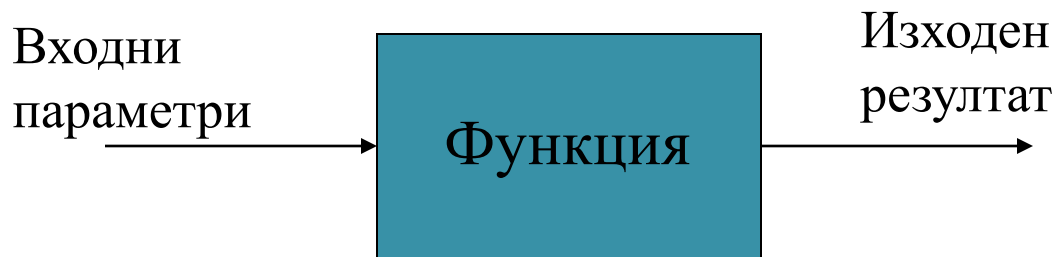


Страничен ефект

От математическа гледна точка всяка една функция изпълнява само едно действие: според зададени стойности на аргументите се изчислява определен резултат.



В езиците за програмиране (C, C++, Pascal и др.) това не е така, и работата на функциите може да бъде свързана с редица действия, които предизвикват т. н. "страничен ефект". Тези действия са коректни от гледна точка на езика (т. е. не дават съобщение за синтактична или run-time грешка), но нарушават основното правило, описано по-горе.

За всеки език съществуват определени “странични ефекти”, които пораждат класически грешки - “клопки”.

Например в C/C++:

Пример 1.

```
#include <iostream>
using namespace std;
int sum(int r);
int glob;
int main()
{
    glob=5;
    cout<<sum(1)<<" "<<sum(1)<<" "<<sum(1);
    return 0;
}
int sum(int r)
{
    int a;
    a=r*glob++;
    return a;
}
```

Пример 2:

```
#include <iostream>
using namespace std;
int avg(int a[10]);
int main ()
{
    int m[10];
    for (int i=0; i<10; i++)
        m[i]=i;
    if (avg(m)+1==1+avg(m))
        cout<<"Yes";
    else
        cout<<"No!";
    return 0;
}

int avg (int a[10])
{
    int res=0;
    for (int i=0; i<10; i++)
    {
        res+=a[i];
        a[i]--;
    }
    return res/10;
}
```

Пример 3:

```
#include <iostream>
using namespace std;
int quadr (int &n, int s);
int main ()
{
    int m=2, k=1;
    cout<<quadr(k, m)*k<<"\n"<<quadr(k, m)*k;
    return 0;
}

int quadr (int &n, int s)
{
    n++;
    return s*s;
}
```

В програмирането често се използва понятие "чиста функция". Една функция се смята за "чиста", ако тя не съдържа странични ефекти и резултат ѝ зависи САМО от параметрите ѝ.

Функциите, които съдържат входни операции, не могат да се смятат за чисти:

```
int f()
{
    cout<<"\na=";
    cin>>a;
    return a*10;
}
```

Препоръка: да се пишат и използват "чисти" функции.