



# Язык программирования С. Общие сведения

## Лекция 3

*Иллюстративный материал к лекциям  
по алгоритмизации и  
программированию*

Автор Саблина Н.Г.

2016 г.



# Содержание



Язык C – история создания, общие сведения

Алфавит языка C

Составляющие алфавита

Лексемы языка C

Идентификаторы и константы

Комментарий

Особенности компиляции в C

Общая структура программ на языке C

Автор



# История появления языка С (1)

Язык программирования С разработан сотрудниками фирмы **Bell Labs** Деннисом Ритчи и Кеном Томпсоном в 1972 г. во время их совместной работы над операционной системой UNIX на ЭВМ PDP-11, т.е. **первоначально** этот язык был задуман как **язык системного программирования**.

Затем язык был достроен концепциями объектно-ориентированного программирования( версия С++), практически не меняя старой синтаксической и семантической основы.

# История появления языка С (2)

Язык программирования С++ был разработан **Бьерном Страуструпом** (Bjarne Stroustrup) в начале 80-х годов. *На первых этапах разработки (1980 г.) язык носил условное название «Си с классами», а в 1983 г. Рик Маскитти придумал название «С++».*

Язык С++ является расширением языка С, поэтому программы, написанные на С, могут обрабатываться компилятором языка С++.

В программах на языке С++ можно использовать тексты на языке С и обращаться к библиотечным функциям языка С



# История появления языка С (3)

- Сегодня язык С – структурированный, модульный, компилируемый, универсальный язык, традиционно используемый для системного программирования.
- Он является переносимым языком, так как прикладные программы, написанные на нем, могут быть легко перенесены с одного компьютера на другой, даже если они имеют различные операционные системы.
- В настоящее время ни одна инструментальная операционная система не может считаться полной, если в ее состав не входит компилятор языка С.





# Особенности языка C (1)

- Среди современных алгоритмических языков языки C и C++ занимают первое место по распространенности и разнообразию версий
- C и C++ среди известных универсальных языков как наиболее концептуально целостные, т.е. учитывают:
  - отлаживаемость программ (быстрый поиск ошибок),
  - гибкость языка при внесении текущих изменений в программу, возможности дальнейшего развития самого языка и его средств программистом и т.д.
  - Язык C является преемником оригинальных решений, воплощенных ранее в языках Ассемблер, Фортран, Алгол.





# Особенности языка С (2)

С и С++ относятся к семейству универсальных языков программирования, т.е.

- ориентирован на широкий круг задач, решаемых при помощи ЭВМ: используется как для **прикладных задач**, так и для **системного программирования**
- Обладает набором средств, присущих современным языкам программирования **высокого уровня**:
  - структурность
  - модульность
  - определяемые типы данных
- в него включены средства для **программирования на уровне Ассемблера** :
  - указатели
  - побитовые операции
  - операции сдвига





# Алфавит языка С

**Алфавит** языка программирования называют набор символов (разрешенный к использованию и воспринимаемый компилятором), с помощью которого могут быть образованы величины, выражения и операторы данного языка

Алфавит языка включает в себя все символы, представленные в таблице кодов ASCII.

- Символы с кодами **от 0 до 127** - **основная таблица кодов ASCII**.
- Их состав и порядок определены американским стандартом на коды обмена информацией.
- Основная таблица кодов ASCII идентична на всех IBM-совместимых компьютерах.
- Символы с кодами **от 128 до 255** - **дополнительная таблица**
- стандартом не регламентируется и, в зависимости от назначения компьютера, содержит различные наборы символов. В частности, она может включать национальный алфавит (например, русский).







# Составляющие алфавита (1)

1. Заглавные и строчные латинские буквы. *Язык C является чувствительным к регистру (case sensitive). Например, имена COLOR, Color и color определяют три различных имени переменных.*
2. Символы-разделители:
  - пробел (ASCII-код 32); основное назначение этого символа – разделение ключевых слов и имен;
  - управляющие символы (ASCII-коды от 0 до 31); применяются при описании строчных символьных констант;
  - управляющие символы с ASCII-кодами: 9 (табуляция), 10 (вниз), 13 (клавиша Enter) используются в качестве разделителей при написании программ.



# Составляющие алфавита (2)

3. Специальные символы, выполняющие определенные функции при построении различных конструкций языка:
  - + - \* / % \ = { } [ ] ( ) < > . , ' " : | ; ^ & # ~ ! ?
  - символ подчеркивания (ASCII-код 95)
  
4. Десять арабских цифр от 0 до 9 (*в идентификаторах цифры могут быть использованы наряду с буквами, начиная со второй позиции*).



# Лексемы языка C

Из символов алфавита формируются лексемы языка:

- *идентификаторы,*
- *ключевые (служебные) слова,*
- *константы,*
- *комментарии.*





# Идентификаторы

- *Идентификаторы* используются для обозначения имен переменных, функций и меток, применяемых в программе.
- **Идентификатор** - произвольная последовательность латинских букв (прописных и строчных), цифр и символа подчеркивания, которая начинается с буквы или символа подчеркивания.
- В языке C идентификатор может состоять из произвольного числа символов, однако два идентификатора считаются различными, если у них различаются первые 32 символа. В языке C++ это ограничение снято.
- Прописные и строчные буквы в идентификаторах различаются, т.е. различными считаются имена: Var1, var1 и VAR1.





# Ключевые слова (1)

В языках С и С++ некоторые идентификаторы употребляются как ключевые (служебные) слова (keywords), которые имеют специальное значение для компилятора. Их употребление строго определено, и эти слова не могут использоваться иначе.

Ключевыми словами стандарта ANSI языка С являются

<b>auto</b>	<b>double</b>	<b>int</b>	<b>struct</b>
<b>break</b>	<b>else</b>	<b>long</b>	<b>switch</b>
<b>case</b>	<b>enum</b>	<b>register</b>	<b>typedef</b>
<b>char</b>	<b>extern</b>	<b>return</b>	<b>union</b>
<b>const</b>	<b>float</b>	<b>short</b>	<b>unsigned</b>
<b>continue</b>	<b>for</b>	<b>signed</b>	<b>void</b>
<b>default</b>	<b>goto</b>	<b>sizeof</b>	<b>volatile</b>
<b>do</b>	<b>if</b>	<b>static</b>	<b>while</b>





## Ключевые слова (2)

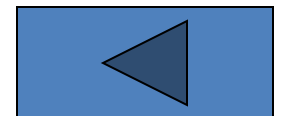
Каждый компилятор может увеличивать количество ключевых слов, так как компилятор учитывает дополнительные возможности того типа компьютеров, для которых он создан.

- *Например, компилятор Borland C++ 3.1 добавляет к ключевым словам стандарта языка C дополнительные слова, предназначенные для работы с памятью и регистрами процессоров семейства Intel, а также позволяющих использовать прерывания и фрагменты программ на другом языке. [Табл. 1](#)*
- *Язык C++ имеет дополнение к ключевым словам языка C [Табл. 2](#)*



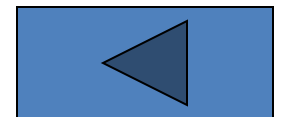
# Ключевые слова компилятора Borland C++ 3.1

asm	_asm	__asm	cdecl
_cdecl	__cdecl	_cs	__cs
_ds	__ds	_es	__es
_export	__export	far	_far
__far	_fastcall	__fastcall	huge
_huge	__huge	interrupt	_interrupt
__interrupt	_loadds	__loadds	near
_near	__near	pascal	_pascal
__pascal	_saveregs	__saveregs	_seg
__seg	__ss	__ss	



# Дополнительные ключевые слова языка C++

asm	catch	class	friend	inline	new
operator	private	protected	public	template	this
throw	try	virtual			







# Константы

*Константа – это лексема, представляющая изображение фиксированной величины, которая не может быть изменена в программе.*

Константы делятся на:

- целые (десятичные, восьмеричные, шестнадцатеричные)
- вещественные (с плавающей точкой),
- перечислимые,
- символьные (литеральные)
- строковые.

[Например:](#)



# Примеры констант

**123** целая константа;

**123.23**, **4.34E-3** вещественные константы;

**'S'**, **'a'** символьные константы;

**"Computer"** строковая константа.

Шестнадцатеричные

Восьмеричные

константы

константы

**0xFFFF**

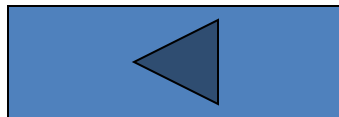
**01**

**0x10**

**055**

**0x1F1A**

**07777**





# Комментарий

Часть программы, которая игнорируется компилятором и служит для удобочитаемости исходного текста программы.

- **`/*` это длинный комментарий,  
расположенный на нескольких строках `*/`**
- **`//`это короткий однострочный комментарий**





# В комментариях можно указать

- имя файла, в котором хранится программа,
- краткое описание назначения программы,
- авторские права,
- используемый язык программирования и версия компилятора,
- версия программы и дата последнего изменения,
- пояснение назначения переменных,
- замечания о назначении операторов и частей программы





# Общая структура



# программ и роль функций в языке C

- Любая программа, написанная на языке C, состоит из одной и более функций, одна из которых должна иметь имя **main**, и именно ей передается управление из операционной системы.
- **Функция** - это самостоятельная единица программы, созданная для решения конкретной задачи.
- Для вызова функции нужно указать ее имя.
- Доступ к функции может быть осуществлен из любого места главной функции **main**, а также из любой функции, описанной ниже





# Обобщенная структура программы на языке C

```
#<Директивы препроцессора>  
<Описание глобальных переменных, констант>  
// Объявление прототипов функций  
function1( );  
function2( ); ...  
main( )  
{  
<тело функции main>  
}  
function1( )  
{  
<тело функции function1>  
}  
function2( )  
{  
<тело функции function2>  
}
```



# Директивы препроцессора

Директивы препроцессора имеют вид:

**#include <имя\_файла.h>**

- позволяют вставить в программу заранее подготовленные тексты из включаемых (заголовочных) файлов.
- Включаемые файлы имеют расширение **h**.
- Стандартные включаемые файлы хранятся в каталоге **INCLUDE**.



# Объявление функций

*Прототип функции* – это заголовок функции, заканчивающийся точкой с запятой

По правилам, принятым в языке С, каждый идентификатор, в том числе и имя функции, должен быть объявлен до его использования в программе

При объявлении функции можно сначала объявить только ее заголовок (прототип), а полный текст - тело функции – привести после описания главной функции main.







# Глобальные объявления

Глобальные переменные и константы доступны во всех функциях программы

**В языке C все функции являются глобальными. Нельзя объявлять одну функцию внутри другой.**





# Главная функция main( )

Выполнение программы начинается с вызова функции main(). Поэтому каждая программа на языке C должна содержать функцию main()

Тело функции main состоит из последовательности операторов

Отдельные операторы в теле функции отделяются друг от друга точкой с запятой.





# Определение некоторых понятий

**Исходный текст** (source code) - текст программы на языке программирования.

**Объектный код** (object code) - текст программы на машинном языке, который не может выполняться компьютером. Получается после компиляции исходного текста файла или программы.





**Библиотека** (library) - набор функций, в том числе из стандартных библиотек, предопределенных переменных и констант, которые могут быть использованы в программе и хранятся в откомпилированном виде.

**Компоновщик** (linker) - программа, строящая выполняемый модуль из объектных модулей. Эта программа собирает откомпилированный текст программы и функции из стандартных библиотек языка C в одну выполняемую программу.



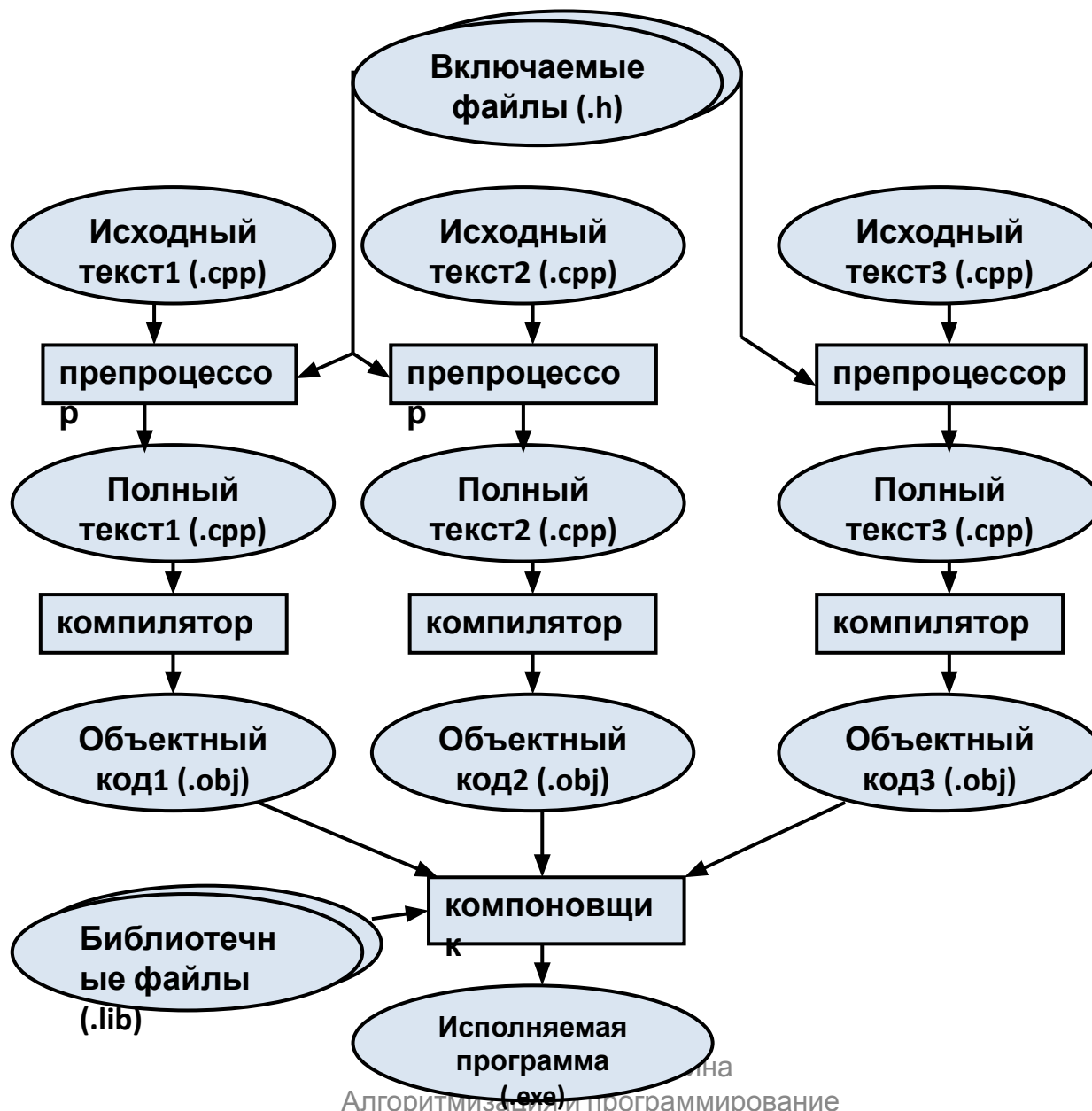


**Время компиляции** (compile time) - период, во время которого происходит компиляция программы. Ошибки, обнаруживаемые во время компиляции (в основном синтаксические), - compile-time error.

**Время выполнения** (run time) - период, во время которого происходит выполнение программы. Ошибки, возникающие в ходе выполнения программы, - run-time error



# Особенности компиляции в С





# Итоги

## Рассмотренные вопросы:

- Язык С
  - история
  - ВОЗМОЖНОСТИ
  - назначение
- Алфавит С – составляющие алфавита
- Лексемы
- Общая структура программ





# Библиографический список

- Подбельский В.В., Фомин С.С. Курс программирования на языке Си: учебник. М.: ДМК Пресс, 2012. – 384 с.
- Павловская Т.А. С/С++. Программирование на языке высокого уровня: учебник для студентов вузов, обучающихся по направлению "Информатика и вычисл. техника" СПб.: Питер, 2005. - 461 с.
- Павловская Т. А., Щупак Ю. А. С++. Объектно-ориентированное программирование. Практикум. Практикум. — СПб.: Питер, 2006. — 265 с: ил.
- Березин Б.И. Начальный курс С и С++ / Б.И. Березин, С.Б. Березин. - М.: ДИАЛОГ-МИФИ, 2001. - 288 с
- Каширин И.Ю., Новичков В.С. От С к С++. Учебное пособие для вузов. – М.: Горячая линия – Телеком, 2012. – 334 с.







Автор:

Саблина Наталья Григорьевна

Ст. преподаватель

каф. РТС УрФУ

