

Podstawy programowania

Laboratorium

Ćwiczenie 3 – Tablice

Tablica - zbiór zmiennych tego samego typu (np. całkowitego, rzeczywistego, tekstowego), do których można odwołać się poprzez jedną wspólną nazwę. W deklaracji tablicy należy określić typ wartości, jaki ma przechowywać tablica, a także liczbę elementów tablicy. W języku C# tablice mogą być jednowymiarowe, dwuwymiarowe, trójwymiarowe itd.

Tablica jednowymiarowa

Przykład ogólny deklaracji tablicy jednowymiarowej:

```
typ_tablicy [] nazwa = new typ_tablicy [rozmiar_tablicy];
```

gdzie: `typ_tablicy` określa podstawowy typ tablicy, tzn. definiuje typ danej każdej komórki w tej tablicy, `rozmiar_tablicy` - liczba elementów, które zawiera tablica.

Przykłady deklaracji tablic jednowymiarowych:

```
Int32 [] tab = new Int32 [20];  
String [] tab = new String [10];  
Double [] tab = new Double [30];
```

Dostęp do konkretnej wartości w tablicy jest realizowany za pośrednictwem indeksu, który wskazuje dany element. Dla deklaracji tablicy:

```
Int32 [] tab = new Int32 [10];
```

aby uzyskać dostęp do pierwszego elementu tablicy `tab`, należy podać indeks 0, drugi element dostępny jest przez indeks 1 itd. Ostatni element tablicy ma indeks równy rozmiarowi tablicy $10 - 1$, czy 9, co przedstawiono poniżej w reprezentacji granicznej tablicy.

0	1	2	3	4	5	6	7	8	9

Polecenie 1:

Napisz program, który w 20-elementowej tablicy jednowymiarowej o nazwie **tab1** umieszcza liczby losowe pochodzące z przedziału od 0 do 9. Program powinien wyświetlać postać tablicy na ekranie, tj. numery indeksów wraz z przyporządkowanymi wartościami w poszczególnych komórkach tablicy. W realizacji zadania zastosuj dowolną pętlę. Zgłoś Prowadzącemu zajęcia zakończenie realizacji tego polecenia.

Polecenie 2:

Napisz program, który w 10-elementowej tablicy jednowymiarowej o nazwie **tab1** umieszcza liczby od 9 do 0 zgodnie z poniższą reprezentacją graficzną tablicy. Program powinien wyświetlać postać tablicy na ekranie, tj. numery indeksów wraz z przyporządkowanymi wartościami w poszczególnych komórkach tablicy. W realizacji zadania zastosuj dowolną pętlę. Zgłoś Prowadzącemu zajęcia zakończenie realizacji tego polecenia.

9	8	7	6	5	4	3	2	1	0
0	1	2	3	4	5	6	7	8	9

Tablica dwuwymiarowa

Przykład ogólny deklaracji tablicy dwuwymiarowej:

```
typ_tablicy [,] nazwa = new typ_tablicy [rozmiar1,rozmiar2];
```

gdzie: `typ_tablicy` określa podstawowy typ tablicy, tzn. definiuje typ danej każdej komórki w tej tablicy, `rozmiar1`, `rozmiar2` - numery indeksów wskazujące współrzędne danego elementu w tablicy. W pewnym sensie numery indeksów można traktować jako numer wiersza oraz numer kolumny w tablicy, przy czym kolejność definiowania (kolumna, wiersz) albo (wiersz, kolumna) jest kwestią umowną zależną od programisty.

Układ dwóch (lub wielu) pętli

W przypadku tablicy dwuwymiarowej, w celu uzyskania dostępu do wszystkich jej komórek niezbędne jest przemiatanie poszczególnych wartości indeksów `rozmiar1` dla poszczególnych wartości indeksów `rozmiar2`. Przydatnym narzędziem do realizacji tego celu jest układ dwóch pętli.

Przykład zapisu układu dwóch pętli pokazano poniżej:

```
for (a = 0; a <= 3 ; a++)
{
    //tutaj znajduje się zestaw instrukcji wykonywany w pętli A
    for (b = 0; b <= 2; b++)
    {
        //tutaj znajduje się zestaw instrukcji wykonywany w pętli B
    };
};
```

Polecenie 3:

Napisz program, który wykonuje układ dwóch pętli z przykładu pokazanego powyżej. Uruchom program w trybie krokowym. Przeanalizuj kolejno zmieniające się wartości zmiennych sterujących w obu pętlach. Ile razy wykona się zestaw instrukcji wykonywany w ramach pętli A oraz pętli B?

Polecenie 4:

Napisz program, który w tablicy o wymiarach 4 wiersze x 6 kolumn o nazwie **tab1** umieszcza liczby losowe pochodzące z przedziału od 100 do 999. Program powinien wyświetlać postać tablicy na ekranie z zachowaniem odpowiedniego układu graficznego tej tablicy. W realizacji zadania zastosuj układ dwóch pętli. Wskazówka - W ramach wyświetlenia poszczególnych elementów pojedynczego wiersza tablicy można użyć polecenia

```
Console.Write(tab1[a,b]+" ");
```

które wyświetla wartość z komórki o indeksach `a,b` oraz dodatkową spację. Natomiast wyświetlenie wszystkich elementów wiersza można zakończyć poleceniem

```
Console.WriteLine();
```

które przeniesie kursor do nowej linii, co umożliwi wyświetlanie poszczególnych elementów kolejnego wiersza. Zgłoś Prowadzącemu zajęcia zakończenie realizacji tego polecenia.

Polecenie 5:

Uzupełnij program z **polecenia 4** o następujące elementy funkcjonalne:

- obliczanie sumy wszystkich elementów tablicy i wyświetlanie tej wartości na ekranie
- odszukiwanie wartości maksymalnej w tablicy i wyświetlanie tej wartości na ekranie
- odszukiwanie wartości maksymalnej w tablicy i wyświetlanie współrzędnych (numerów indeksów) tej wartości na ekranie
- odszukiwanie wartości minimalnej w tablicy i wyświetlanie tej wartości na ekranie
- odszukiwanie wartości minimalnej w tablicy i wyświetlanie współrzędnych (numerów indeksów) tej wartości na ekranie
- wykonanie transpozycji tablicy **tab1** i wyświetlenie na ekranie postaci tablicy transponowanej **tab2**. Transpozycja oznacza, że każdy kolejny wiersz starej tablicy stanowi nową kolumnę nowej tablicy, natomiast każda kolumna starej tablicy – staje się wierszem nowej tablicy.

Polecenie 6:

Napisz program, który w tablicy o wymiarach 10 wierszy x 10 kolumn o nazwie **tab1** umieszcza liczby losowe pochodzące z przedziału od 0 do 999. Program powinien wyświetlać postać tablicy na ekranie z zachowaniem odpowiedniego układu graficznego tej tablicy. Z uwagi na to, że w grupie losowanych liczb są liczby jedno-, dwu- i trzycyfrowe, zaproponuj odpowiedni algorytm umieszczający za wyświetlanymi liczbami odpowiednią liczbę spacji, zapewniających wyświetlenie kolumn bez widocznych przesunięć.