Пользовательские функции

Дисциплина: Конструирование программ и языки программирования

Функция — это поименованный набор описаний и операторов, выполняющих определенную задачу.

Описание функции:

```
тип имя_функции (список_переменных)
{
тело_функции
}
```

В общем виде структура программы на C++ может иметь вид:

```
директивы компилятора
тип имя_1 (список_переменных)
{
тело_функции_1;
}
тип имя_2 (список_переменных)
{
тело_функции_2;
}
…
int main (список_переменных)
{
// тело функции main может содержать операторы
вызова функций имя_1, имя_2….
тело_основной_функции;
}
```

Программа, которая выводит на экран треугольник, построенный из символов «звездочка» и «пробел»:

```
#include <iostream>
using namespace std;
void fun()
{
cout<<"* ";
}
int main ()
{
int i, j;
for (i=0; i<5; i++)
{
for (j=0; j<5-i; j++)
fun();
cout<<"\n";}
system ("pause");
return 0;
}
```

```
* * * * *
* * * *
* * *
* *
*
Для продолжения нажмите любую клавишу . . .
```

Пример функции, возвращающей значение при проверке пароля:

```
#include <iostream>
#include <string>

using namespace std;

string check_pass (string password)
{
    string valid_pass = "qwerty123";
    string error_message;
    if (password == valid_pass) {
        error_message = "Доступ разрешен.";
    } else {
        error_message = "Неверный пароль!";
    }
    return error_message;
}

int main()
{
    string user_pass;
    cout << "Введите пароль: ";
    getline (cin, user_pass);
    string error_msg = check_pass (user_pass);
    cout << error_msg << endl;
    return 0;
}
```

Программа, которая выводит таблицу умножения на заданное число.

```
#include <iostream>
using namespace std;
int func(int a, int b)
{
return (a*b);
}
int main ()
{
int i, j;
cout<<"i=";
cin>>i;
for (j=1; j<=10; j++)
cout<<i<<"*"<<j<<"="<<func(i, j)<<" ";
cout<<endl; system ("pause");
return 0;
}
```

Формальные и фактические параметры

Формальные параметры существуют в прототипе и теле определения функции. Они задаются некоторыми уникальными именами и внутри функции доступны как локальные переменные.

Фактические параметры существуют в основной программе. Они указываются при вызове функции на месте формальных.

```
int n = -25;        // глобальная переменная
int modul (int n) {  // n - формальный параметр
  if(n<0) n = -1 * n;
return n;
}

int main(void) {
  cout << modul(n);  // 25, значение глобальной переменной n будет передано в функцию
 cout << n;  // -25, но работа внутри функции пойдёт с локальной переменной n
 return 0;
}
```

Из комбинации лени и логики получаются программисты!

Atkritka.com