

# Динамични променливи



# Определения

- ▣ **Статични променливи** - променливи, за които се заделя памет преди изпълнението на програмата.
- ▣ **Динамични променливи** - променливи, за които се заделя памет по време на изпълнението на програмата.
- ▣ Динамичните променливи се създават с изрична команда от програмиста и затова негово задължение е да ги унищожи, когато вече не са му нужни.

# Заделяне на памет

общ вид:

```
указател = new тип;  
указател = new тип[размер];
```

*тип* – указва типа на динамичната променлива

*указател* – името на указателя, който ще сочи към нея

*размер* – указва броят елементи, за които се заделя памет

действие:

- заделя се памет за един или повече елементи от указания тип и указателят се насочва към първия от тях

пример:

```
int *pa = new int; // pa сочи към динамична променлива от тип int  
char *pb = new char[20]; // pb сочи към низ от 20 символа
```

# Освобождаване на памет

общ вид:

```
delete указател;  
delete[ ] указател;
```

*указател* – който сочи към динамичната променлива

действие:

- освобождава се паметта, заделена за тази динамична променлива

пример:

```
delete pa;    // освобождаваме паметта на динамичната променлива pa  
delete[] pb;  // освобождаваме динамичния масив, сочен от pb
```

# Адресна аритметика

- към указателите могат да бъдат прилагани аритметичните операции събиране и изваждане
- в този случай указателят с премества с толкова елемента от същия тип, колкото са прибавени (или извадени) към стойността му
- операторът индексирание (`[]`) също е приложим към указатели, ако те сочат към масив

пример:

```
int a[] = {1, 2, 3}, *pa; // a е масив с елем.1, 2, 3, pa е указател
pa = a;                  // pa вече сочи към a[0]
*(pa +1) = 0;            // a[1] вече има стойност 0
*(a+1) = 10;             // a[1] вече има стойност 10
pa[2] = 5;               // a[2] вече има стойност 5
pa++;                   // pa вече сочи към a[1]
```

# Пример за динамични променливи

```
int i, n, *p;
cout << "Колко числа искате да въведете? ";
cin >> n;
p = new int[n]; // заделяме блок от паметта за n броя int
               // и насочваме указателя p към първия от тях
for (i = 0; i < n; i++) {
    cout << "Въведете число: ";
    cin >> p[i];
}
cout << "Въведохте: ";
for (i = 0; i < n; i++)
    cout << p[i] << " ";
cout << endl;
delete[] p; // освобождаваме заделената памет
```

# Край

