

# Синтаксис мови C++. Структура програми

## Тема 2

# Склад мови

## *Звичайна мова*

- Символи
- Слова
  
- Словосполучення
- Речення

## *Мова програмування*

Символи

Елементарні  
конструкції

Вирази

Оператори

# Структура програми

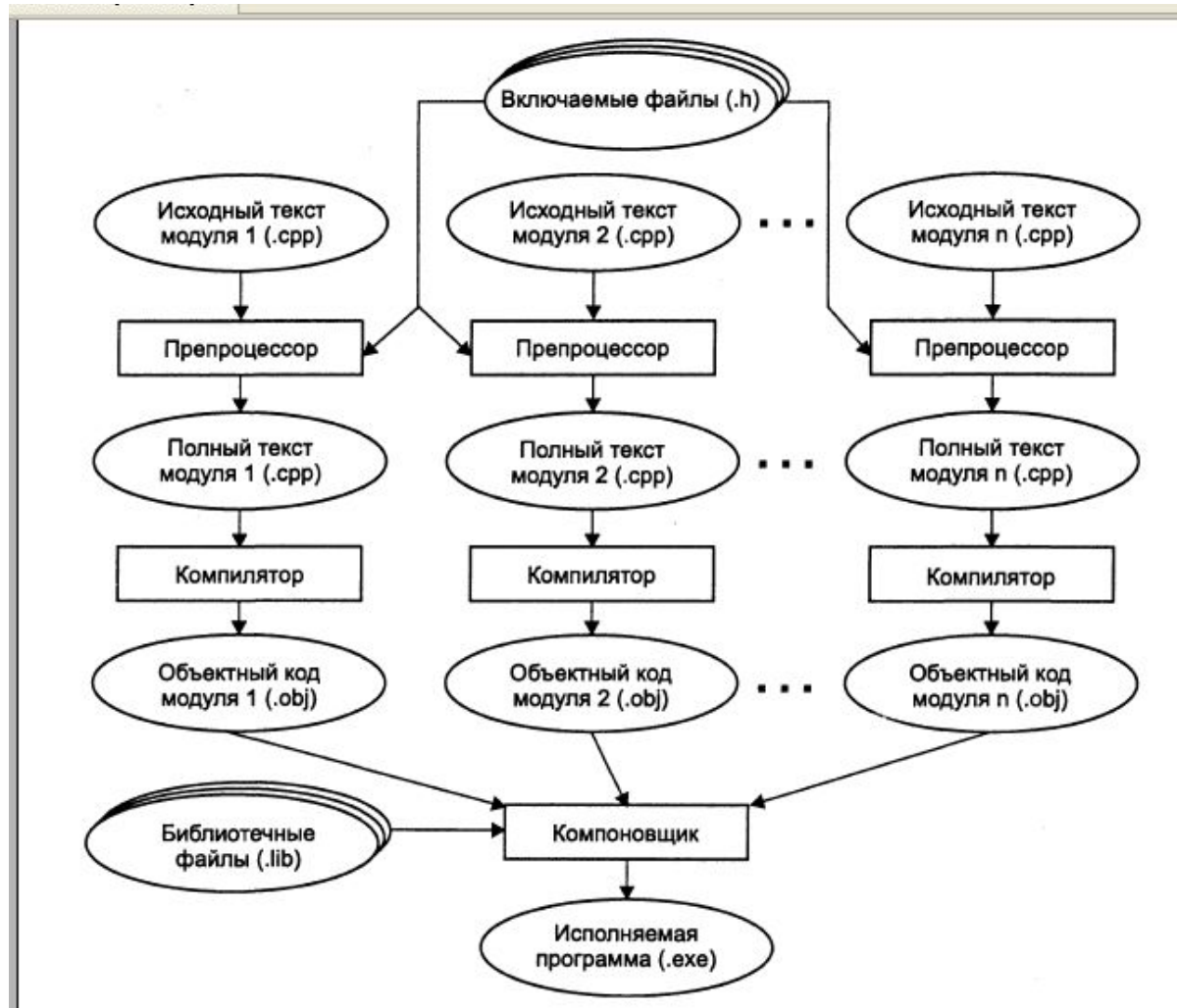


Рис. 1 – Этапы створення програми в середовищі C++

# Структура програми.

## *Директиви препроцесора*

- Починаються зі знаку **#**
- Один рядок – одна директива
- ***Приклад.***

```
#include "myfile"  
#include <iostream.h>  
#include "stdio.h"
```

# Структура програми.

## *Коментарі*

- Однорядкові - //
- Багаторядкові - /\* ..... \*/

- ***Приклад.***

// Найпростіша програма

/\* Найпростіша

програма \*/

# Структура програми. Функції

- Заголовок:
  - Тип результату (тип даних або void)
  - Ім'я
  - Параметри
- Тіло функції
- main – головна функція програми

- Приклад.

```
int main ()  
{ .....  
  return 1; }
```

# Алфавіт мови

- *прописні й рядкові латинські букви й символ підкреслення (  ), що вживається поряд з буквами;*
- *арабські цифри від 0 до 9;*
- *спеціальні символи: + - \* / = < > <= >= = \$ % . , ' ( ) [ ] { } /\* \*/ : ; “ # !*
- *символи форматування - пробіл, символи табуляції, символи переведення рядка й формату.*

# Лексеми

- імена (ідентифікатори);
- ключові слова;
- знаки операцій;
- роздільники;
- КОНСТАНТИ.



# Лексеми. Ключові слова

**Таблиця 1.1.** Список ключевых слов C++

---

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

---

# Лексеми. *Імена*

- Починається з букви або знака підкреслення
- Можна використати заголовні й рядкові латинські букви, цифри й знак підкреслення
- Заголовні й малі літери розрізняються
- ***Приклад.***

abc A12 Name\_Of\_Student Abc

~~12x a-b~~

# Лексеми. *Ключові слова*

- Приклад.

```
int max (int x, int y)
```

```
{  if (x > y)
```

```
    return x;
```

```
else    return y;}
```

# Операції мови C++.

Операція	Краткое описание
<b>Унарные операции</b>	
++	увеличение на 1
--	уменьшение на 1 <sup>1</sup>
sizeof	размер
~	поразрядное отрицание
!	логическое отрицание
-	арифметическое отрицание (унарный минус)
+	унарный плюс
&	взятие адреса
*	адресация
new	выделение памяти
delete	освобождение памяти
(type)	преобразование типа

*Приклад.*

`x++`

`--y`

`x=x+1`

# Операції мови C++.

Бинарные и тернарная операции	
*	умножение
/	деление
%	остаток от деления
+	сложение
-	вычитание
<<	сдвиг влево
>>	сдвиг вправо
<	меньше
<=	меньше или равно
>	больше
>=	больше или равно
==	равно
!=	не равно
&	поразрядная конъюнкция (И)
^	поразрядное исключающее ИЛИ
	поразрядная дизъюнкция (ИЛИ)
&&	логическое И
	логическое ИЛИ

# Операції мови C++.

Операція	Краткое описание
? :	условная операция (тернарная)
=	присваивание
*=	умножение с присваиванием
/=	деление с присваиванием
%=	остаток от деления с присваиванием
+=	сложение с присваиванием
-=	вычитание с присваиванием
<<=	сдвиг влево с присваиванием
>>=	сдвиг вправо с присваиванием
&=	поразрядное И с присваиванием
=	поразрядное ИЛИ с присваиванием
^=	поразрядное исключающее ИЛИ с присваиванием
.	последовательное вычисление

# Операції мови C++.

Операції збільшення на 1 – інкремент

Операції зменшення на 1 – декремент

```
x=x+10;
```

//синтаксис операцій інкремента і декремента

```
++/*ім'я змінної*/; // префіксний інкремент
```

```
/* ім'я змінної */++; // постфіксний інкремент
```

```
--/* ім'я змінної */; // префіксний декремент
```

```
/* ім'я змінної */--; // постфіксний декремент
```

## Приклад 1:

```
int t=1, z; // t*5, потім +1 z=5, t=2
```

```
z=(t++)*5;
```

## Приклад 2:

```
int s=2, f;
```

```
f=(++s)/3; // s =3, f =1
```

# Операції мови C++.

- Операції порівняння

== != < > <= >=

- Логічні операції

&& || !

- Бітові операції

& | ^ ~ << >>

- Умовна операція

операнд1?операнд2:операнд3



# Операції мови C++.

## Приклад умовної (тернарної операції):

```
a > b ? cout << a : cout << b;    // если a > b, то выполняется cout << a, иначе  
                                  выполняется cout << b
```

## Приклад логічних операцій :

```
(x1 < x2)&& (x2 < x3); //логічне И
```

```
(a<b) || (a=b); //логічне ИЛИ
```

```
x1==x2 //рівність
```

```
!= //нерівність
```

# Приклади бітових операцій

## ■ Бітові операції – виконання операцій над окремими бітами числа

- $\sim 1$  (true) дорівнює 0 (false)
- $\sim 0$  (true) дорівнює 1 (false)
- Оператор & визначається так:
- $1$  (true) &  $1$  (true) дорівнює  $1$  (true)
- $1$  (true) &  $0$  (false) дорівнює  $0$  (false)
- $0$  (false) &  $0$  (false) дорівнює  $0$  (false)
- $0$  (false) &  $1$  (true) дорівнює  $0$  (false)
- Для оператора |:
- $1$  (true) |  $1$  (true) дорівнює  $1$  (true)
- $1$  (true) |  $0$  (false) дорівнює  $1$  (true)
- $0$  (false) |  $0$  (false) дорівнює  $0$  (false)
- $0$  (false) |  $1$  (true) дорівнює  $1$  (true)

# Приклади бітових операцій

- *Бітові операції*

$$1 \mid 2 = 3,$$

$$1 \rightarrow (01)_2$$

АБО

$$\begin{array}{r} 2 \rightarrow (10)_2 \\ \hline (11)_2 = 3 \end{array}$$

# Операції мови C++.

- Операції присвоювання

=    +=    -=    \*=    /=    %=  
|=    &=    ^=    <<=    >>=

i=i+1    i+=1;

# Операції мови C++.

Скорочений запис арифметичних операцій:

Замість  $x=x-2$      $x-=2$

Замість  $x=x*2$      $x*=2$

Замість  $x=x/2$      $x/=2$

Замість  $x=x\%2$      $x\%=2$

Приклад:

```
x=9;
```

```
y=4;
```

```
result = x% y;   /* результат = 1 */
```

Використовується для перевірки парності-кратності числа:

```
21 % 6; // результат : 3
```

```
21 % 7; // результат : 0
```

# Вирази

- $X * 12 + Y$   
 $val < 3$   
 $-9$
- $x + y - 12;$   
 $a = b + 1;$

# Порядок обчислення виразів

- ::
- ., ->, [], виклик функції, ++ (постфіксне), -- (постфіксне), typeid
- sizeof, ++ (префіксне), -- (префіксне), ~, !  
-, +, &, \*, new, delete
- .\*, ->\*
- \*, /, %
- +, -
- <<, >>
- < > <= >=
- == !=

# Порядок обчислення виразів

- &
- ^
- |
- &&
- ||
- =, \*=, /=, %=, +=, -=, <<=, >>=, &=, |=, ^=
- ?:
- throw
- ,



# Порядок обчислення виразів

- Щоб змінити послідовність обчислення виражень, можна скористатися круглими дужками
- Операції з однаковим пріоритетом виконуються зліва направо
- Виключення: якщо у виразі кілька операцій присвоювання, то вони виконуються справа наліво

$$x = y = 2$$

# Змінні. Константи

- Оголошення змінної

**тип *ідентифікатор1, ідентифікатор2;***

*Приклад.* int x;

- Ініціалізація змінної

**тип *ідентифікатор=значення;***

*Приклад.* int y=12;

- Опис константи

**const тип ідентифікатор=значення;**

**const int BITS\_IN\_WORD = 32;**

# Константы

Таблица 1.2. Константы в языке C++

Константа	Формат	Примеры
Целая	Десятичный: последовательность десятичных цифр, начинающаяся не с нуля, если это не число нуль  Восьмеричный: нуль, за которым следуют восьмеричные цифры (0,1,2,3,4,5,6,7)  Шестнадцатеричный: 0x или 0X, за которым следуют шестнадцатеричные цифры (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)	8, 0, 199226  01, 020, 07155  0xA, 0x1B8, 0X00FF
Вещественная	Десятичный: [цифры].[цифры] <sup>2</sup>  Экспоненциальный: [цифры][.][цифры]{E e}{+ -}[цифры] <sup>3</sup>	5.7, .001, 35.  0.2E6, .11e-3, 5E10
Символьная	Один или два символа, заключенных в апострофы	'A', 'ю', '*', 'db', '\0', '\n', \012', '\x07\x07'
Строковая	Последовательность символов, заключенная в кавычки	"Здесь был Vasia", "\tЗначение r=\0xF5\n"

# Операція присвоювання

- `x = 2;`  
~~`cond = x < 2;`~~  
~~`3 = 5;`~~
- `z = x = y + 3;`  
`value = 5 + (r = 9 - c);`  
`e = a < d ? a++ : a = d`
- `int x = 0;`  
`x = 3;`  
`x = 4;`  
`x = x + 1;`

# Операція присвоювання

- ```
int x = 0;
++x;
--x;
int y = ++x;
int z = x++;
```

```
int a, b=2, c=1;
a=b+c++;
cout <<"a=" <<a <<" c= " <<c;
/* Результат a=3 c=2.
```
- ```
x = x + 5;
y = y * 3;
z = z - (x + y);
```

```
x += 5;
y *= 3;
z -= x + y;
```