

# Wykład VII - tablice

niedziela, 13 maja 2018 09:19

## Tablice

**Tablica** - zbiór określonej liczby elementów tego samego typu

- elementy tablicy są umieszczone w pamięci jeden za drugim;
- dostęp: do poszczególnych elementów tablicy uzyskuje się przez podanie nazwy i indeksu;
- indeks pierwszego elementu tablicy to 0;

## Deklaracja tablicy

**typ nazwa [liczba\_elementów]**

- typ - typ elementów tablicy
- nazwa - nazwa tablicy
- liczba\_elementów - liczba elementów tablicy

### PRZYKŁADY:

```
int ListaGr[9];  
/*będzie wykorzystywana do pamietania liczb calkowitych*/  
unsigned short ints[52] \;  
/*będzie wykorzystywana do pamietania liczb calkowitych bez znaku*/  
float temperatury[31];  
/*będzie wykorzystywana do przechowywania temperatur dziennych w miesiacu*/  
char name[20];  
/*może być wykorzystywana do przechowywania napisów*/
```

## Inicjalizacja tablic

- **typ nazwa[liczba\_elementów] = {war\_p0, war\_p1,...};**
- **typ nazwa[] = {war\_p0, war\_p1, ..., war\_pm};**

### Uwagi:

- liczba elementów tablicy wynosi m+1;
- elementy lokalnej tablicy, tj. tablicy deklarowanej w funkcji, nie są standardowo inicjalizowane;
- elementy globalnej tablicy, tj. tablicy deklarowanej poza jakąkolwiek funkcją, są standardowo inicjalizowane zerami.

### PRZYKŁADY:

```
int ListaGr[10] = {10, 23, 4, 65, 77, 33, 17, 93, 27, 81};  
int ListaGr[10] = {10, 23, 4, 65, 77, 33, 17, 0, 0, 0};  
int ListaGr[] = {10, 23, 4};
```

Program	Obraz na ekranie
<pre>/*inicjalizacja tablic*/ #include &lt;stdio.h&gt;  int ListaGr[6] = {10, 5, 37}, StudLata[5], n; int main() {     unsigned short StudWydz [4];     printf("\n StudWydz\n");     for(n=0; n&lt;4; n++)     {         printf("%i ", StudWydz[n]);</pre>	<pre>StudWydz 12288 64 12292 64  StudLata 0 0 0 0  ListaGr 10 5 37 0 0 0</pre>

<pre>     }     for(n=0; n&lt;5;n++)     {         printf("%i ", StudLata[n]);     }     printf("\n\n ListaGr\n");     for(n=0; n&lt;6; n++)     {         printf("%i ", ListaGr[n]);     }     return 0; } </pre>	
--	--

## Dostęp do elementów tablicy

### **nazwa[indeks]**

- nazwa - nazwa tablicy;
- indeks - numer elementu tablicy

### **PRZYKŁADY:**

1. Nadanie wartości elementowi  
ListaGr[2] = 75;
2. Przekazanie wartości elementu do zmiennej  
a = ListaGr[2];

**Uwaga:** nadanie wartości elementowi ListaGr[3] = 175; to ingerencja w obszar pamięci poza tablicą, ale nie wystąpi błąd kompilacji.

## Przykłady

### **Poprawne operacje na tablicach:**

- int ListaGr[3];
- ListaGr[0] = a;
- ListaGr[a] = 53;
- b = ListaGr[a+2];
- ListaGr[ListaGr[a]] = ListaGr[2] +5;

PROGRAM	OBRAZ NA EKRANIE
<pre> /*wykorzystanie tablic*/ #include &lt;stdio.h&gt;  int ListaGr[] = {1, 2, 7, 40, 12071}; int n, suma = 0; int main() {     for(n=0; n&lt;5; n++)     {         suma += ListaGr[n];     }     printf("%i", suma);     return 0; } </pre>	12121

## Tablice wielowymiarowe

**typ nazwa [liczba\_elem\_1] [liczba\_elem\_2] ... [liczba\_elem\_4]**

- typ - typ elementów tablicy
- nazwa - nazwa tablicy
- liczba\_elem\_i - liczba elementów tablicy w wymiarze i

#### Przykład:

```
int Stud[2][10];
```

Tablica wielowymiarowa	Tablica jednowymiarowa	Obraz na ekranie
<pre>/* tablice wielowymiarowe */ #include &lt;stdio.h&gt;  const int Wiersze = 3; int Kolumny = 4;  int n, m;  int main() {     int stud [Wiersze][Kolumny];     Kolumny = 5;     printf("%i ", sizeof(stud));     for(n=0; n&lt;Wiersze; n++)     {         printf("\n\n");         for(m=0; m&lt;4; m++)         {             stud[n][m]=(n+1)*(m+1);             printf("%i ", stud[n][m]);         }     }     return 0; }</pre>	<pre>/*tablice pseudowielowymiarowe*/ #include &lt;stdio.h&gt;  const int Wiersze=3; int Kolumny=4;  int n,m;  int main() {     int stud [Wiersze*Kolumny];     printf("%i ", sizeof stud);     for(n=0; n&lt;Wiersze; n++)     {         printf("\n\n");         for(m=0; m&lt;Kolumny; m++)         {             stud[n*Kolumny+m]=(n+1)*(m+1);             printf("%i ", stud[n*Kolumny+m]);         }     }     return 0; }</pre>	<pre>48  1 2 3 4 2 4 6 8 3 6 9 12</pre>

### Tablice jako parametry

**typ\_fun nazwa\_fun (typ\_tab nazwa\_tab []) instrukcja**

- typ\_fun - typ wartości funkcji
- nazwa\_fun - nazwa funkcji
- typ\_tab - typ elementów tablicy
- nazwa\_tab - nazwa tablicy;

#### Przykład

Kod	Obraz na ekranie
<pre>/*tablica jako parametr funkcji*/ #include &lt;stdio.h&gt;  void printarray (int arg[], int length) {     int n;     for (n=0; n&lt;length; n++)         printf("%i ", arg[n] ++);     printf("\n\n"); }</pre>	<pre>5 10 15  2 4 6 8 10  3 5 7 9 11</pre>

```
}  
  
int main()  
{  
    int n;  
    int firstarray[] = {5, 10, 15};  
    int secondarray[] = {2, 4, 6, 8, 10};  
    printarray (firstarray, 3);  
    printarray(secondarray, 5);  
    for (n=0; n<5; n++)  
        printf("%i ", secondarray[n]);  
    return 0;  
}
```

Definicja funkcji z parametrem będącym tablicą n-wymiarową

**typ\_fun nazwa\_fun (typ\_tab nazwa\_tab [] [liczba\_elem2])**