

# ОСНОВЫ C++

# Сборка программы

Сборка программы производится в несколько этапов.

1. Препроцессинг
2. Компиляция
3. Линковка (компоновка)

Компиляция условно делится на следующие этапы.

1. Лексический анализ: `.c/.cpp` -> лексемы
2. Синтаксический анализ: лексемы -> абстрактное синтаксическое дерево
3. Семантический анализ: АСД -> конкретное синтаксическое дерево
4. Оптимизация: упрощение построения в соответствии с допущениями конструкций
5. Генерация объектного кода: КСД -> `.o`

# Преппроцессинг

При разработки языка Си было решено обрабатывать код в 2 этапа - перед непосредственной компиляцией, программа подается на вход программе, именуемой **препроцессором**. В его задачи входит обработка инструкций подключения (`#include`), условной компиляции (`#ifdef`, `#if`, ...), макроподстановок (`#define`), а также ошибок, предупреждений (`#error`, `#warning`) и некоторых других, которые определяются еще до этапа компиляции программы.

Все инструкции препроцессора начинаются с решетки.

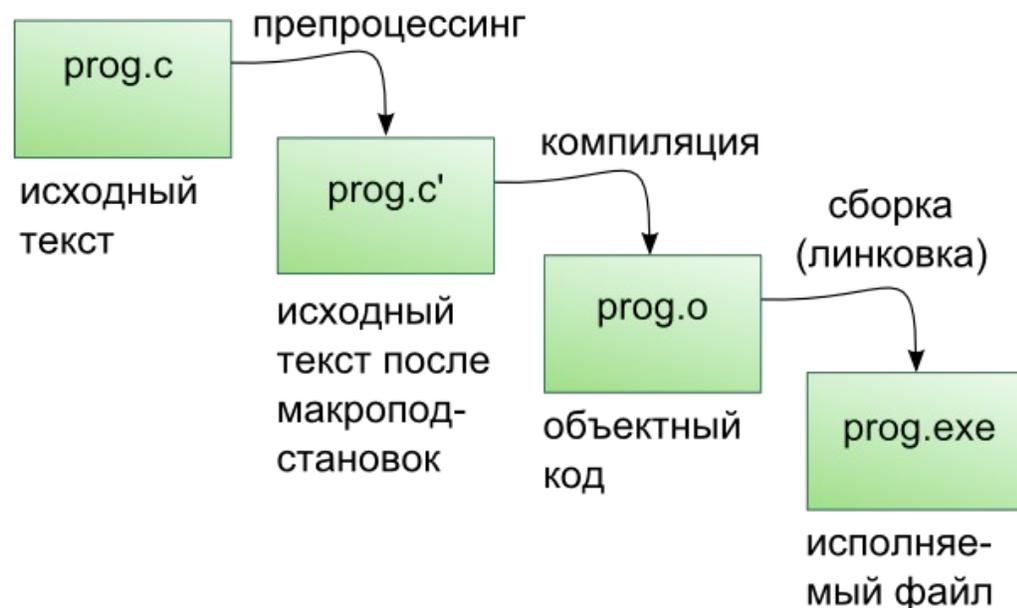
На данный момент нас интересуют только инструкции

`#include <library>` - подключает библиотеку `library` из заранее указанных библиотечных директорий

`#include "header"` - подключает заголовочный файл `header` из директории с проектом

# Компиляция и компоновка

В процессе компиляции происходит ряд преобразований над файлами исходного кода для получения промежуточного результата - объектного файла. Объектные файлы состоят из кусков конечного машинного кода и информации, необходимой для связывания объектных файлов воедино, а также присоединения библиотек. Этот последний этап называется компоновкой или линковкой.



# Библиотека `iostream`

Библиотека ввода-вывода. Содержит базовые классы и функции для работы с **вводом** (получение информации с клавиатуры-мыши и прочих устройств ввода) и **выводом** (передача информации экрану (консоли), а также прочим устройствам вывода). Все объекты стандартных библиотек C++ помещены в специальное пространство имен `std`. Потому для обращения к таким объектам нужно либо

- ▶ Указывать перед названием каждого такого объекта `std::`, где двойное двоеточие называется оператором расширения области видимости.
- ▶ Один раз использовать директиву `using namespace std;` которая позволит обращаться к объектам только по их имени (идентификатору).

Нас интересуют объекты

`std::cin` - объект потокового ввода из консоли

`std::cout` - объект потокового вывода в консоль

# Типы данных

Основными типами данных являются:

Целые числа (Ключевое слово **unsigned** делает беззнаковым):

- ▶ **int, long** (4 байта)
- ▶ **short** (2 байта)
- ▶ **long long** (8 байт)

Вещественные числа:

- ▶ **float** (4 байта)
- ▶ **double** (8 байт)

Символьный тип - **char** (1 байт) - число, соответствующее символу в кодировке ASCII.

Логический тип - **bool** (1 байт) - принимает два значения - **true** (1, истина) и **false** (0, ложь).

**void** - используется в качестве типа возвращаемого значения, говоря этим, что функция ничего не возвращает.

# Переменные

Переменной называется ячейка памяти, которая может хранить значение заданного типа.

Объявление переменной (без инициализации)

Тип переменной	Имя переменной	;
----------------	----------------	---

Определение переменной (ранее объявленной) / инициализация

Имя переменной	=	Выражение или значение	;
----------------	---	------------------------	---

Объявление с инициализацией

Тип переменной	Имя переменной	=	Выражение или значение	;
----------------	----------------	---	------------------------	---

<code>int a; char b; double c;</code>	<code>a = 0; b = 'X'; c = 3.14;</code>	<code>int a = 0; char b = 'X'; double c = 3.14;</code>
---	--	--

# Точка входа и функции

В качестве точки входа в программу используется функция `int main()`.

Функцией называется участок кода, который может быть вызван с заданными аргументами, а так же возвращает значение заданного типа.

Возвращаемое значение	Название функции	(Аргументы функции)	{Алгоритм}
-----------------------	------------------	---------------------	------------

Аргументы - это объявления переменных через запятую.

Для возврата значения, а также прекращения выполнения функции используется оператор `return`.

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

# Литералы

Целые числа:

- ▶ Десятичные пишутся как есть (123)
- ▶ Шестнадцатеричные с приставкой **0x** (0xBEEF)
- ▶ Двоичные - с **0b** (0b01011100)

Вещественные числа:

- ▶ Через разделительную точку (3.14)
- ▶ Экспоненциальная форма (314e-2)

Символы задаются в одинарных кавычках.

Строчки - двойными кавычками.

Есть модификаторы, которые ставятся на конец числовых литералов:

- ▶ L - long
- ▶ LL - long long
- ▶ U - unsigned
- ▶ F - float