

# Лекция 1. Основные понятия языка

---

**Вводятся базовые для всего дальнейшего изложения понятия: из каких простейших «кирпичиков» состоят все тексты на языке программирования, что понимают под типом данных, какие встроенные типы данных есть в языке C++.**

# Литература

1. Пол И. Объектно-ориентированное программирование с использованием С++. - К: ДиаСофт, 1995
2. Шилдт Г. Теория и практика С++. - СПб.: ВHV, 1996
3. Подбельский В.В. Язык Си++: Учебное пособие. - М.: ФиС, 1996
4. Страуструп Б. Язык программирования С++. - СПб.: БИНОМ, 1999
5. Р. Лафоре. Объектно-ориентированное программирование в С++. — СПб: ПИТЕР, 2003 г. — 928 с.
6. **Мейерс С.** Эффективное использование С++. - М: ДМК, 2000
7. **Элджер Д.** С++: Библиотека программиста. - СПб.: Питер, 1999
8. Голуб А.И. С и С++. Правила программирования. М: БИНОМ, 1996
9. Аммерааль Л. STL для программистов на С++. - М: ДМК, 2000.
10. Дж. Коплиен. Программирование на С++ . — СПб: ПИТЕР, 2005 г. — 479 с.

# Литература

11. **Павловская Т.А.** С/С++. Программирование на языке высокого уровня. — СПб.: Питер, 2001, 2003.
12. Павловская Т.А., Щупак Ю.А. С/С++. Структурное программирование: Практикум. — СПб: ПИТЕР, 2002.
13. **Павловская Т.А., Щупак Ю.А.** С++. Объектно-ориентированное программирование: Практикум. — СПб: ПИТЕР, 2004.
14. Саттер Г. Решение сложных задач на С++. Серия С++ In-Depth, т.4. — Москва: Издательский дом «Вильямс», 2002 г. — 400 с.
15. Александреску А. Современное проектирование на С++. Серия С++ In-Depth, т.3. — Москва: Издательский дом «Вильямс», 2002 г. — 336 с.
16. Кениг Э., Му Б. Эффективное программирование на С++. Серия С++ In-Depth, т.2. — Москва: Издательский дом «Вильямс», 2002 г. — 384 с.

# Литература

17. **Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.** Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: ПИТЕР, 2001 г. — 368 с.
18. Влиссидес Дж. Применение шаблонов проектирования. Дополнительные штрихи. — Москва: Издательский дом «Вильямс», 2003 г. — 144 с.
19. Шаллоуей А., Тротт Д. Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию. — Москва: Издательский дом «Вильямс», 2002 г. — 288 с.
20. Штерн В. Основы С++. Методы программной инженерии. — Москва: ЛОРИ, 2003 г. — 860 с.
21. В. Лаптев. С++. Экспресс-курс. — СПб: БХВ-Петербург, 2004 г. — 512 с.
22. Б. Эккель. Философия С++. — СПб: ПИТЕР, 2004 г. — 572 с.

# Литература

23. К. Бек. Экстремальное программирование. — СПб: ПИТЕР, 2002.
24. К. Бек, М. Фаулер. Экстремальное программирование. Планирование. — СПб: ПИТЕР, 2003 г. — 144 с.
25. К. Ауэр, Р. Миллер. Экстремальное программирование. Постановка процесса. — СПб: ПИТЕР, 2003 г. — 368 с.
26. К. Лоудон. С++. Краткий справочник. — СПб: ПИТЕР, 2004 г. — 220 с.
27. С. Уэллин. Как не надо программировать на С++. — СПб: ПИТЕР, 2004 г. — 240 с.
28. А. Крупник. Изучаем С++. — СПб: ПИТЕР, 2003 г. — 251 с.
29. С. Макконнелл. Совершенный код. — СПб: «Питер», 2005. — 896 с.
30. **Стандарт С++: INTERNATIONAL ISO/IEC STANDARD 14882 1998-09-01, 2003**

Единственный экземпляр!

Минус  
Lex  
Voodoo  
Библиотека

# Как Забыть C++ ДЛЯ "ЧАЙНИКОВ"™

Как перестать  
программировать и  
начать жить

*Здоровое питание с нуля*

*Что такое девушки (вводный курс)*

*Забыть ООП за 5 дней*

*Выходные - для чего они*

*Как выбросить компьютер*

*Кино, театр, зоопарк*

*А сегодня - пятница??*

*Интернет - фтопку!*



# Критерии качества ПО

## Внешние характеристики

- корректность
- практичность
- эффективность
- надежность
- целостность
- адаптируемость
- ...

## Внутренние характеристики

- удобство сопровождения
- тестируемость
- удобочитаемость
- гибкость
- портируемость
- ...

# Основные критерии качества программы

- надежность
- возможность точно планировать производство и сопровождение

Для достижения этих целей программа должна:

- иметь простую структуру
- быть хорошо читаемой
- быть легко модифицируемой



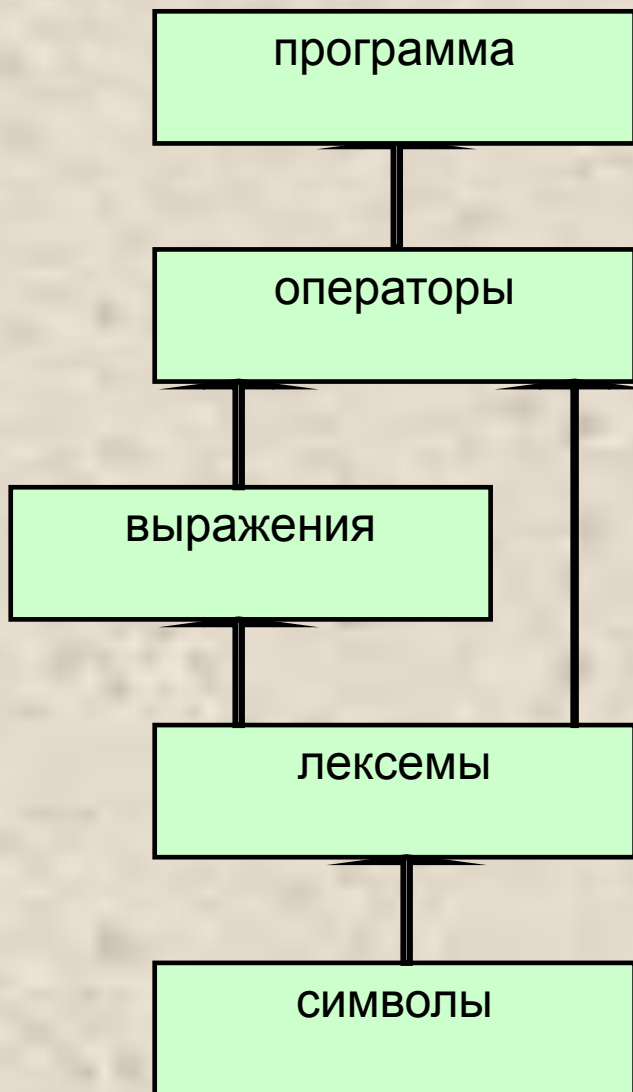
# Парадигмы программирования

- Парадигма — способ организации программы, то есть принцип ее построения. Наиболее распространенными являются процедурная и объектно-ориентированная парадигмы.

Они различаются способом декомпозиции, положенным в основу при создании программы.

- *Процедурная декомпозиция* состоит в том, что задача, реализуемая программой, делится на подзадачи, а они, в свою очередь — на более мелкие этапы, то есть выполняется пошаговая детализация алгоритма решения задачи.
- *Объектно-ориентированная декомпозиция* предполагает разбиение предметной области на объекты и реализацию этих объектов и их взаимосвязей в виде программы.
- Кроме того, существуют *функциональная* и *логическая* парадигмы.

# Состав языка



`a=b; for (int i=0;i<n;++i)`

`a++ - --b/c`

- идентификаторы
- ключевые слова
- константы
- знаки операций
- разделители

`a-z, A-Z, 0-9, " , {,},|,/,%,...`

примеры

# Пример структуры программы

```
директивы препроцессора  
описания  
int main() {  
    операторы главной функции  
}  
int f1() {  
    операторы функции f1  
}  
int f2() {  
    операторы функции f2  
}
```

# Константы

Вид	Примеры
<i>Целые</i> дес.	8 0 199226
Восьм.	01 020 07155
<u>Шестн.</u>	<u>0xA</u> <u>0x1B8</u> <u>0X00FF</u>
<i>Веществ.</i>	5.7 .001 35.
<u>Вещ. с плав. т.</u>	<u>0.2E6</u> <u>.11e-3</u> <u>5E10</u>
<i>Символьные</i>	'A' 'ю' '*' 'db' '\0' <u>'\n'</u>
<u>'\012'</u>	<u>'\x07\x07'</u>
<i>Строковые</i>	"Здесь был Vasia" " \tЗначение r=\0xF5\n"

# Управляющие последовательности

<code>\a</code>	7	Звуковой сигнал
<code>\b</code>	8	Возврат на шаг
<code>\f</code>	C	Перевод страницы (формата)
<code>\n</code>	A	Перевод строки
<code>\r</code>	D	Возврат каретки
<code>\t</code>	9	Горизонтальная табуляция
<code>\v</code>	B	Вертикальная табуляция
<code>\\</code>	5C	Обратная косая черта
<code>\'</code>	27	Апостроф
<code>\"</code>	22	Кавычка
<code>\?</code>	3F	Вопросительный знак
<code>\0ddd</code>		Восьмеричный код символа
<code>\0xdd</code>	dd	Шестнадцатиричный код символа

# Типы данных

Тип данных определяет:

- *внутреннее представление* данных в памяти компьютера => *множество значений*, которые могут принимать величины этого типа;
- *операции и функции*, которые можно применять к величинам этого типа.

Типы в C++ делятся на *основные* (fundamental) и *составные* (compound). Тип может описывать объект, ссылку или функцию.

# Основные (стандартные) типы данных:

int (целый);

intergal

char (символьный);

wchar\_t (расширенный символьный);

bool (логический);

float (вещественный);

double (вещественный с двойной точностью).

Спецификаторы:

short (короткий);

long (длинный);

signed (знаковый);

unsigned (беззнаковый).

+ void

# Составные типы

- *arrays* of objects of a given type;
- *functions*, which have parameters of given types and return void or references or objects of a given type;
- *pointers* to void or objects or functions of a given type;
- *references* to objects or functions of a given type;
- *classes* containing a sequence of objects of various types, a set of types, enumerations and functions for manipulating these objects, and a set of restrictions on the access to these entities;
- *unions*, which are classes capable of containing objects of different types at different times;
- *enumerations*, which comprise a set of named constant values. Each distinct enumeration constitutes a different *enumerated type*;
- *pointers to non-static class members*



# Диапазоны для IBM PC-совместимых

Тип	Диапазон значений	Размер(байт)
bool	true и false	1
<i>signed char</i>	-128 ... 127	1
unsigned char	0 ... 255	1
<i>signed short int</i>	-32 768 ... 32 767	2
unsigned short int	0 ... 65 535	2
<i>signed long int</i>	-2 147 483 648 ... 2 147 483 647	4
unsigned long int	0 ... 4 294 967 295	4
<b>float</b>	3.4e-38 ... 3.4e+38	4
<b>double</b>	1.7e-308 ... 1.7e+308	8
<b>long double</b>	3.4e-4932 ... 3.4e+4932	10

- There are four *signed integer types*: signed char, short int, int, and long int. In this list, each type provides at least as much storage as those preceding it in the list. Plain ints have the natural size suggested by the architecture of the execution environment.
- For each of the signed integer types, there exists a corresponding (but different) *unsigned integer type*: unsigned char, unsigned short int, unsigned int, and unsigned long int, each of which occupies the same amount of storage and has the same alignment requirements.
- There are three *floating point types*: float, double, and long double. The type double provides at least as much precision as float, and the type long double provides at least as much precision as double.

# Диапазоны типов по стандарту

- $\text{sizeof(float)} \leq \text{sizeof(double)} \leq \text{sizeof(long double)}$
- $\text{sizeof(char)} \leq \text{sizeof(short)} \leq \text{sizeof(int)} \leq \text{sizeof(long)}$
- Минимальные и максимальные значения определены в файлах:
  - `<climits>` - целые
  - `<cfloat>` - вещественные

# Явное задание типа констант

- 0X22UL
- 05Lu
- 2E+6L
- 1.82f
- L"Vasia"