

Materiały powtórzeniowe

Lekcja 11. Tablice

Witaj w materiałach powtórzeniowych przygotowanych specjalnie dla Ciebie. Znajdziesz tutaj powtórkę materiału z ostatniej lekcji oraz zadania do wykonania dla chętnych. Z pewnością sobie poradzisz. Powodzenia!



Cel lekcji

Celem lekcji jest omówienie zastosowania i używania tablic.

Tablice można sobie wyobrazić jak taki worek, do którego możemy wrzucać wiele elementów. Zamiast tworzyć dużą ilość zmiennych, tworzymy jedną tablicę do której dodajemy elementy.

Tablica to struktura danych pozwalająca na przechowywanie zbioru elementów określonego typu. Najprostszą tablicę można zilustrować w poniższy sposób (tablica przechowująca zmienne liczbowe typu int):

Tablica o nazwie **liczby**

Numer elementu(indeksu)	0	1	2	3
Wartość	24	25	87	93

W powyższym przykładzie tablica ma nazwę **liczby** i przechowuje 4 wartości liczbowe. Tablica pozwala na przechowywanie określonej ilości wartości i co wynika z przedstawionej definicji pojęcia tablica – wszystkie te zmienne muszą być tego samego typu. **Pamiętajmy, iż elementy w tablicy są numerowane od 0! W powyższej tablicy element pierwszy (indeks 0) ma wartość 24, na drugim miejscu (indeks 1) jest liczba 25 itd.**

Aby móc używać w programie tablice należy zadeklarować zmienną tablicową oraz utworzyć samą tablicę (obiekt tablicy). W przypadku tablic istnieje kilka możliwości ich deklarowania.



Składnia:

Typ[] nazwa = new Typ[rozmiar]; // deklaracja jednolinijkowa

LUB

Typ[] nazwa; // deklaracja zmiennej tablicowej

nazwa = new Typ[rozmiar]; // utworzenie tablicy (obiektu) i przypisanie referencji

Przykład:

```
static void Main(string[] args)
{
    int [] tablica_int = new int[3]; //tablica ma 3 elementy typu int
    tablica_int[0] = 14;
    tablica_int[1] = 16;
    tablica_int[2] = 18;
}
```

Aby wypisać teraz elementy tej tablicy na ekranie, można posłużyć się pętlą for:

```
for(int i=0; i<3;i++){
    Console.WriteLine("Liczba pod indeksem {0} w tablicy ma wartość {1}",i,tablica_int[i]);
    //do wartości odpowiednich elementów w tablicy odwołujemy się poprzez nazwa_tablicy[indeks]
}
```

Efekt działania kodu:



```
Liczba pod indeksem 0 w tablicy ma wartość 14  
Liczba pod indeksem 1 w tablicy ma wartość 16  
Liczba pod indeksem 2 w tablicy ma wartość 18
```

Możemy zadeklarować tablicę także od razu inicjalizując jej elementy, według składni:

```
Typ[] nazwa = new Typ[]{Lista inicjalizacyjna};
```

lub

skrótowa wersja:

```
Typ[] nazwa = {Lista inicjalizacyjna};
```

Praktyczny przykład deklaracji tablicy:

```
int[] uczestnicy = new int[] { 15, 39, 13, 58, 41 };
```

Po nazwie typu określającego typ wartości, jakie będą przechowywane w tablicy (np. *byte*, *int*, *float*, *string*) pojawiają się puste nawiasy kwadratowe. Nawiasy te informują, że deklarowana zmienna będzie zmienną tablicową. Następnie pojawia się nazwa zmiennej. Tablica ta jest od razu wypełniana wartościami, dlatego też (analogicznie jak to było w przypadku „zwykłych” zmiennych).

Nawiasy kwadratowe występujące po słowach *new int* mogą zawierać informację o liczbie elementów (rozmiar tablicy), jakie będą przechowywane w tablicy. Jednak liczba ta musi być zgodna z liczbą elementów, które zostały wypisane w nawiasach klamrowych. Dlatego też, eliminując potencjalne miejsce popełnienia błędu, najlepiej jest nie wpisywać rozmiaru w tej deklaracji.

W skróconej wersji bez operatora *new* można zainicjalizować tablicę w następujący sposób:

```
int[] uczestnicy = { 15, 39, 13, 58, 41 };
```



Pętla foreach

Czasem potrzebujemy wykonać jakieś działanie dla każdego elementu w tablicy.

Jak mogliście zauważyć, wykorzystanie w tym celu pętli for jest mało wygodnym sposobem. W związku z tym do języka C# dodano dodatkowy rodzaj pętli, służący właśnie do działań na tablicach (i innych kolekcjach danych, które poznacie w przyszłym semestrze).

Składnia:

foreach([typ zmiennej] nazwa in nazwaKolekcji)

Do zrozumienia składni posłużmy się przykładem. Załóżmy, że byliśmy na zakupach i wracamy do domu z torbą pełną zakupów (mleko, masło, ser, pomidor itd.). Naszym zadaniem jest teraz wyciągnięcie każdego produktu z torby i schowanie go do lodówki. Algorytm naszej pętli wygląda następująco:

Weź każdy produkt z torby i schowaj go do lodówki.

Przenieśmy to do kodu:

```
static void Main(string[] args)
{
    // tworzymy tablicę typu string z kilkoma nazwami produktów
    string[] torbaNaZakupy = { "mleko", "masło", "ser", "pomidor" };

    // deklarujemy pętle
    // typ elementu musi być zgodny z typem zmiennych w naszej tablicy
    foreach (string produkt in torbaNaZakupy)
    {
        Console.WriteLine($"wkładam {produkt} do lodówki");
    }
}
```



Dodatkowa misja Programisty:

Zadanie 1

Stworzyć program konsolowy, w którym stworzysz tablicę 4 elementową, wypełnisz ją nazwiskami i wyświetlisz w konsoli.

Zadanie 2

Napisać program w konsoli, który znajdzie najmniejszą wartość w tablicy.

