

■ Tablice

Tablice służą do przechowywania większej liczby obiektów tego samego typu. Tablice mogą być jednowymiarowe (często nazywane wektorami) lub wielowymiarowe (dwa lub więcej wymiarowe). W C# numerowanie elementów tablicy zaczyna się od 0. Aby odwołać się do tablicy jednowymiarowej należy podać nazwę tablicy i numer elementu, np. `tab1D[0]`. Aby odwołać się do tablicy dwuwymiarowej należy podać nazwę tablicy oraz numer wiersza i kolumny, np. `tab2D[2, 3]`.

```
int[] a;           // deklaracja tablicy jednowymiarowej typu int
a = new int[5];    // utworzenie tablicy jednowymiarowej o pięciu
                  // elementach

double[,] b;       // deklaracja tablicy dwuwymiarowej typu double
b = new double[3, 5]; // utworzenie tablicy dwuwymiarowej o rozmiarze 3x5

int[] c = new int[3]; // deklaracja i utworzenie tablicy jednowymiarowej
                    // 3 elementowej

int[] d = new int[2] { 4, 6 }; // deklaracja, utworzenie i
                              // zainicjowanie tablicy liczbami 4 i 6

int[] e = { 4, 6 }; // deklaracja, utworzenie i zainicjowanie tablicy
                  // liczbami 4 i 6 - uproszczona wersja
```

Przykład 5.1 Utworzyć pięcioelementową tablicę do przechowywania danych typu `int`. Do tablicy wczytać 5 liczb całkowitych i obliczyć sumę wszystkich elementów tablicy.

```
void Test51()
{
    int[] tab;
    tab = new int[5];

    // wczytywanie danych do tablicy
    for (int i = 0; i < 5; i++)
    {
        Console.Write("Podaj a{0}: ", i);
        tab[i] = Int32.Parse(Console.ReadLine());
    }
    // operacja na tablicy - sumowanie elementów
    int sum = 0;
    for (int i = 0; i < 5; i++)
    {
        sum += tab[i];
    }

    // wypisanie wyniku
    Console.WriteLine("Suma elementów tablicy wynosi: {0}", sum);
}
```

Przykład 5.2 Utworzyć dwuwymiarową tablicę o 3 wierszach i 3 kolumnach do przechowywania liczb typu `int`. Do tablicy wczytać 9 liczb całkowitych i obliczyć sumę elementów tablicy.

```
void Test52()
{
    int[,] tab = new int[3, 3];

    // wczytywanie danych do tablicy
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            Console.Write("Podaj a{0}{1}: ", i, j);
            tab[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    // operacja na tablicy - sumowanie elementów
    int sum = 0;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            sum += tab[i, j];
        }
    }

    // wypisanie wyniku
    Console.WriteLine("Suma elementów tablicy wynosi: {0}", sum);
}
```

Przykład 5.3 Zastosowanie pętli `foreach` do iteracji po elementach tablicy

```
void Test53()
{
    int[] tab = new int[5];

    // wczytywanie danych do tablicy
    for (int i = 0; i < 5; i++)
    {
        Console.Write("Podaj a{0}: ", i);
        tab[i] = Int32.Parse(Console.ReadLine());
    }

    // operacja na tablicy - sumowanie elementów
    int sum = 0;
    foreach (int a in tab)
    {
        sum += a;
    }

    // wypisanie wyniku
    Console.WriteLine("Suma elementów tablicy wynosi: {0}", sum);
}
```

Zadanie 5.1 Napisać program, który wczytuje 5 liczb do tablicy jednowymiarowej, a następnie znajduje największy element tablicy i jego pozycję (indeks) w tablicy.

Zadanie 5.2 Napisać program do wykonywania następujących operacji na wektorach

- Mnożenie wektora przez liczbę
- Mnożenie wektora przez wektor

Zadanie 5.3 Napisać program do wykonywania następujących operacji na macierzach

- Mnożenie macierzy przez liczbę
- Mnożenie macierzy przez wektor
- Mnożenie macierzy przez macierz

Zadanie 5.4 Napisać program, który wypełnia tablicę 5x5 w następujący sposób

1	0	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

Uwaga: Do wypełniania tablicy należy wykorzystać pętlę `for`.

Zadanie 5.5 Napisać program do sortowania jednowymiarowej tablicy metodą bąbelkową.

Wskazówka: Porównujemy dwa sąsiednie elementy tablicy. Jeśli są w dobrej kolejności, przechodzimy do porównania kolejnych dwóch elementów, jeśli w złej – zamieniamy miejscami i przechodzimy do porównywania kolejnych dwóch elementów. Aby uzyskać posortowaną tablicę, powyższą operację należy powtórzyć $n-1$ razy, gdzie n – liczba elementów tablicy.

Zadanie 5.6 Napisać program do sortowania jednowymiarowej tablicy korzystając z następującego algorytmu:

Znajdujemy największy element w całej tablicy i zamieniamy miejscami z pierwszym elementem, następnie znajdujemy największy element wśród pozostałych (tj. oprócz pierwszego) i zmieniamy miejscami z elementem na pozycji 2. Powyższą operację powtarzamy $n-1$ razy, gdzie n – liczba elementów tablicy.