

Alocarea dinamică a memoriei

POINTERI



1. Competențe	3
2. Noțiuni introductive.	4
3. Declararea variabilelor de tip pointer.	7
4. Pointerul NULL	9
5. Operatori pentru variabilele de tip pointer	11
6. Aplicații	16
7. Bibliografie și webografie	17



Competențe generale

- *identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea*
- *elaborarea algoritmilor de rezolvare a problemelor*

Competențe specifice

- *evidențierea necesității structurării datelor*
- *prelucrarea datelor structurate*
- *alegerea structurii de date adecvată rezolvării unei probleme*



2. Noțiuni introductive

O **structură de date** reprezintă un ansamblu de date caracterizat prin relațiile existente între ele și prin operațiile care pot fi efectuate cu datele respective.

În cazul structurilor dinamice de date, dimensiunea zonei alocate **nu este fixă**. Alocarea sau eliberarea zonei de memorie folosite de structură se face în timpul execuției programului în funcție de numărul de componente ale structurii.

Spațiul de memorie necesar variabilelor alocate dinamic se rezervă într-o zonă de memorie, special destinată, numită HEAP.

Prin **alocare dinamica a memoriei** se înțelege alocarea unor variabile în Heap, alocare care se face în timpul executării programului.



O variabilă se caracterizează prin:

- tip;
- valoare;
- adresă.

Exemplu

```
int x=7;
```

La declararea unei variabile, aceasta se memorează în RAM și ocupă un număr de octeți succesivi, număr care depinde de tipul variabilei.

Exemplu

`char` – 1 octet

`int` – 4 octeți

`long` – 4 octeți

`float` – 4 octeți



Pentru a memora adresa unei variabile se utilizează o variabilă de tip **pointer (referință)**.

POINTER = adresa unei variabile

VARIABILĂ DE TIP POINTER = variabilă care memorează adresa unei alte variabile



3. Declararea variabilelor de tip pointer

Sintaxa:

```
tip_dată *nume_variabilă;
```

unde:

tip_dată* reprezintă tipul datei stocate la adresa memorată în pointer;

nume_variabilă este identificatorul variabilei de tip pointer în care se memorează adresa unei variabile de memorie.



Exemplu 1

```
char *p;
```

- **p** este pointer către tipul **char**;
- **p** memorează adresa unei variabile de tip caracter;

Exemplu 2

```
int *x;
```

- **x** este pointer către tipul **int**;
- **x** memorează adresa unei variabile de tip întreg;



4. Pointerul NULL

În timpul execuției unui program există posibilitatea ca un pointer să memoreze adresa unei locații de memorie “goală”, în care nu se găsește nicio valoare. Valoarea acestui pointer este NULL, adică pointerul nu indică nimic.

Acest tip de pointer este folosit de obicei în două situații:

- ca terminator în diverse structuri de date;
- pentru inițializarea valorii unui pointer.



Exemplu

```
int *p;  
p=NULL;
```



5. Operatori pentru variabile de tip pointer

Pe variabilele de tip pointer se pot aplica mai multe tipuri de operatori, printre care:

- operatori specifici;
- operatorul de atribuite.



1). *Operatori specifici*

a). *Operatorul & (operator de adresare)*

- determină adresa unei variabile alocate static;

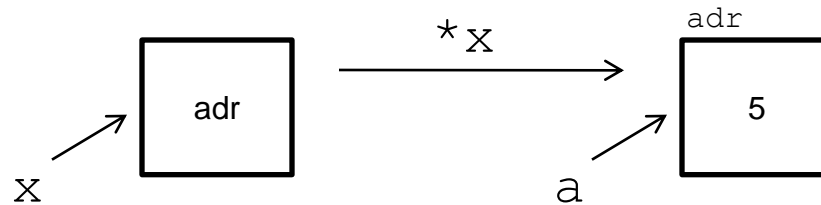
b). *Operatorul * (operator de redirectare)*

- determină valoarea variabilei de memorie care se găsește la adresa memorată în pointer;



Exemplu

```
int a;           //variabilă alocată static
int *x;         //variabilă pointer către întreg
a=5;           //inițializarea variabilei a
x=&a;          //memorarea în x a adresei variabilei a
cout<<*x;     //afișarea valorii variabilei a
```



2). *Operatorul de atribuire*

- unei variabile de tip pointer i se poate atribui numai o valoare care reprezintă o adresă de memorie;



Exemplu

```
int a=5, *p;  
p=&a;
```

//pointerului p i se atribuie adresa variabilei a

Exemplu

```
int *p, *q;  
q=p;
```

//pointerului q i se atribuie valoarea pointerului p

Exemplu

```
int a=10, *p;  
p=&a;  
*p=20;
```

//pointerului p i se atribuie adresa variabilei a
//variabilei a care are adresa memorată în
//pointerul p i se atribuie valoarea 20

Exemplu

```
int *p=NULL; //pointerului p i se atribuie constanta NULL
```



Fișă de lucru

- Aplicații



7. Bibliografie și webografie

1. Miloșescu M., *Informatică. Manual pentru clasa a X-a*, Editura Didactică și Pedagogică, București, 2005
2. Mateescu G, Moraru P., *Informatica. Manual pentru calsa a XI*, Editura Donaris, Sibiu, 2006
3. Popescu C., *Culegere de probleme de informatică*, Editura Donaris-Info, Sibiu, 2002
4. [http://en.wikipedia.org/wiki/Pointer_\(computer_programming\)](http://en.wikipedia.org/wiki/Pointer_(computer_programming))
5. <http://www.cplusplus.com/articles/EN3hAqkS/>
6. <http://www.cplusplus.com/doc/tutorial/dynamic/>
7. <http://www.cplusplus.com/articles/G6vU7k9E/>

