

# Модуль 1. Основы программирования на языке высокого уровня

**Рассматриваются критерии качества программ, этапы решения задач на компьютере, базовые понятия: из каких простейших «кирпичиков» состоят все тексты на языке программирования, что понимают под типом данных, какие встроенные типы данных есть в языке C++.**

# Критерии качества ПО

## Внешние характеристики

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

## Внутренние характеристики

- удобство сопровождения
- тестируемость
- удобочитаемость
- гибкость
- портируемость

Основные - надежность, возможность точно планировать производство и сопровождение. Для этой цели 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.

# Этапы разработки программы

- ❖ формулировка и постановка задачи;
- ❖ выбор метода решения задачи;
- ❖ разработка алгоритма решения задачи;
- ❖ написание текста программы;
- ❖ набор текста программы и сохранение в электронном виде;
- ❖ трансляция программы;
- ❖ исправление ошибок;
- ❖ выполнение программы;
- ❖ анализ и документирование полученных результатов.

# Запись алгоритмов

## ❖ **словесная**

*Алгоритм покупок на лимитированную сумму*

1. Сумма покупок равна нулю
2. Если все нужные товары выбраны, перейти к пункту 5, иначе выбрать товар.
3. Если цена товара, добавленная к сумме покупок, не превысит имеющийся лимит, положить товар в корзину, иначе товар оставить на полке и перейти к пункту 5.
4. Перейти к пункту 2.
5. Идти в кассу оплачивать товары.

## ❖ **Графическая**

## ❖ **На алгоритмическом языке**

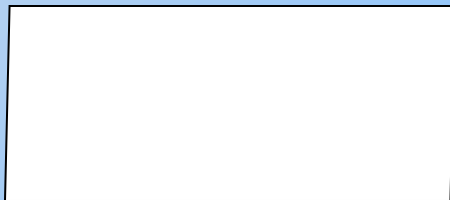
# Основные блоки структурной схемы алгоритма



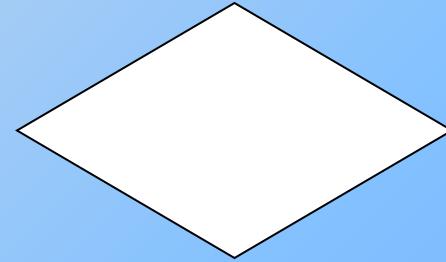
начало, конец программы, вход,  
выход в подпрограммах



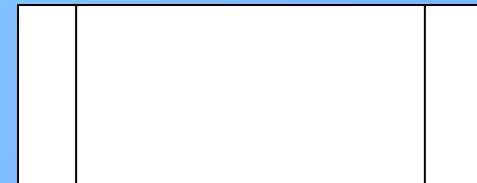
выполнение операций,  
изменяющих обрабатываемые  
данные



ВВОД-ВЫВОД ДАННЫХ

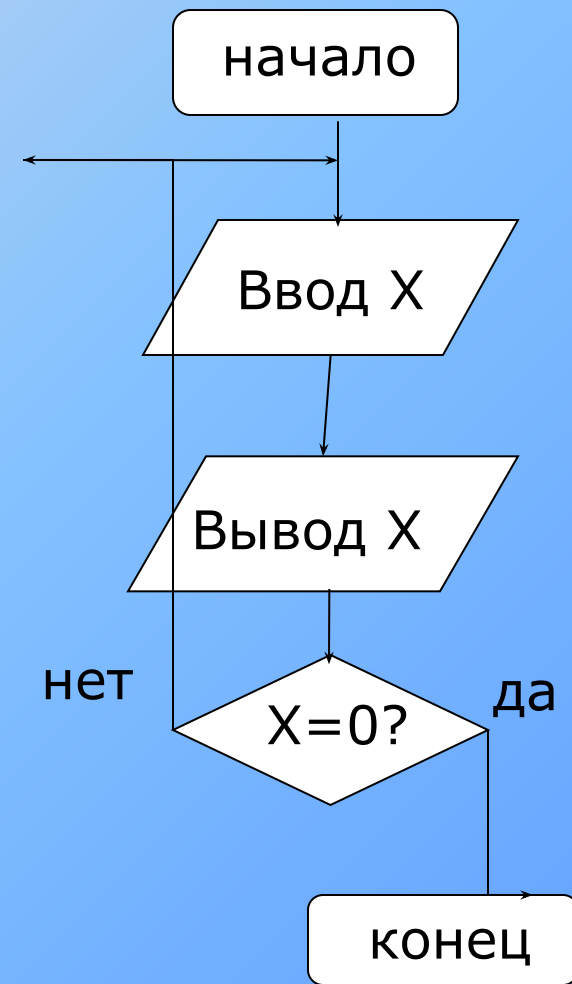
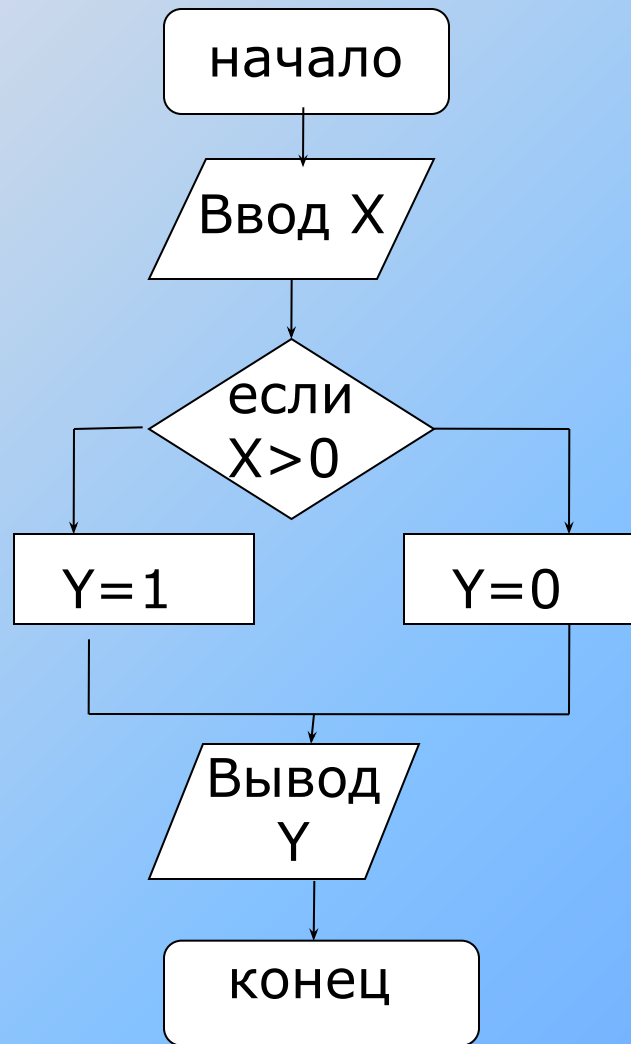
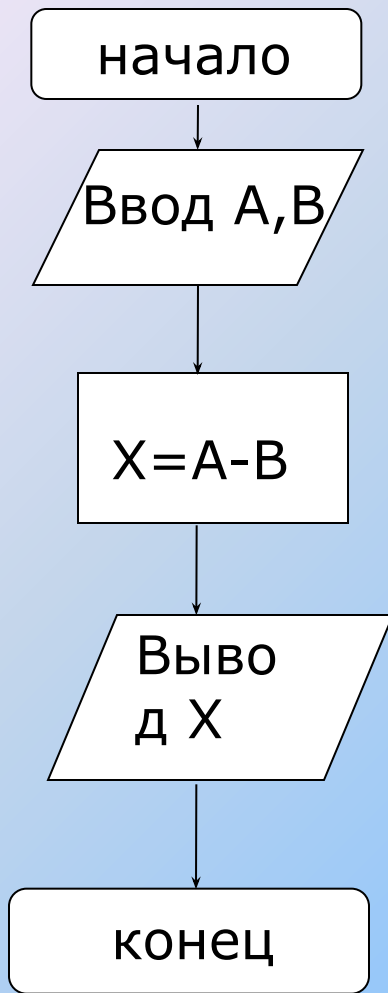


выбор пути обработки  
данных в зависимости от  
условия



обращение к  
подпрограмме

# Примеры блок-схем



# Пример структуры программы

```
директивы препроцессора  
Описания данных и/или функции  
int main() {  
    операторы и описания  
    главной функции  
}
```

## Примеры директив препроцессора:

1) #define - указывает правила замены в тексте.

```
#define ZERO 0
```

Означает , что каждое использование в программе имени ZERO будет заменяться на 0

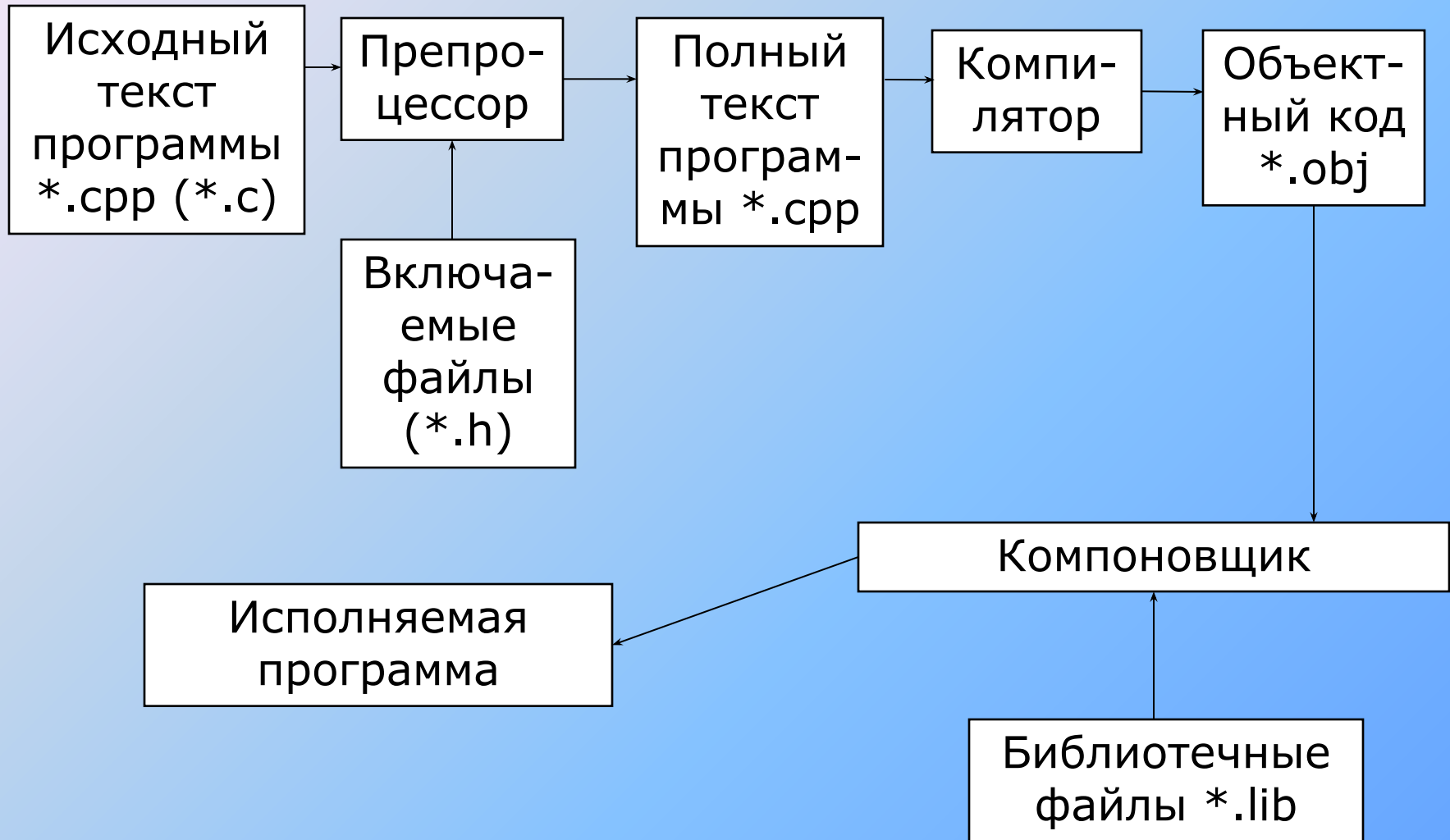
2) Директива #include включает в текст программы содержимое указанного файла. Эта директива имеет две формы:

```
#include " имя_файла "
```

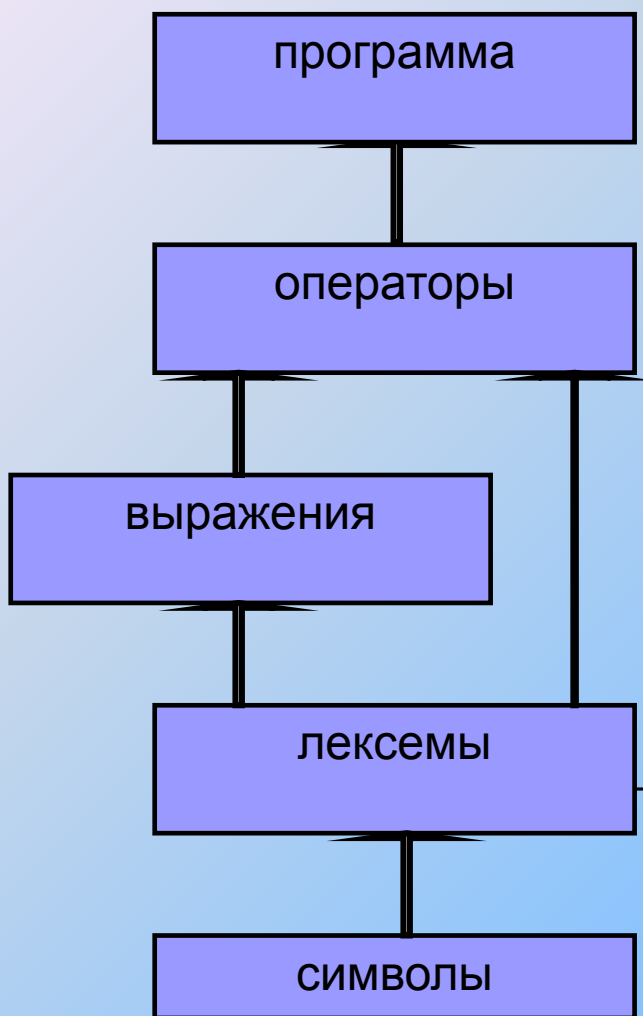
```
#include <имя_файла>
```



# Формирование исполняемой программы



# Состав языка



Примеры:

```
a=b++;
```

```
for (i=0;i<n;++i)
```

```
a+b-t123/c
```

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

a-z, A-Z, 0-9, “, {,},|,/,%,...

Комментарии: /\* это комментарий!!!\*/  
// это тоже комментарий!

# Список ключевых слов

<b>asm</b>	<b>do</b>	<b>if</b>	<b>return</b>	<b>typedef</b>
<b>auto</b>	<b>double</b>	<b>inline</b>	<b>short</b>	<b>typeid</b>
<b>bool</b>	<b>dynamic_cast</b>	<b>int</b>	<b>signed</b>	<b>typename</b>
<b>break</b>	<b>else</b>	<b>long</b>	<b>sizeof</b>	<b>union</b>
<b>case</b>	<b>enum</b>	<b>mutable</b>	<b>static</b>	<b>unsigned</b>
<b>catch</b>	<b>explicit</b>	<b>namespace</b>	<b>static_cast</b>	<b>using</b>
<b>char</b>	<b>export</b>	<b>new</b>	<b>struct</b>	<b>virtual</b>
<b>class</b>	<b>extern</b>	<b>operator</b>	<b>switch</b>	<b>void</b>
<b>const</b>	<b>false</b>	<b>private</b>	<b>template</b>	<b>volatile</b>
<b>const_cast</b>	<b>float</b>	<b>protected</b>	<b>this</b>	<b>wchar_t</b>
<b>continue</b>	<b>for</b>	<b>public</b>	<b>throw</b>	<b>while</b>
<b>default</b>	<b>friend</b>	<b>register</b>	<b>true</b>	

# Константы

<b>Вид</b>	<b>Примеры</b>
<i>Логическая</i>	<i>true (истина) false (ложь)</i>
<i>Целые дес.</i>	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"
Явное указание типа констант: 0X22UL, 05Lu, 2E+6L, 1.82f	

# 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);

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

- ❖ Массивы;
- ❖ Структуры;
- ❖ Объединения;
- ❖ Перечисления;
- ❖ Ссылки;
- ❖ Указатели
- ❖ Классы



# 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

# Диапазоны типов по стандарту

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