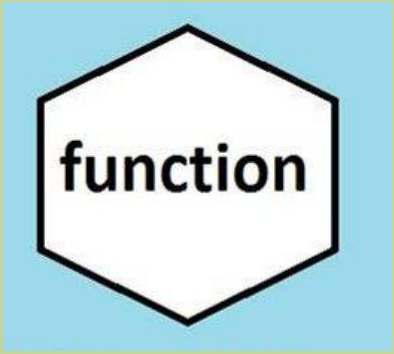


ФУНКЦИИ

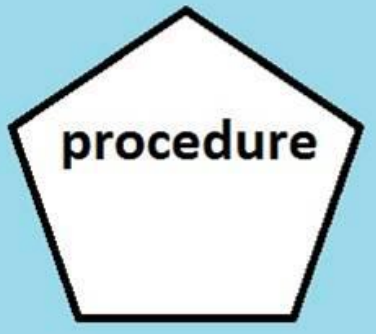
Повторение пройденного материала

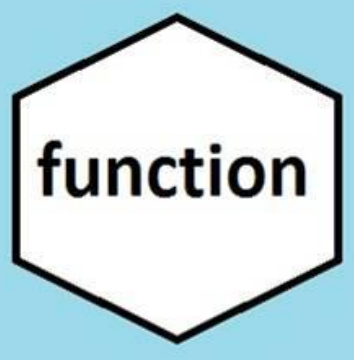
Подробное изучение функции

Повторение пройденного материала

В программировании!!!

Я процедура --- я могу просто что то обрабатывать выполнять, но я ничего не возвращаю!!!





Я функция --- я могу принимать какие то значения и на основании этих значений возвращать результат методом return ... ;

Правила описания функции

Правила использования

Правила использования функции

При использовании (ВЫЗОВЕ) функции нужно вспомнить её имя!

При использовании(ВЫЗОВЕ) функции нужно помнить о том какое значение оно возвращает!

Вызывать функцию можно только там где это уместно!

```
int victoria () {...}     // это описание

int main()
        {
                victoria();     // это вызов

                return 0;
        }
```

Борис решил использовать функцию

Майор отдает приказ (функцию).

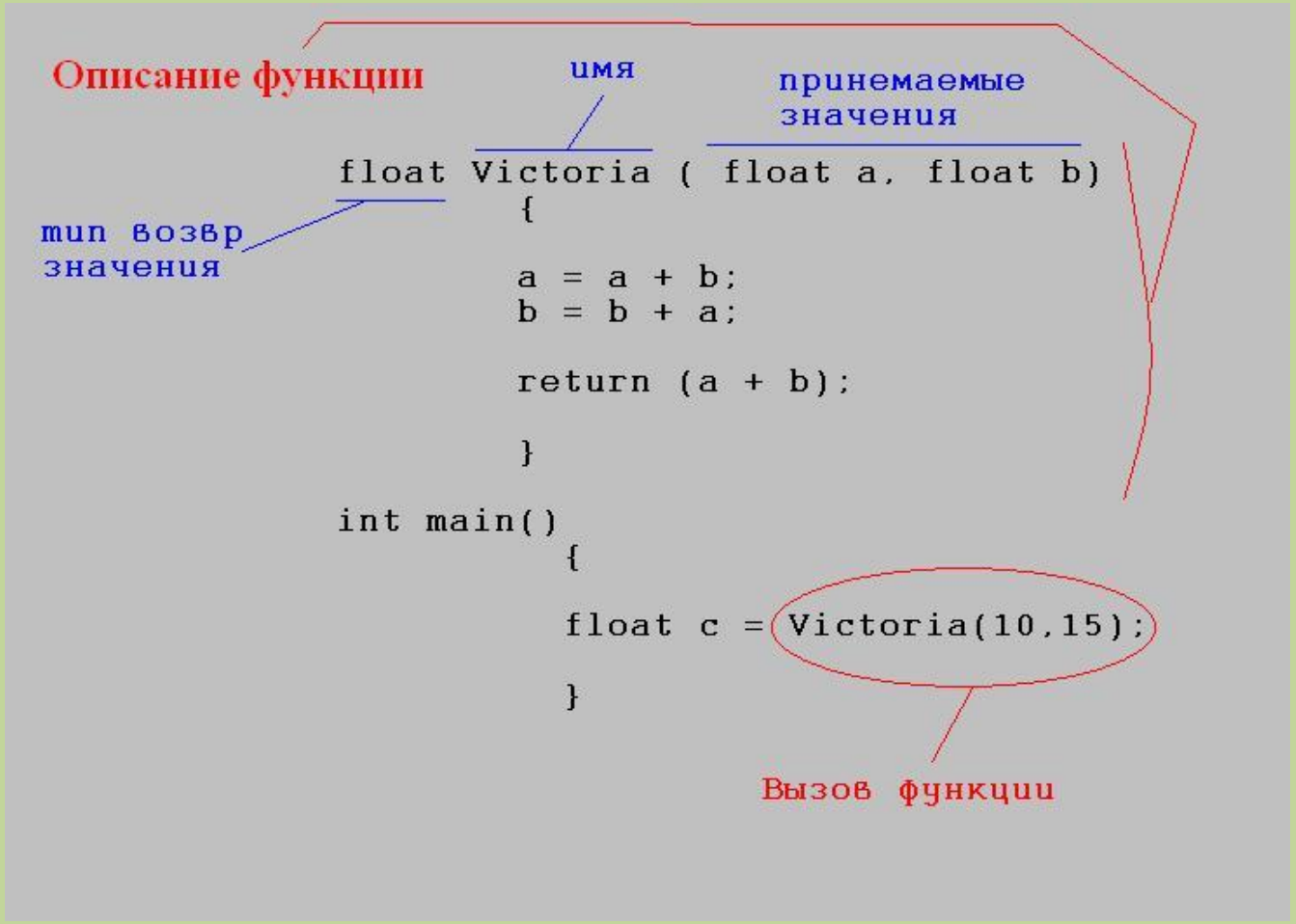
Майор отдает приказ (процедуру)

Функция по взрослому и с чем ее едят.

Функция по своей сути – это подпрограмма, которая может манипулировать данными и возвращать некоторое значение.

Каждая функция имеет свое описание (объявление) и вызов (определение).

Каждая функция имеет тип данных возврата (это как раз то что мы раньше называли фамилией) и собственное имя.


Описание функции
имя
принемаемые значения
тип возвр значения
float Victoria (float a, float b)
{
a = a + b;
b = b + a;
return (a + b);
}
int main()
{
float c = Victoria(10,15);
}
Вызов функции

пример

```cpp
#include <iostream>
#include <cmath>
#include <windows.h> // русский шрифт
using namespace std;

float Victoria (float a, float b)
    {
        a = a + b;
        b = b + a;
        return (a + b);
    }

int main()
{
    SetConsoleCP(1251);      // русский шрифт
    SetConsoleOutputCP(1251); // русский шрифт


    cout << "----------------------------  " << endl;
    cout << " ИЗУЧАЕМ ФУНКЦИИ В С++  " << endl;
    cout << "----------------------------  " << endl;

    float c = Victoria(10,15);

    cout << "Отправим в функцию Victoria числа 10 и 15 и получим " << c << endl;

    return 0;
}
```

```
C:\C++PROJECTS\new\bin\Debug\new.exe
----------------------------
 ИЗУЧАЕМ ФУНКЦИИ В С++
----------------------------
Отправим в функцию Victoria числа 10 и 15 и получим 65

Process returned 0 (0x0)   execution time : 0.281 s
Press any key to continue.
```

Описание (объявления) функции

- При помощи простого описания внутри кода программы
- При помощи записи в отдельный файл и включение описания в код программы командой #include библиотечный вариант
- Описание функции непосредственно перед вызовом.

Значение параметров по умолчанию

- В описании функции можно указать значения которые будут переданы в функцию по умолчанию.

```
float Victoria (float a = 10, float b = 15) {…}
```

- По умолчанию это тогда когда эти значения не указаны. Например так.

```
Victoria();
```

пример

```
1   #include <iostream>
2   #include <cmath>
3   #include <windows.h> // русский шрифт
4   using namespace std;
5
6   float Victoria (float a = 10, float b = 15)
7       {
8           a = a + b;
9           b = b + a;
10          return (a + b);
11      }
12
13  int main()
14  {
15      SetConsoleCP(1251);       // русский
16      SetConsoleOutputCP(1251); // русск
17
18
19      cout << "--------------------------  " << endl;
20      cout << " ИЗУЧАЕМ ФУНКЦИИ В С++  " << endl;
21      cout << "--------------------------  " << endl;
22
23      float c = Victoria();
24
25      cout << "Отправим в функцию Victoria числа 10 и 15 и получим " << c << endl;
26
27      return 0;
28  }
```

```
C:\C++PROJECTS\new\bin\Debug\new.exe
--------------------------
 ИЗУЧАЕМ ФУНКЦИИ В С++
--------------------------
Отправим в функцию Victoria числа 10 и 15 и получим 65

Process returned 0 (0x0)   execution time : 0.141 s
Press any key to continue.
```

Перегруженные функции

Перегруженные функции

В языке C++ предусмотрено создание перегруженных функций.

У перегруженной функции всегда одно и тоже имя и один и тот же тип возвращаемых данных!!!

Соответственно она описывается в отличии от обычной несколько раз!

Перегруженная функция

```
float Boris (float a) {...}

float Boris (float a, float b) {...}

float Boris (int d, int e) {...}
```

Перегруженая функция Boris

```
int main ()
        {

        float a = 10.5;
        float b = -15.7;
        int d = 4;
        int e = 2;

        float f =  Boris (d,e);
        float f =  Boris (a);
        float f =  Boris (a,b);

       return 0;
       }
```

Варианты вызова функции Boris

пример

```
1   #include <iostream>
2   #include <cmath>
3   #include <windows.h> // русский шрифт
4   using namespace std;
5
6   // ОПИСАНИЯ ПЕРЕГРУЖЕННОЙ ФУНКЦИИ БОР
7   float Boris (float a)
8       {
```

```
C:\C++PROJECTS\new\bin\Debug\new.exe
--------------------------------------
 ИЗУЧАЕМ  ПЕРЕГРУЖЕННУЮ ФУНКЦИЮ В C++
--------------------------------------
 Вызван третий вариант функции Boris (int d, int e)
 Вызван первый вариант функции Boris (float a)
 Вызван второй вариант функции Boris (float a, float b)

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

```
29  int main()
30  {
31      SetConsoleCP(1251);       // русский шрифт
32      SetConsoleOutputCP(1251); // русский шрифт
33
34      cout << "--------------------------------------  " << endl;
35      cout << " ИЗУЧАЕМ  ПЕРЕГРУЖЕННУЮ ФУНКЦИЮ В C++  " << endl;
36      cout << "--------------------------------------  " << endl;
37
38      float a = 10.5;
39      float b = -15.7;
40      int d = 4;
41      int e = 2;
42
43      Boris(d,e);  // вызываем третий вариант
44      Boris (a);   // вызываем первый вариант
45      Boris (a,b); // вызываем второй вариант
46
47      return 0;
48  }
49
```

Перегруженные функции «мутанты»

- Это такой способ создания функций который может возвращать разные значения. Т.е. каждое перегруженное описание имеет разный возвращаемый тип.
- При таком варианте объявлений легко запутаться. Поэтому не рекомендую использовать «мутации»!!!

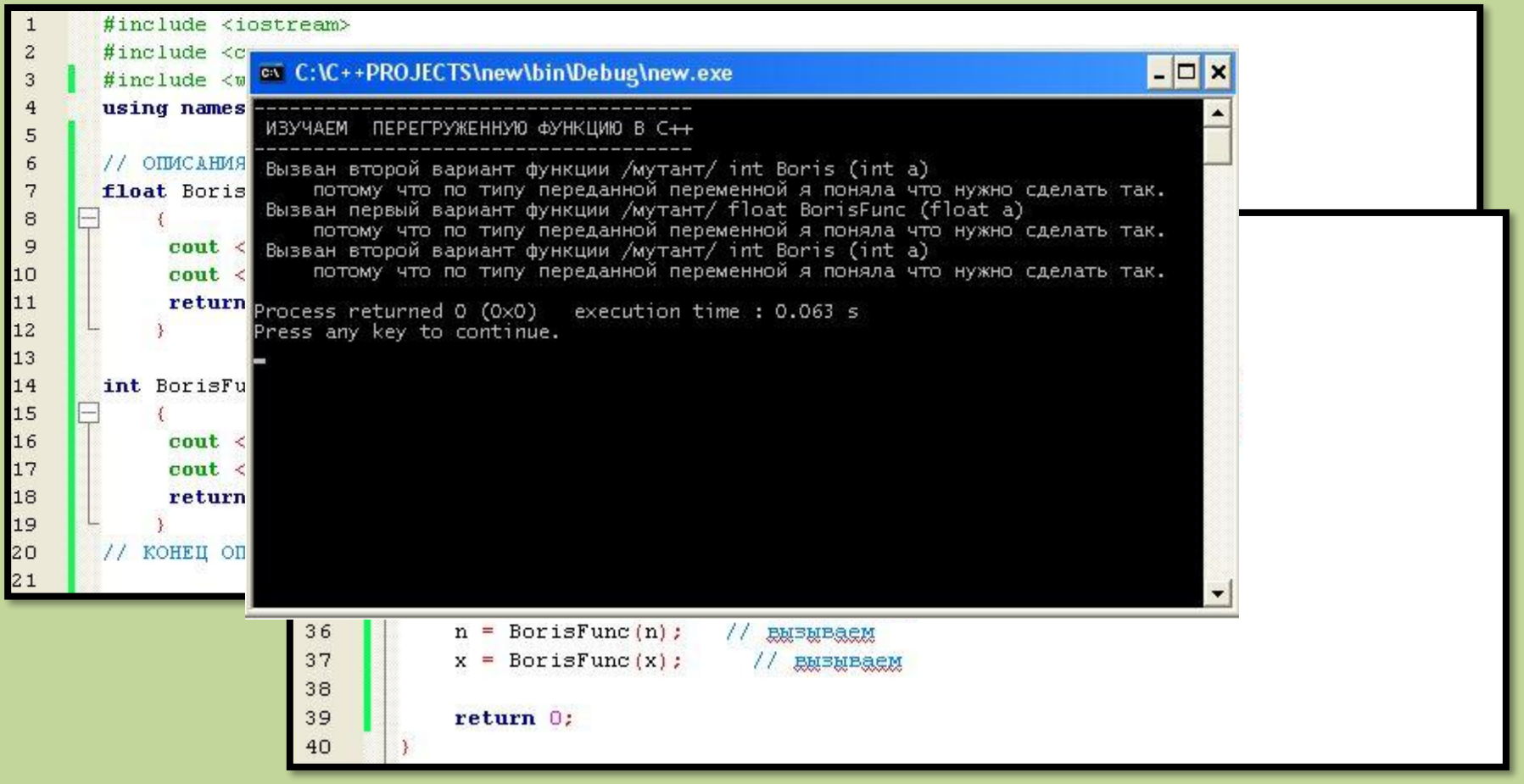
Перегруженная функция «мутант»

```
int BorisFunc (int a) {return a;}

float BorisFunc (float a) {return a;}


int main ()
            {
              float c;
              int d;

              BorisFunc (c);
              BorisFunc (d);

            return 0;

            }
```

Описание перегруженной функции "мутант"

Вызов перегруженной функции "мутант"

Перегруженная функция «мутант»



Домашнее задание

Написать программу калькулятор состоящую из функций, которые будут принимать два значения float и возвращать одно значение float.

Чтобы в этой программе без реализации «интерфейса» (средства общения с пользователем) были описаны функции расчета сложения, вычитания, умножения, деления, возведения в степень(2,3,4 – способом перегрузки), получения косинуса, синуса и тангенса, арккосинуса, арксинуса и арктангенса.

Чтобы были реализованы примеры выведения вычислений в консоль используя вызовы этих функций.