

Функции и класове памет

Програма на С – съвкупност от равнопоставени функции – разбиват програмата на малки, лесни за управление и настройка елементи.

дефиниция <тип> <име>([списък от параметри]) { <декларации > < програмен код> }	декларация, прототип <тип> <име>([списък от параметри]); или <тип> <име>([списък от типове]);
---	--

списък от параметри - <тип> <име>[, <тип> <име> ...]

Могат да имат параметри (0 или повече) и да връщат резултат. тип **void**

Параметрите са локални автоматични променливи, които се инициализират със стойностите подадени при извикване на функциите (копиране).

Завършване на функцията

return; **return <израз>** резултатът зададен от <израз> също се копира

извикване: <име> ([списък от изрази])

```

...
int k; long mk; double dk;
...
k=func(mk*2, dk, 5);

```

```

int func(long pi,double pd, int pw){
...
return pi+pw;
}

```

еднопасов компилатор, функцията трябва да е декларирана преди извикването

Промяна на стойност на променлива в извиквана функция

<pre> int main(){ int k=4; func(k) printf("%d\n",k); } </pre> <p>извеждана стойност - 4</p>	<pre> void func(int m){ m++; } </pre>	<pre> int main(){ int k=4; func(&k) printf("%d\n",k); } </pre> <p>извеждана стойност - 5</p>	<pre> void func(int *m){ (*m)++; } </pre>
--	--	---	--

Класове памет (Storage Classes) – определят в кой сегмент на паметта се съхранява променливата, колко време тази памет е запазена за нея, областта на видимост (scope) на променливата т.е. къде в програмата променливата е достъпна. Четири класа памет – automatic, register, external, static

automatic – дефинират се в началото на блок, памет им се заделя (случайна стойност) при влизане в блока и се освобождава при излизане от него. Локални са за блока и всички включени в него блокове. Извън него – недостъпни по име. Могат да се инициализират с израз включващ променливи, като инициализацията се извършва всеки път при заделянето памет за променливата.

<pre> int main(){ int i,j; for (i=0;i<5;i++){ int k; for(j=2;j<8;j++){ int w = i*j; w=func(j); } } } </pre> <p>----- i,j ----- i,j,k ----- i,j,k,w w – 5 пъти иниц. ----- i,j,k ----- i,j -----</p>	<pre> int func(int s){ int p; ... } </pre> <p>----- s,p; i,j,k,w съществуват но са недостъпни чрез името си -----</p>
---	---

static - дефинират се в началото на блок, памет им се заделя (стойност 0) преди началото на програмата и се освобождава след края ѝ. Локални са за блока и всички включени в него блокове. Извън него – недостъпни по име. Могат да се инициализират с константен израз. Инициализацията се извършва един път преди началото на програмата. Стойността на променливата зависи и от кода извън функцията в която е дефинирана.

<pre>#include <stdio.h> void sum(int); void sum_a(int); int main(){ int count,num; printf("Please enter 3 numbers\n"); for (count=0;count<3;count++){ printf("Enter an int:"); scanf("%d",&num); sum(num); sum_a(num); } return 0; } void sum(int num){ static int sum=0; sum+=num; printf("in sum: the current total is %d\n",sum); } void sum_a(int num){ int sum=0; sum+=num; printf("in sum_a: the current total is %d\n\n",sum); }</pre>	<pre>Please enter 3 numbers Enter an int:1 in sum: the current total is 1 in sum_a: the current total is 1 Enter an int:2 in sum: the current total is 3 in sum_a: the current total is 2 Enter an int:4 in sum: the current total is 7 in sum_a: the current total is 4</pre>
--	--

Външни променливи – глобални статични променливи. Проблеми! Дефиниция и декларация.

extern

```
int e1,e2,e3;
void ex1(){
    int e1;          -----
    extern int e2;   e1 – локална, e2- външна, e3 – само ако е в същия файл, външната e1 е недостъпна
    ...
}
void ex2(){
    int e2;          -----
    extern int e1;   e1 – външна, e2- локална, e3 – само ако е в същия файл, външната e2 е недостъпна
    ...
}
```