

ФУНКЦИИ

Функции, которые не возвращают значений

Пример 1. Функция без параметров и ничего не возвращает

```
#include <iostream>
using namespace std;

void my_fun1()
{
    cout << " Вывод на экран из функции. \n";
}

int main()
{
    setlocale(0, "");
    my_fun1();
    return 0;
}
```

Пример 2. Функция с параметрами. Не возвращает значения

```
#include <iostream>
using namespace std;

void my_fun2(int a, double b)
{
    cout << a<<" "<<b<< endl;
}

int main()
{
    setlocale(0, "");
    my_fun2(1024,3.1415);
    return 0;
}
```

1

Функции, возвращающие значения

Пример 3. Функция с параметрами. Возвращает значение

```
#include <iostream>
using namespace std;
int max2(int a, int b)
{
    return ((a>b)?a:b);
}

int main()
{
    setlocale(0, "");
    cout << "Введите два целых числа: " << endl;
    int n1, n2;
    cin >> n1>>n2;
    cout<<"Большее из них - это " <<max2(n1,n2)<<endl;
    return 0;
}
```

Функции с параметрами по умолчанию

Пример. Функция с параметрами по умолчанию

```
#include <iostream>
using namespace std;
double total(double salary, double tax=100.0)
{
    return salary-tax;
}

int main()
{
    setlocale(0, "");
    cout <<"Зарплата сегодня " <<total(1000) << endl;
    cout <<"Зарплата завтра " <<total(1000,50) << endl;
    return 0;
}
```

Параметры, которые имеют значения по умолчанию, должны быть указаны в заголовке функции последними. Параметры по умолчанию нужно обязательно указывать в прототипе функции.

Где можно размещать функции

Функция, расположенная перед main

```
#include <iostream>
using namespace std;
int fun(int a, int b)
{
    return a+b;
}

int main()
{
    cout<<fun(2,3)<<endl;
    return 0;
}
```

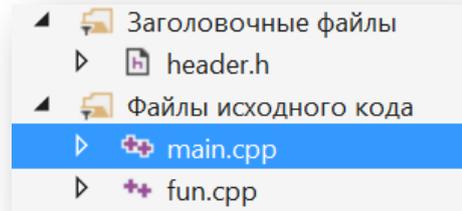
Прототип функции перед main, описание функции после main

```
#include <iostream>
using namespace std;

int fun(int a, int b);
// в прототипе необязательно указывать имена параметров
// нужно указывать их типы
int fun(int, int);

int main()
{
    cout << fun(2, 3) << endl;
    return 0;
}
int fun(int a, int b)
{
    return a + b;
}
```

Если функций много, то удобно использовать многофайловую компоновку.



МНОГОФАЙЛОВАЯ КОМПОНОВКА

header.h

main.cpp

fun.cpp

Прототипы
функций

Главная
программа

Описание
функций

main.cpp

```
#include <iostream>
// файл заголовка, в котором описан прототип функции
#include "header.h"
using namespace std;

int main()
{
    cout << myfun(2, 3) << endl;
    return 0;
}
```

header.h

```
#pragma once
int myfun(int a, int b);
```

fun.cpp

```
int myfun(int a, int b)
{
    return a + b;
}
```

Директива препроцессора

Директива **#pragma once** не позволяет включать файл в проект несколько раз, то есть предотвращает переопределение.

Перегрузка функций

Перегрузка функций позволяет создать несколько функций с одним и тем же именем, но с разными параметрами.

Пример. Перегрузка функций

```
double myfun(int a, int b)
{
    return a + b;
}
double myfun(double a, double b, double c)
{
    return a + b + c;
}

int main()
{
    cout << myfun(2, 3) << endl;
    cout << myfun(1,2,3) << endl;
    return 0;
}
```

Пример, когда перегрузка функций не работает

```
int myfun(int a, int b);
double myfun(double a, double b);
```

Задания для закрепления пройденного материала

1. Описать функцию, находящую количество цифр целого положительного числа.
2. Описать функцию, возвращающую True, если целый параметр является палиндромом (т. е. его запись читается одинаково слева направо и справа налево), и False в противном случае.
3. Описать функцию логического типа, возвращающую True, если целый параметр является квадратом некоторого целого числа, и False в противном случае.
4. Описать функцию целого типа, вычисляющую N-й элемент последовательности чисел Фибоначчи.
5. Описать функцию логического типа, возвращающую True, если целый параметр N (> 1) является простым числом, и False в противном случае.

6. Описать функцию вещественного типа, вычисляющую двойной факториал:

$$N!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot N, \quad \text{если } N \text{ — нечетное;}$$

$$N!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot N, \quad \text{если } N \text{ — четное}$$

7. Описать функцию вещественного типа (параметры x, ε — вещественные, $\varepsilon > 0$), находящую приближенное значение функции $\cos(x)$:

$$\cos(x) = 1 - x^2/(2!) + x^4/(4!) - \dots + (-1)^n \cdot x^{2n}/((2 \cdot n)!) + \dots$$

8. Описать функцию целого типа, находящую наибольший общий делитель двух целых положительных чисел A и B , используя алгоритм Евклида:

$$\text{НОД}(A, B) = \text{НОД}(B, A \bmod B), \quad \text{если } B \neq 0; \quad \text{НОД}(A, 0) = A,$$

9. **Д/З.** Выполните задания **Практик 1-3** в виде функций.

Контрольные вопросы

1. Что такое прототип функции?
2. Может ли функция ничего не возвращать?
3. Что такое параметры функции по умолчанию?
4. Что такое перегрузка функций?
5. Для чего нужен заголовочный файл?
6. Что описывается в заголовочном файле кроме прототипов функций?
7. Что такое формальные и фактические параметры функции?
8. Должны ли совпадать имена формальных и фактических параметров?
9. Должны ли совпадать типы формальных и фактических параметров?
10. Если используется многофайловая компоновка, то в каком файле описываются функции?
11. Что обычно содержит файл `main`?
12. Какие директивы препроцессора вы знаете?