

Базовые понятия языка Си

Рассматриваемый материал

- Процедуры
- Функции

Зачем нужны процедуры?

```
printf ( "Ошибка программы" );
```

много раз!

```
void Error()
```

```
{
```

```
    printf("Ошибка программы");
```

```
}
```

```
main()
```

```
{
```

```
    int n;
```

```
    scanf ( "%d", &n );
```

```
    if ( n < 0 ) Error();
```

```
    ...
```

```
}
```

ВЫЗОВ
процедуры

Что такое процедура?

Процедура – вспомогательный алгоритм, который выполняет некоторые действия.

- текст (расшифровка) процедуры записывается **после основной программы**
- в программе может быть **много процедур**
- чтобы процедура заработала, нужно **вызвать** её по имени из основной программы или из другой процедуры

Процедура с параметрами

```
void printBin ( int n )
{
    int k;
    k = 128;
    while ( k > 0 )
    {
        printf ( "%d" , n / k );
        n = n % k;
        k = k / 2;
    }
}
```

локальные
переменные

Параметры – данные,
изменяющие работу
процедуры.

```
main ()
{
    printBin ( 99 );
}
```

значение параметра
(**аргумент**)

Изменяемые параметры

Задача. Написать процедуру, которая меняет местами значения двух переменных.

```
void Swap ( int a, int b )  
{  
    int c;  
    c=a; a=b; b=c;  
}  
main()  
{  
    int x=2, y=3;  
    Swap ( x, y );  
    printf ( "%d %d", x, y );  
}
```

передача по
значению

Процедура работает с
копиями переданных
значений параметров!

Почему не работает?

2 3

Изменяемые параметры (Си)

```
void Swap ( int * adrA, int * adrB )  
{  
    int c;  
    c = *adrA; *adrA = *adrB; *adrB = c;  
}
```

передаются адреса
переменных

значение
переменной по
адресу

Вызов:

```
int a, b;  
Swap ( &a, &b );    // правильно  
Swap ( 2, 3 );     // неправильно  
Swap ( &a, b+3 );  // неправильно
```

Что такое функция?

Функция – это вспомогательный алгоритм, который возвращает *значение-результат* (число, символ или объект другого типа).

Задача. Написать функцию, которая вычисляет сумму цифр числа.

Алгоритм:

```
сумма = 0
пока n != 0
    сумма = сумма + n % 10
    n = n / 10
```


Сумма цифр числа

тип результата

```
Int sumDigits ( int n )
{
    int sum=0;
    while ( n != 0 )
    {
        sum += n % 10;
        n /= 10;
    }
    return sum;
}

main()
{
    printf ( "%d", sumDigits(12345) );
}
```

передача
результата