystaniem programów komputerowych i sieci Internet (MEL).

Przebieg zajęć:

I. Faza wstępna

- 1 Przywitanie.
- 2 Sprawdzenie obecności uczniów na zajęciach.
- 3 Przypomnienie podstawowych informacji dotyczących wykorzystania funkcji.

II. Faza zasadnicza	
Zapoznanie uczniów z celem zajęć	<ul style="list-style-type: none"> • sprawne komunikowanie się w języku polskim z wykorzystaniem terminologii informatycznej • sprawne wykorzystywanie narzędzi TIK w życiu codziennym, a także kształcenia myślenia komputacyjnego • rozwiązywanie problemów z wykorzystaniem narzędzi informatycznych
Przekazanie uczniom najważniejszych informacji zawartych w tematyce zajęć, popartych licznymi przykładami. Skrypt str. 102 - 126	
Wspólne rozwiązywanie ćwiczeń z wykorzystaniem sprzętu komputerowego.	
Indywidualizacja pracy w trakcie trwania zajęć z wszystkimi uczniami.	

III Faza końcowa

- 1 Prośba o wykonanie nieprzerobionych zadań.
- 2 Podsumowanie i szybkie przeanalizowanie zajęć.
- 3 Podziękowanie za zajęcia

Wskazówki do prowadzenia zajęć:

1. Zagadnienie można zrealizować w czasie pięciu godzin lekcyjnych. Proponuje następujący podział:
 - Lekcja 1: zagadnienie 6.1.1
 - Lekcja 2: zagadnienie 6.1.2 i 6.1.3
 - Lekcja 3, 4, 5: prace nad projektem strony **skrypt str. 127**
2. Na pierwszej lekcji warto przypomnieć wiadomości związane z zapisywaniem plików, rozszerzeniami plików i ich znaczeniem, ścieżkami względnymi i bezwzględnymi. Informacje te są niezwykle istotne i przydają się na każdym etapie tworzenia stron WWW, gdy pojawia się konieczność odwoływania się do plików umieszczonych w różnych lokalizacjach.

3. Założenie konta na stronie wordpress.com i zapoznanie się z systemem proponuję zrealizować na pierwszych dwóch zajęciach.
4. Na trzecich zajęciach powinno nastąpić omówienie projektu, podział na grupy, rozdzielenie zadań oraz gromadzenie przez grupy materiałów.
5. Jedna osoba w zespole powinna być administratorem systemu, czyli podać swój adres e-mail w serwisie do obsługi systemu CMS i ustalić hasło. Warto tutaj przypomnieć o zasadach tworzenia bezpiecznego hasła i konieczności jego zapamiętania. Administrator z założenia nie dzieli się dostępem do swojego konta, ale tworzy oddzielne konta dla poszczególnych użytkowników. Dobrze, jeśli każdy zespół wyznaczy na administratora osobę, która sprawdzi się również jako kierownik projektu.
6. Ważnym etapem pracy nad projektem jest sprawdzenie działania strony na różnych urządzeniach. Warto przypominać uczniom o konieczności zadbania o responsywność przygotowywanej przez nich strony internetowej.
7. Treści do umieszczenia na projektowanej stronie uczniowie mogą przygotować w domu. Dzięki temu czas lekcji zostanie w całości przeznaczony na kwestie techniczne.
8. Warto pozwolić uczniom na zgłaszanie własnych propozycji tematów. Mogą np. przygotować stronę internetową na temat związany z zagadnieniem omawianym aktualnie na innym przedmiocie.
9. Uczniowie mogą wykorzystać w budowanej stronie odpowiednie wtyczki, które poszerzą funkcje systemu CMS.
10. Na każdych zajęciach trwa weryfikacja postępów uczniów, dyskusje w grupach, wskazówki nauczyciela.
11. Piąte zajęcia przeznaczam na prezentacja projektów przez poszczególne zespoły oraz ustną ocenę projektów.