

Część XIX C++ w

Każda poznana do tej pory zmienna może przechowywać jedną liczbę. Jeśli zacniemy pisać bardziej rozbudowane programy, okaże się to niewystarczające. Warto więc poznać zmienne, które mogą przechowywać nawet kilkadziesiąt wartości.

Na początku dowiemy się, czym są tablice. Odkryjemy również, że za pomocą tablicy (oczywiście odpowiedniego typu) możemy stworzyć łańcuch znakowy. W dalszej części poznamy bardzo ciekawy typ danych - wskaźnikowy. Taka zmienna może przechowywać adres komórki (w pamięci naszego komputera), w której znajduje się wartość na przykład innej zmiennej. Choć zagadnienie to wydaje się skomplikowane, warto stawić mu czoło.

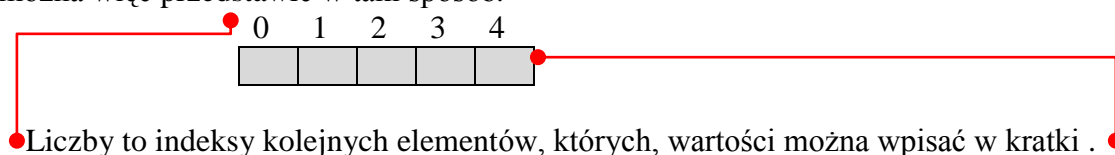
Na koniec omówione typy danych wykorzystamy w praktyce - tablice i wskaźniki prześlemy jako argumenty do stworzonej przez nas funkcji. Odkryjemy w ten sposób wiele ciekawych zależności.

Niezwykłe tablice

Poznane typy danych pozwalają przechowywać pojedyncze liczby. Dzięki tablicom zgromadzimy wiele wartości w jednym miejscu.

W każdym programie (no, może poza najprostszymi kilkuliniowymi kodami) dużą rolę odgrywają zmienne. Na przykład, gdy chcieliśmy obliczyć pole prostokąta, prosiliśmy użytkownika o podanie dwóch liczb określających długości boków figury. Liczby te następnie zapisaliśmy do dwóch zmiennych o różnych nazwach (na przykład **a** i **b**). Zastanówmy się, ile zmiennych potrzebowalibyśmy w programie obliczającym powierzchnię na przykład mieszkania? Z pewnością zbyt wielu. W jaki więc sposób zapisać dużo wartości w kodzie źródłowym? Najlepszym sposobem jest wykorzystanie do tego celu zmiennej tablicowej (określanej najczęściej jako tablica lub macierz). Poznajmy więc zasadę tworzenia tablic i korzystania z nich.

Tablica jest zbiorem elementów jednego typu. Wszystkie te elementy umieszczane są w pamięci naszego komputera razem w jednym miejscu i do każdego z tych elementów możemy się odwołać, posługując się nazwą tablicy i indeksem określającym numer danego elementu. Najprostszą tablicę można więc przedstawić w taki sposób.



Elementy tablicy numerowane są od 0.

Tak więc w zadeklarowanej przez nas tablicy pięcioelementowej kolejne elementy są ponumerowane od 0 do 4. Indeks 0 odnosi się do pierwszego elementu, indeks 1 do drugiego i tak dalej, i aż do indeksu 4, który reprezentuje piąty (ostatni) element tablicy **oceny**

Tablica jednowymiarowa

Najprostszym przykładem tablicy jest jej jednowymiarowa postać. Spójrzmy na przykład wykorzystujący tego typu tablicę.

Tablicę deklaruje się podobnie jak każdą inną zmienną z tą różnicą, że w wypadku tablicy po jej nazwie w nawiasach kwadratowych wpisujemy rozmiar tablicy. Nawiasy kwadratowe mówią kompilatorowi, że ma do czynienia właśnie z tablicą

Ćwiczenie 1

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko66**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int oceny[5];

    oceny[0] = 4;
    oceny[1] = 5;
    oceny[2] = 4;
    oceny[3] = 3;
    oceny[4] = 2;

    for (int i=0; i<=4; i++)
    {
        cout << "Przedmiot nr " << i;
        cout << " - ocena: " << oceny[i];
        cout << endl;
    }

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

```

Wewnątrz funkcji **main** wpisujemy instrukcję. Należy ją rozumieć w następujący sposób: Stwórz zmienną tablicową o nazwie **oceny**, która składa się z pięciu elementów typu całkowitoliczbowego (**int**).

Wpiszemy teraz do tablicy kolejne wartości. Aby to zrobić, wystarczy podać nazwę tablicy, w nawiasach kwadratowych umieścić numer elementu, który chcemy wpisać, i za pomocą operatora przypisania nadać mu określoną wartość (takiego typu, jak ustaliliśmy w deklaracji zmiennej tablicowej).

Tablica **oceny** powinna zostać wypełniona wartościami. Wyświetlimy je na ekranie, aby sprawdzić, czy rzeczywiście tak się stało. Do wyświetlenia najlepiej użyć pętli for. Tworzymy więc pętlę 0 (w której zmienna iteracyjna przybiera wartości od 0 do 4 - zgodnie z indeksami naszej tablicy, więc możemy ją wykorzystać do indeksowania tablicy **oceny**).

3. Z menu **Uruchom** wybieramy **Kompiluj i uruchom**.

Inicjacja tablicy

Wartości tablicy możemy wpisać również podczas jej deklaracji. W tym celu wartości kolejnych elementów tablicy oddzielone od siebie przecinkami umieszczamy w nawiasach klamrowych - na przykład

```
int oceny[5] = {4,5,4,3,2};
```

Ćwiczenie nr 2

Pętle (najczęściej typu for) wykorzystuje się często zarówno do wpisywania do tablicy wartości, jak też do wyprowadzania ich na ekran.

Spójrzmy na przykład poniżej w którym do tablicy **potęgi dwójki** wpisujemy 10 kolejnych potęg liczby 2. Wykorzystując drugą pętlę, wyświetlamy wartości tablicy na ekranie

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko67**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int potegi_dwojki[10];
    for (int j=0; j<10; j++)
    {
        int wynik = 1;
        for (int i=0; i<j; i++) wynik*=2;
        potegi_dwojki[j] = wynik;
    }

    for(int i = 0; i < 10; i++)
    {
        cout << "2 do potegi " << i << " wynosi ";
        cout << potegi_dwojki[i] << endl;
    }

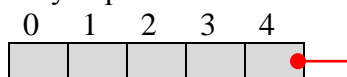
    cout << endl << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

3. Z menu **Uruchom** wybieramy **Kompiluj i uruchom**.

Tablice wielowymiarowe

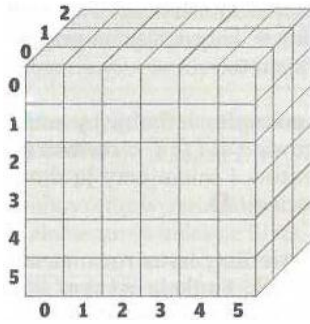
Poznane przez nas do tej pory tablice były jednowymiarowe - można je było przedstawić w postaci ponumerowanych pól.



W języku C++ (a także w prawie każdym języku programowania) można tworzyć tablice znacznie bardziej złożone - jak na przykład tablica dwuwymiarowa, którą możemy graficznie zobrazować w taki sposób- czyli jako tabelę składającą się z określonej liczby kolumn i wierszy.

	0	1	2	3	4
0					
1					
2					
3					
4					

Bez problemów stworzymy tablicę trójwymiarową oraz tablicę o większej liczbie wymiarów. Jednak ze względu na to, że tablice o liczbie wymiarów większych od 3 trudno sobie nawet wyobrazić, rzadko się je stosuje.



Ćwiczenie nr 3

Zobaczmy jak zadeklarować tablicę dwuwymiarową i jak z niej korzystać

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko68**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

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

using namespace std;

int main(int argc, char *argv[])
{
    int tablica[3][4];
    int liczba=1;

    for (int i=0; i<3; i++)
    {
        for (int j=0; j<4; j++)
        {
            tablica[i][j]=liczba;
            liczba++;
        }
    }

    for (int i=0; i<3; i++)
    {
        for (int j=0; j<4; j++)
        {
            cout << "tablica[" << i << "][" << j << " ]";
            cout << "=" << tablica[i][j] << endl;
            liczba++;
        }
    }

    cout << endl;

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

• Tablicę dwuwymiarową deklarujemy podobnie jak jednowymiarową. Jediną różnicą jest dodatkowa para nawiasów kwadratowych.

Rozmiar tablicy dwuwymiarowej określamy jako liczbę wierszy i kolumn (podając te liczby w nawiasach kwadratowych). Nasza tablica ma mieć wymiar 3 na 4.

Do tablicy w podwójnej pętli wpisujemy kolejne cyfry. Dzięki temu przekonamy się, w jaki sposób są one zapisywane w tablicy. Tworzymy więc pętlę, która będzie przechodziła przez kolejne wiersze tablicy, oraz pętlę przechodzącą przez kolejne kolumny tablicy w ramach danego wiersza. Do tablicy wpisujemy wartość zmiennej **liczba**, która jest w każdym przebiegu pętli zwiększona o 1.

3. Z menu **Uruchom** wybieramy **Kompiluj i uruchom**.

Tablice bez wymiarów

Aby zadeklarować tablicę, musimy podać typ elementów, które będą się w niej znajdować, nazwę tablicy oraz jej rozmiar. Gdy nie podamy żadnego typu, kompilator jako domyślny typ ustawi **int**. A co się stanie, jeśli nie podamy rozmiaru tablicy? Sprawdźmy to, wpisując w kodzie linijkę.

```
int tablica[]
```

Próba kompilacji takiego kodu kończy się niepowodzeniem - pojawia się błąd treści: „storage size of 'tablica' isn't known" (ang. rozmiar tablicy nie jest znany).

Spróbujmy jednak od razu przy deklaracji zainicjować tablicę, czyli przypisać wartości elementom tablicy.

```
int tablica[]={6,5,4,3,2,1,0};  
cout << tablica [4];
```

Taki program kompiluje się bez problemu, a próba wyświetlenia dowolnego elementu tablicy kończy się sukcesem. Dlaczego się tak dzieje? Kompilator na podstawie danych inicjujących tablicę sam ustala odpowiednią wielkość tablicy. Jak więc widzimy, nie musimy podawać rozmiaru tablicy jeśli podczas deklaracji inicjujemy ją.

Ćwiczenie 4

1. Utwórz nowy projekt w Dev C++ i zapisz go na pulpicie w folderze o nazwie **nazwisko69**
2. Wprowadź do projektu modyfikacje tak aby wyglądał jak poniżej

```
#include <cstdlib>  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char *argv[])  
{  
  
    int tablica[]={6,5,4,3,2,1,0};  
    cout << tablica [4];  
  
    cout << endl << endl;  
    system("PAUSE");  
    return EXIT_SUCCESS;  
}
```

3. Z menu **Uruchom** wybieramy **Kompiluj i uruchom**

Podobnie, jak w wypadku tablicy jednowymiarowej, przy okazji deklaracji tablicy dwuwymiarowej możemy od razu wpisać do niej wartości. Są dwa sposoby zainicjowania tablicy dwuwymiarowej

- ▶ możemy w nawiasach klamrowych podać kolejne wartości, oddzielając je od siebie przecinkami (liczba elementów w tablicy równa jest iloczynowi zadeklarowanych przez nas wierszy i kolumn),

```
tablica[3][4]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

- ▶ możemy podać wartości, obejmując kolejne wiersze dodatkowymi nawiasami klamrowymi - taki zapis jest bardziej czytelny

```
tablica[3][4]= {{1, 2, 3, 4,} {5, 6, 7, 8,} {9, 10, 11, 12}};
```

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