

Część X C++

Typ znakowy służy do reprezentacji pojedynczych znaków ASCII, czyli liter, cyfr, znaków przestankowych i innych specjalnych znaków widocznych na naszej klawiaturze (oraz wielu innych, których nie znajdziemy na klawiszach).

Aby stworzyć zmienną typu znakowego, używamy słowa kluczowego **char**.

Ćwiczenie 1

1. Utwórz nowy projekt w Dev C++ i zapisz go w folderze **nazwisko33**
2. Do zmiennej o nazwie znaki przypisz małą literę k - pamiętając, aby ująć ją w znaki apostrofu -wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
3. Skompiluj i uruchom program
4. Przeanalizuj program

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    char znak1 = 'k';
    char znak2 = 107;

    cout << znak1 << endl;
    cout << znak2 << endl << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

```
k
k
Aby kontynuować, naciśnij dowolny klawisz . . .
```

Po uruchomieniu programu nasza litera zostanie wyświetlona na ekranie

Do zmiennej znakowej można przypisać również liczbę

Czemu typ znakowy zaliczamy do typów arytmetycznych, czyli liczbowych, ponieważ tak naprawdę to typ znakowy jest typem liczbowym.

Zmienna typu znakowego może bowiem przechować liczbę od 0 do 255, a każdemu znakowi w kodzie ASCII odpowiada konkretna liczba (na przykład małej literze k odpowiada liczba 107).

Ćwiczenie 2

1. Uruchom plik o nazwie **nazwisko33 z rozszerzeniem DEV** (znajduje się w folderze

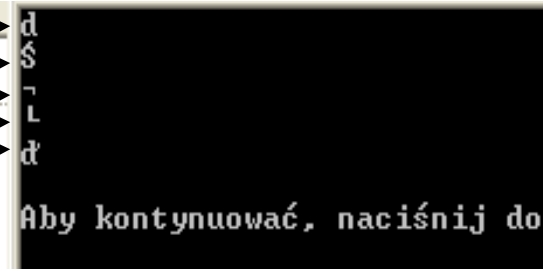
- nazwisko 33)
2. Wprowadź modyfikacje zapisując do zmiennej znakowej dowolną liczbę z przedziału od 0 do 255. Wprowadź pięć takich zmiennych.(możesz skorzystać z poniższego zrzutu ekranu)
 3. Skompiluj i uruchom program
 4. Przeanalizuj program

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    char znak1 = 100;
    char znak2 = 151;
    char znak3 = 170;
    char znak4 = 192;
    char znak5 = 212;
    cout << znak1 << endl;
    cout << znak2 << endl;
    cout << znak3 << endl;
    cout << znak4 << endl;
    cout << znak5 << endl << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```



Zobacz że każdej liczbie odpowiada litera bądź znak

Prawda czy fałsz

Najprostszym, ale bardzo często używanym typem danych jest tryb logiczny. Do jego określenia służy **słowo kluczowe bool**. Do zmiennej typu logicznego możemy zapisać wyłącznie 0 lub 1. Zmienną tego typu najczęściej stosuje się do określenia, czy coś jest włączone, czy wyłączone lub prawdziwe albo fałszywe.

Jeśli relacja jest prawdziwa np $7 > 3$, to rezultatem jest 1. Gdy relacja jest nieprawdziwa, otrzymujemy wartość 0.

Aby zobaczyć, w jaki sposób zdefiniować zmienną logiczną i jak z niej korzystać, wykonujemy ćwiczenie.

Ćwiczenie 3

1. Utwórz nowy projekt w Dev C++ i zapisz go w folderze **nazwisko34**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
3. Skompiluj i uruchom program

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    bool wlaczony;

    wlaczony = 1;
    cout << wlaczony << endl;

    wlaczony = false;
    cout << wlaczony << endl;

    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Aby wyróżnić wartości zapisywane do zmiennej typu logicznego od zmiennej typu arytmetycznego, zamiast 1 i 0 stosuje się określenia **true** i **false**. Pierwsze z nich oznacza prawdę i przypisuje zmiennej wartość 1, drugie - fałsz i przypisuje zmiennej wartość 0.

Sterowanie programem

Instrukcje znajdujące się w funkcji głównej wykonywane są jedna po drugiej. W ten sposób nie da się jednak tworzyć dużych i praktycznych aplikacji. Dlatego warto poznać polecenia, za pomocą których możemy sterować wykonywaniem kodu źródłowego.

Na początku poznamy **instrukcje warunkowe**. Są to takie twory języka, które pozwalają komputerowi wykonywać pewien blok instrukcji tylko wtedy, gdy zostaje spełniony określony przez nas warunek.

Oprócz instrukcji warunkowych poznamy również **pętle**. Dzięki nim każda instrukcja może być wykonana wielokrotnie (oczywiście mamy duży wpływ na liczbę powtórzeń).

Szybko się przekonamy, że instrukcje warunkowe i pętle znacznie ułatwiają programowanie oraz oczywiście pozwalają na pisanie lepszych, bardziej rozbudowanych i praktycznych programów.

Instrukcja warunkowa if (z ang. jeśli).

instrukcję warunkową if możemy ogólnie zapisać w następującej postaci:

if (warunek) instrukcja;

Działanie instrukcji if można rozumieć następująco: Jeśli warunek jest prawdziwy, wykonaj instrukcję. Gdy warunek jest fałszywy, nie rób nic.

Działanie operatorów relacji polega na wyświetlaniu liczby 0 w wypadku, gdy relacja była fałszywa, oraz 1, gdy była prawdziwa. Zobaczmy teraz, w jaki sposób od wyniku tej relacji uzależnić wykonanie fragmentu kodu.

Ćwiczenie 4

Napisać program, który sprawdza relację dwóch liczb oraz w zależności od tego, która jest większa, wyświetla na ekranie odpowiedni komunikat.

1. Utwórz nowy projekt w Dev C++ i zapisz go w folderze **nazwisko35**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
3. Skompiluj i uruchom program
4. Przeanalizuj program

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    float a, b;
    cout << "Podaj liczbę A : ";
    cin >> a;
    cout << "Podaj liczbę B : ";
    cin >> b;

    if ( a < b ) cout << "Liczba A jest mniejsza od B." << endl;

    if ( a > b ) cout << "Liczba A jest większa od B." << endl;

    if (a == b) cout << "Liczby A i B są równe." << endl;

    cout << endl << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Warunkowe wykonanie kilku poleceń

W poprzednim programie przy spełnieniu warunku zostaje wykonana jedna instrukcja cout. Jeśli chcemy aby w przypadku spełnienia warunku było wykonanych kilka poleceń wystarczy ująć je w nawiasy klamrowe.

Ćwiczenie 5

1. Utwórz nowy projekt o nazwie **nazwisko36**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej
3. Skompiluj i uruchom program
4. Przeanalizuj program

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    float a, b;
    cout << "Podaj liczbe A : ";
    cin >> a;
    cout << "Podaj liczbe B : ";
    cin >> b;

    if ( a < b ) {
        cout << "Liczba A jest mniejsza od B." << endl;
        cout << "Roznica pomiedzy zmiennymi wynosi " << b - a << endl;
    }

    if ( a >= b ) {
        cout << "Liczba A jest wieksza lub rowna B." << endl;
        cout << "Roznica pomiedzy zmiennymi wynosi " << a - b << endl;
    }

    cout << endl << endl;

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Wszystkie pliki z nazwiskiem i kolejnym numerem umieszczamy w swoim folderze nazwiskoplusplus na serwerze.