

Wojciech Myszka

Język C: dynamiczne tablice dwuwymiarowe i funkcje

2016-05-07 09:04:46 +0200

1. Wstęp

Język C pozwala deklarować tablice w „klasyczny sposób” to znaczy określając ich typ i liczbę elementów. Deklaracja tablicy jednowymiarowej wygląda tak:

```
int tab1d [10];
```

(w nawiasach podajemy liczbę elementów).

W przypadku tablic dwuwymiarowych deklaracja wygląda tak:

```
double tab2d [10][12];
```

w pierwszym nawiasie kwadratowym podana jest liczba wierszy, w drugim liczba kolumn.

Do elementów tablic dobieramy się w standardowy (cokolwiek to znaczy) sposób, na przykład:

```
tab2d [3][5] = tab1d [7];  
tab1d [3] = round (tab2d [4][4]);
```

Natomiast zapis z dwoma nawiasami kwadratowymi sugeruje, że mamy do czynienia z bardziej skomplikowanym tworem: jednowymiarową tablicą (**adresów**) jednowymiarowych tablic.

W istocie, nawet przy powyższych deklaracjach poprawny jest zapis

```
tab2d [3]
```

na przykład

```
print (" Adres □ wiersza □ 3 □ wynosi: □%p\n" , tab2d [3]);
```

Zwracam uwagę na format wydruku %p przeznaczony do drukowania wartości wskaźników.

2. Przykład

Poniższy przykład pokazuje jak można zadeklarować dynamiczną tablicę dwuwymiarową (i przekazać jej adres do funkcji).

Najpierw deklarujemy wskaźnik, w którym będą zapamiętane wszystkie adresy:

```
double **t;
```

Jeżeli zapisać to w nawiasach **double** ***(*)** to widać że deklarujemy standardowo adres tablicy jednowymiarowej *****(*****). Druga gwiazdka mówi o tym, że elementami tej tablicy również będą adresy ***t**.

Cały przykład wygląda tak:

```
/* -*- Mode: C; indent-tabs-mode: t; c-basic-offset: 4; tab-width: 4 -*-  
*/  
* main.c  
* Copyright (C) 2012 Wojciech Myszka <myszka@asusux>  
*  
* tablice is free software: you can redistribute it and/or modify it  
* under the terms of the GNU General Public License as published by the  
* Free Software Foundation, either version 3 of the License, or  
* (at your option) any later version.  
*  
* tablice is distributed in the hope that it will be useful, but  
* WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
* See the GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License also  
* with this program. If not, see <http://www.gnu.org/licenses/>.  
*/  
#include <stdio.h>  
#include <stdlib.h>  
void dane(int, int, double **);  
  
double srednia(int N, int M, double **tablica);  
  
int main()  
{  
    int i, j;  
    /* Liczba wierszy N */  
    int N = 10;  
    /* Liczba kolumn M */
```

```

int M = 12;
/*
   Deklarujemy tablicę zawierającą adresy kolejnych wierszy...
   te dwie gwiazdki są bardzo ważne:
   pierwsza (od lewej) mówi, że będzie to tablica jednowymiarowa...
   druga - ...adresów (wskaźników)
*/
double **t;
/* ...i przydzielamy pamięć dla N adresów double: */
t = malloc( N * sizeof( double * ) );
if ( t == NULL )
{
    /* Oraz sprawdzamy, czy pamięć została przydzielona */
    printf("Nie_moge_przydzielic_pamieci!\n");
    goto koniec;
}
/* Teraz przydzielamy pamięć dla każdego wiersza */
for ( i = 0; i < N; i++ )
{
    t[i] = malloc( M * sizeof( double ) );
    if ( t[i] == NULL )
    {
        /* i sprawdzamy czy została przydzielona */
        printf("Nie_moge_przydzielic_pamieci!\n");
        goto koniec;
    }
}
/* Do elementów tablicy dobieramy się normalnie: */
t[1][10] = 100.;
dane(N, M, t);
for ( i = 0; i < N; i++ )
{
    for ( j = 0; j < M; j++ )
        printf("%.3f", t[i][j]);
    printf("\n");
}
return ( 0 );
koniec:
return 1;
}

```

```
void dane(int N, int M, double **tablica)
{
    /*
     * Tablica dwuwymiarowa w funkcji
     */
    int i, j;
    for ( i = 0; i < N; i++ )
        for ( j = 0; j < M; j++ )
            tablica[i][j] = -1;
}
```

3. Wersja PDF tego dokumentu...

... pod adresem.

Wersja: 50 z **drobnymi modyfikacjami!** data ostatniej modyfikacji 2016-05-07 09:04:46 +0200