Лекция 5

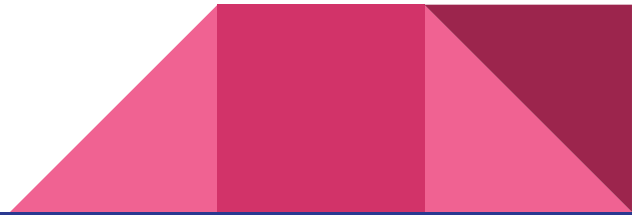
# Структура программы на языке C++

# Структура программ

**Структура программ** - это разметка рабочей области (области кода) с целью четкого определения основных блоков программ и синтаксиса.

Структура программ несколько отличается в зависимости от среды программирования.

Мы ориентируемся на IDE Microsoft Visual Studio



# Структура программ для Microsoft Visual Studio

1. **// struct_program.cpp**: определяет точку входа для консольного приложения.
2. **#include "stdafx.h"**
3. //здесь подключаем все необходимые препроцессорные директивы
4. **int main() {** // начало главной функции с именем main
5. //здесь будет находится ваш программный код
6. **}**

С 4-й по 6-ю строки объявлена функция main.

# Структура программ для Microsoft Visual Studio

Строка 4 – это заголовок функции, который состоит из типа возвращаемых данных (в данном случае int), этой функцией, и имени функции, а также круглых скобок, в которых объявляются параметры функции.

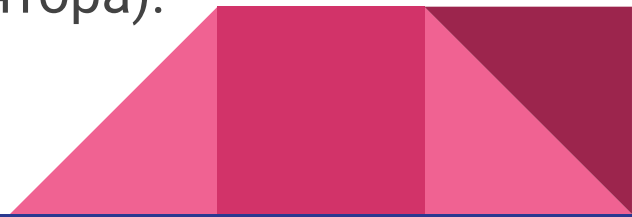
**int** — целочисленный тип данных

Между фигурными скобочками размещается основной программный код, называемый еще телом функции. Это самая простая структура программы.

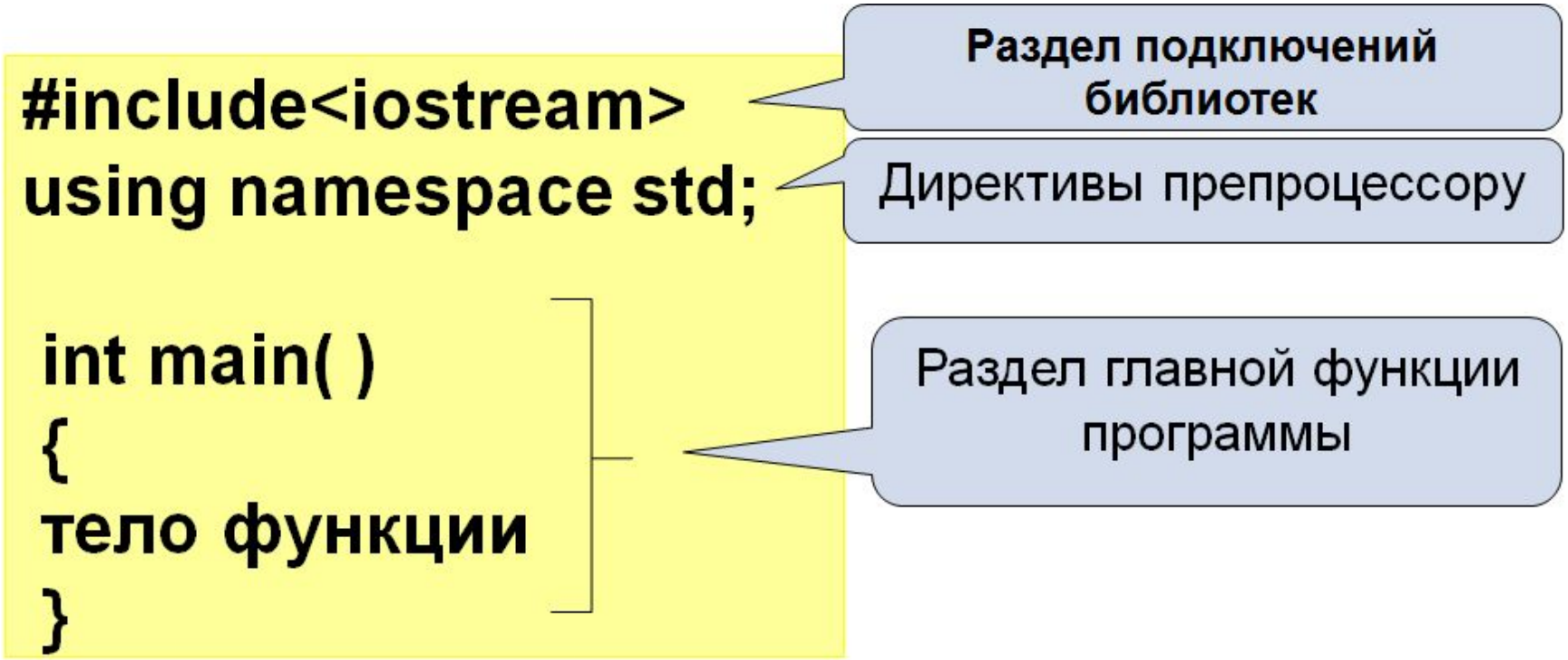
# Программа на языке C++ состоит из:

1. директив препроцессора,
2. указаний компилятору,
3. объявлений переменных и/или констант,
4. объявлений и определений функций.

**Препроцессор** — это компьютерная программа, принимающая данные на входе и выдающая данные, предназначенные для входа другой программы (например, компилятора).



# Структура программы на С++



# Директива препроцессора –

это инструкция, которая включает в текст программы файл, содержащий описание множества функций, что позволяет правильно компилировать программу.

**Это важно**

- все директивы препроцессора начинаются со знака #;
- после директивы препроцессора точка с запятой не ставится.

# Синтаксис подключения заголовочных файлов:

Директива #include позволяет включать в текст программы указанный файл. Имя файла может быть указано двумя способами:

**#include <some_file.h>**

**#include "my_file.h”**

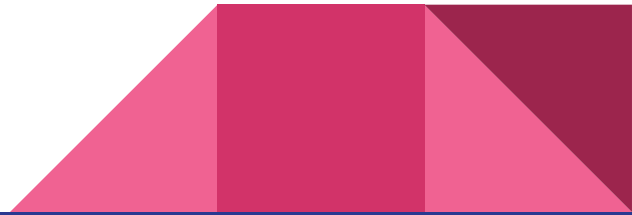
Если файл является стандартной библиотекой и находится в папке компилятора, он заключается в угловые скобки <>. Если файл находится в текущем каталоге проекта, он указывается в кавычках "".

# Стандартная Библиотека —

коллекция классов и функций, написанных на базовом языке.

Основные заголовочные файлы:

- iostream – потоки ввода/вывода
- fstream – файловые потоки
- sstream – строковые потоки



# Пространства имен (namespace)

Директива using открывает доступ к пространству имен (англ. namespace) std, в котором определяются средства стандартной библиотеки языка C++.

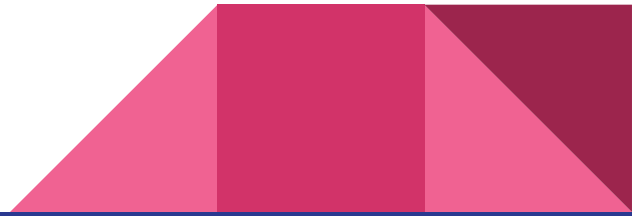
**using namespace std**

# Функция main()

Выполнение программы начинается со специальной стартовой функции **main**.

В момент запуска программы, управление передается данной функции.

Функция **main** обязательно должна быть определена в одном из модулей программы. Модуль, содержащий функцию **main** принято называть главным модулем.



# Функция main()

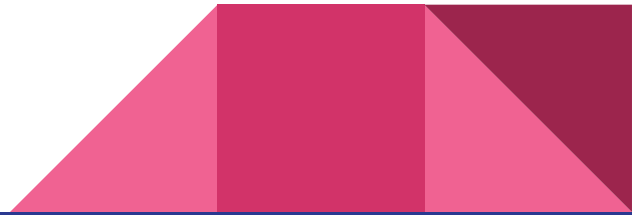
Стандарт предусматривает два формата функции:

//без параметров

тип **main( ){/* … */}**

//с двумя параметрами

тип **main(int argc, char* argv[]){/* … */}**

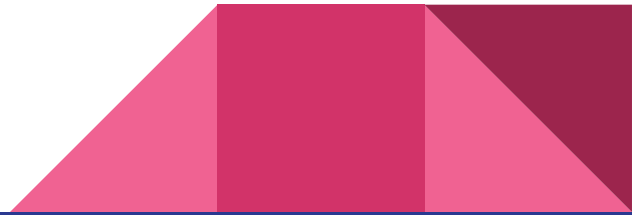


# Функция main()

Функция **main** может возвращать определенное значение, или не возвращать ничего.

Если функция не возвращает никакого значения, то она должна иметь тип **void** (такие функции иногда называют процедурами)

Функция может возвращать значение, тип которого в большинстве случаев аналогично типу самой функции.



# Директива #define

Директива **#define** служит для поиск и замена одного набора символов на другой.

Идентификаторы, заменяющие текстовые или числовые константы, называют **именованными константами**.

Идентификаторы, заменяющие фрагменты программ, называют **макроопределениями.**

Директива **#define** имеет две синтаксические формы:

**#define идентификатор текст**

**#define идентификатор (список параметров) текст**

# Объявление переменных

Язык C++ требует явного объявления всех переменных используемых в программе вместе с указанием соответствующих им типов.

Объявления переменной имеет следующий формат:

**<спецификатор типа> имя_1, имя_2, ..., имя_n;**

**Спецификатор типа** – одно или несколько ключевых слов, определяющие тип объявляемой переменной.

# Объявление переменных

**Глобальные переменные** описываются вне функций и действуют от конца описания до конца файла.

**Локальная переменная** описывается внутри функции и действует от конца описания до конца функции.