



9-11 классы

## Программирование на C++

Презентация занятия

# Функции. Процедурное программирование.

6 занятие



Минцифры  
России



**20.35**  
УНИВЕРСИТЕТ

Программирование  
на C++

# Теоретическая часть

Функции.  
Процедурное программирование.

6 занятие



2020

## Тема: Функции. Процедурное программирование.

# $f(x)$

```
int main()  
{  
    |  
    return (0);  
}
```



## Тема: Функции. Процедурное программирование.

### Функции

Функция это группа выражений, которая выполняет конкретное задание.

Функции в программировании - это изолированный блок кода, обращение к которому в процессе выполнения программы может быть многократным.

Функции используют, чтобы сократить объём исходного кода: рационально вынести часто повторяющиеся выражения в отдельный блок и, затем, по мере надобности, обращаться к нему.

Вы можете определять ваши собственные функции в C++.

## Тема: Функции. Процедурное программирование.

Использование функций может дать множество преимуществ, включая следующие:

- Вы можете использовать код многократно, используя функции.
- Вы можете легко тестировать отдельные функции.
- Если есть необходимость сделать некоторые изменения в программе, вы можете модифицировать отдельную функцию, не изменяя структуру программы.
- Вы можете использовать одну и ту же функцию для ввода разной информации.



## Тема: Функции. Процедурное программирование.

В каждой исправной программе на языке C++ должна быть хотя бы одна функция - функция `main()`.

“Точка входа в программу”

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      |
7      |         return (0);
8  }
```

## Тема: Функции. Процедурное программирование.

### Возвращаемый тип

Функция `main` имеет следующий общий вид:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      |
7      |     return (0);
8  }
```

Возвращаемый тип функции объявляется до ее имени. В вышеуказанном примере возвращаемый тип `int`, который указывает, что функция возвращает целочисленное значение.

Иногда, функция выполняет необходимые операции без возвращения значения. Такие функции определяются с помощью ключевого слова `void`.

## Тема: Функции. Процедурное программирование.

### Возвращаемый тип

void это базовый тип данных, имеющий пустое множество значений.

### Определяем функцию

Для определения функции в языке C++ используется следующий синтаксис:

```
3   return_type function_name( parameter list )
4   {
5       |   body of the function
6   }
```



## Тема: Функции. Процедурное программирование.

```
3   return_type function_name( parameter list )
4   {
5       |   body of the function
6   }
```

**return-type**(возвращаемый тип): Тип данных возвращаемый функцией.

**function name**: Имя функции.

**parameters**(список параметров): При вызове функции, вы передаете значения в параметр. Это значение ссылается на актуальный параметр или аргумент. Для каждого параметра указывается его тип и имя (в объявлении имени можно опускать). Параметры опциональны; это значит, что у вас может быть функция без параметров.

**body of the function**(тело функции): Набор выражений, который определяет, что делает функция.

## Тема: Функции. Процедурное программирование.

### Определяем функцию

Для примера, давайте определим функцию, которая не возвращает значение, а только выводит строку текста на экран.

Наша функция, названная `print_hello`, возвращает `void` и не имеет параметров.

Теперь мы используем нашу функцию в функции `main()`.

Вы должны объявить функцию перед ее вызовом.

```
1  #include <iostream>
2  using namespace std;
3
4  void print_hello()
5  {
6      cout << "Hello!" << endl;
7  }
8
9  int main()
10 {
11     print_hello();
12
13     return (0);
14 }
```



**Тема: Функции. Процедурное программирование.**

**Размещение объявления функции  
после `main()` приведет к ошибке.**



## Тема: Функции. Процедурное программирование.

num\_1, num\_2 - параметры  
функции

1, 10 - аргументы функции

$f(x)$

```
1  #include <iostream>
2  using namespace std;
3
4  void sum(int num_1, int num_2)
5  {
6      int sum;
7
8      sum = num_1 + num_2;
9      cout << sum << endl;
10 }
11
12 int main()
13 {
14     sum(1, 10);
15
16     return (0);
17 }
```



## Тема: Функции. Процедурное программирование.

```
1  #include <iostream>
2  using namespace std;
3
4  void sum(int num_1, int num_2)
5  {
6      int sum;
7
8      sum = num_1 + num_2;
9      cout << sum << endl;
10 }
11
12 int main()
13 {
14     sum(1, 10);
15
16     return (0);
17 }
```

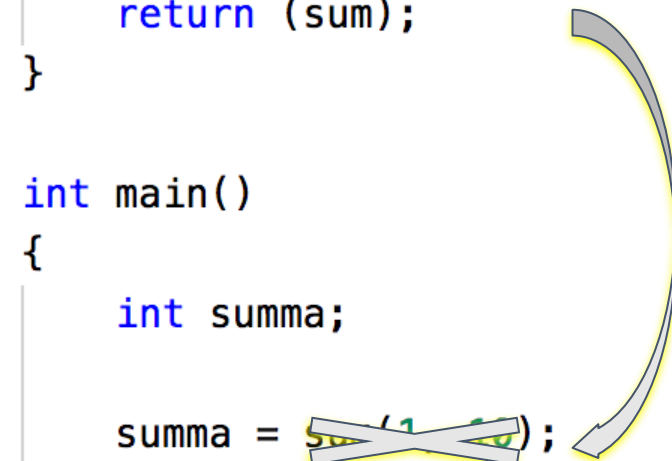


## Тема: Функции. Процедурное программирование.

Давайте создадим простую функцию, которая возвращает сумму двух параметров.

Функция `sum` принимает два параметра типа `int`, и возвращает `int`.

```
1  #include <iostream>
2  using namespace std;
3
4  int sum(int num_1, int num_2)
5  {
6      int sum;
7
8      sum = num_1 + num_2;
9      return (sum);
10 }
11
12 int main()
13 {
14     int summa;
15
16     summa = sum(1, 20);
17     cout << summa << endl;
18     return (0);
19 }
```



## Тема: Функции. Процедурное программирование.

### Множество параметров

Вы можете определить в функции столько параметров, сколько вы захотите, разделяя их запятыми.

(Суммирование не двух чисел, а трёх, четырёх и т д)



## Тема: Функции. Процедурное программирование.

Объявление(declaration) функции необходимо при ее определении(definition) в одном исходном файле и вызове в другом файле. В таком случае, вам необходимо объявить функцию вверху файла, вызывающего функцию.





Программирование  
на C++

# Практическая часть

Функции.  
Процедурное программирование.

6 занятие



2020



## Тема: Функции. Процедурное программирование.

### Задание 1

Написать программу, в которой будет функция (помимо функции `main`), которая принимает 2 аргумента (2 целочисленных числа) и вывод в стандартный выходной поток (печатает на экран) частное двух чисел.

\*

Обработать случай нулей





## Тема: Функции. Процедурное программирование.

### Задание 2

Программа должна переводить число, введенное с клавиатуры в метрах, в километры (создать отдельную функцию!)



## Тема: Функции. Процедурное программирование.

### Задание 3

Представьте, что вы создаёте собственную игру. Вам нужно реализовать печать карты на экран (под картой в вашей игре понимается прямоугольник с заданными размерами, ограниченный символом '#') Напишите функцию, которая будет реализовывать данное поведение.

```
void print_map(int width, int height);
```

\*

Добавьте возможность выбора символа границы карты (ещё один аргумент в функции)

```
// 4 x 3
####
#  #
####

// 2 x 2
##
##

// 5 x 5
#####
#  #
#  #
#  #
#####

// 1 x 5
#####
```



## Тема: Функции. Процедурное программирование.

```
void print_map(int ht, int wh)
{
    for(int y = 0; y < ht; y++)
    {
        for (int x = 0; x < wh; x++)
        { ...
        }
        cout << endl;
    }
}
```





## Тема: Функции. Процедурное программирование.

```
void print_map(int ht, int wh, char border)
{
    for(int y = 0; y < ht; y++)
    {
        for (int x = 0; x < wh; x++)
        {
            if (y == 0 || y == ht - 1)
                cout << border;
            else if (x == 0 || x == wh - 1)
                cout << border;
            else
                cout << ' ';
        }
        cout << endl;
    }
}
```



## Тема: Функции. Процедурное программирование.

### Задание 2

Добавьте замок на карту  
(дополнительные параметры  
для координат и символа)

```
Please, enter height and width
10
10
Enter a char for representation of border
#
Enter the coords of the castle
5
5
Enter a char for representation of the castle
%

#####
#           #
#           #
#           #
#           #
#    %    #
#           #
#           #
#           #
#####
```





## Тема: Функции. Процедурное программирование.

```
int main()
{
    system("CLS");
    int width, height, x_c, y_c;
    char ch_of_castle, ch_of_border;
    cout << "Please, enter height and width";
    cin >> width;
    cin >> height;
    cout << "Enter a char for representation of border" << endl;
    cin >> ch_of_border;
    cout << "Enter the coords of the castle" << endl;
    cin >> x_c;
    cin >> y_c;
    cout << "Enter a char for representation of the castle" << endl;
    cin >> ch_of_castle;
    cout << endl;

    print_map(height, width, ch_of_border, x_c, y_c, ch_of_castle);

    return 0;
}
```







## Тема: Функции. Процедурное программирование.

```
void print_map(int ht, int wh, char border, int x_c, int y_c, char castle)
{
    for(int y = 0; y < ht; y++)
    {
        for (int x = 0; x < wh; x++)
        {
            if (y == 0 || y == ht - 1)
                cout << border;
            else if (x == 0 || x == wh - 1)
                cout << border;
            else if (x == x_c && y == y_c)
                cout << castle;
            else
                cout << ' ';
        }
        cout << endl;
    }
}
```



## Тема: Функции. Процедурное программирование.

### Задание 2



*Пример расположения дисков для шифрования.*



## Тема: Функции. Процедурное программирование.

### Задание 4

Реализовать шифр Цезаря. Написать функцию, которая будет применять шифр Цезаря к строке, переданной в качестве параметра функции. Реализовать для букв нижнего регистра. Смещение по алфавиту - 10 позиций

\*Для верхнего регистра

