СИ++

ЛЕКСИЧЕСКИЕ ОСНОВЫ ЯЗЫКА СИ++

Структура программы

- ***директивы препроцессора*** – определяют действия по преобразованию программы перед компиляцией, а также включают инструкции, которым компилятор следует во время компиляции;

- ***объявления*** – описания переменных, функций, структур, классов и типов данных;

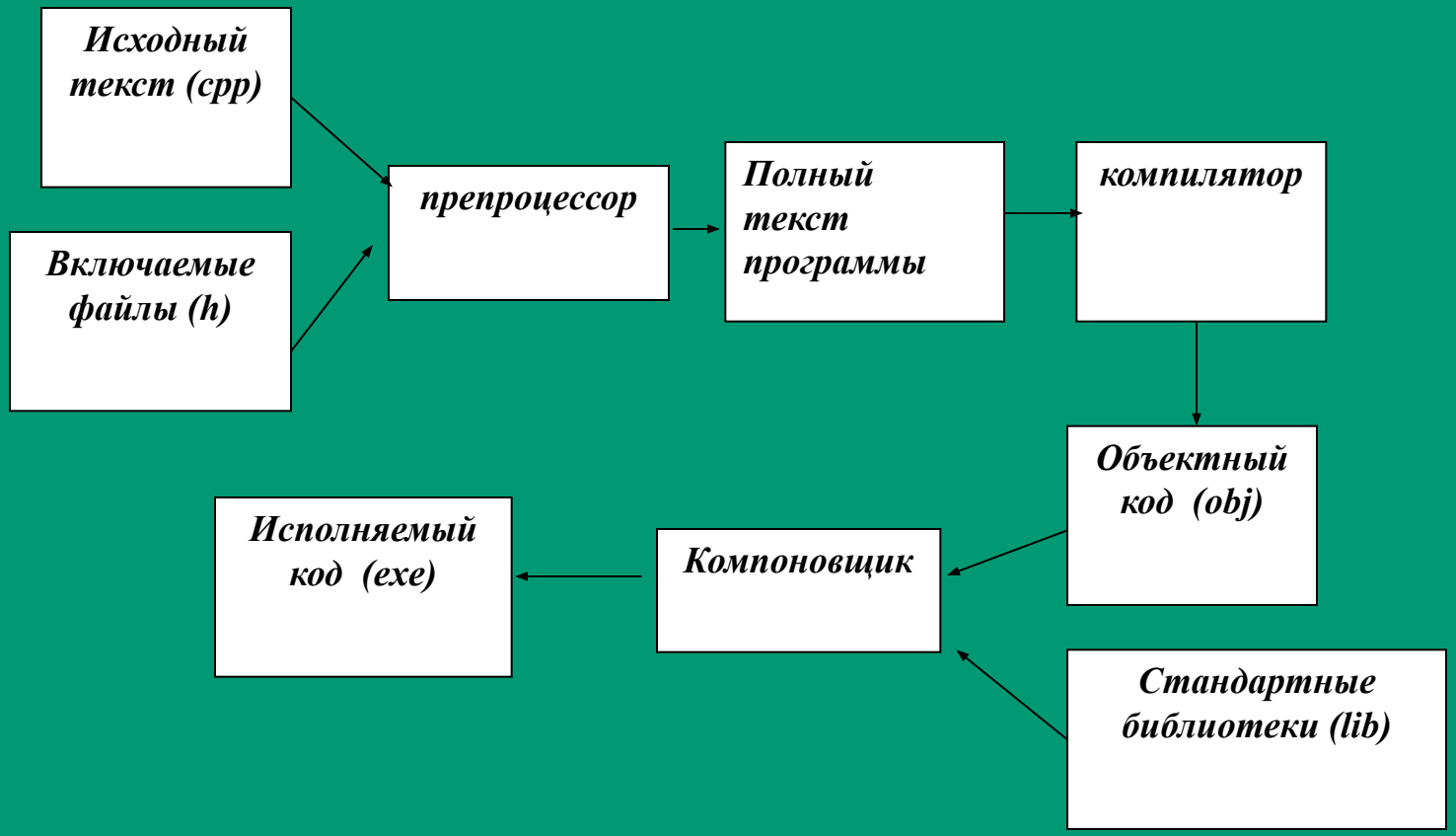
- ***определения*** – тела выполняемых функций проекта.

Структура программы на Си++

```
#директивы препроцессора
. . . . . .
#директивы препроцессора
функция а ( )
    {операторы}
функция в ( )
    {операторы}
void main ( )        //функция, с которой начинается
выполнение                    программы
    {операторы
        описания
        присваивания
        функция
        пустой оператор
            составной
            выбора
            циклов
            перехода}
```

Директивы препроцессора - управляют преобразованием текста программы до ее компиляции. Исходная программа, подготовленная на СИ в виде текстового файла, проходит 3 этапа обработки:

1) препроцессорное преобразование текста ;
2) компиляция;
3) компоновка (редактирование связей или сборка).


Исходный текст (cpp)
Включаемые файлы (h)
препроцессор
Полный текст программы
компилятор
Объектный код (obj)
Стандартные библиотеки (lib)
Компоновщик
Исполняемый код (exe)

1) #define - указывает правила замены в тексте.

```
#define ZERO 0.0
```

2) #include< имя заголовочного файла> -

```
#include <stdio.h>
//препроцессорная директива
```

СИ++

ПРОГРАММА

набор описаний и определений

состоит из набора функций

функция с именем **main**

void main ()

Определение функции задает ее тело, которое представляет собой составной оператор (блок), содержащий другие объявления и операторы. Определение функции также задает имя функции, тип возвращаемого значения и атрибуты ее формальных параметров:

```
int f (int a, int b)
 {
    return (a+b)/2;
 }
```

ОПРЕДЕЛЕНИЯ

Пример:

```
int y = 10 ; //именованная константа
float x ;   //переменная
```

СИ++

Пример программы на Си:

```
#include <stdio.h>  /препроцессорная директива
void main()         //функция
{                   //начало
printf("Hello! "); //печать
}                   //конец
```

Пример программы на Turbo pascal:

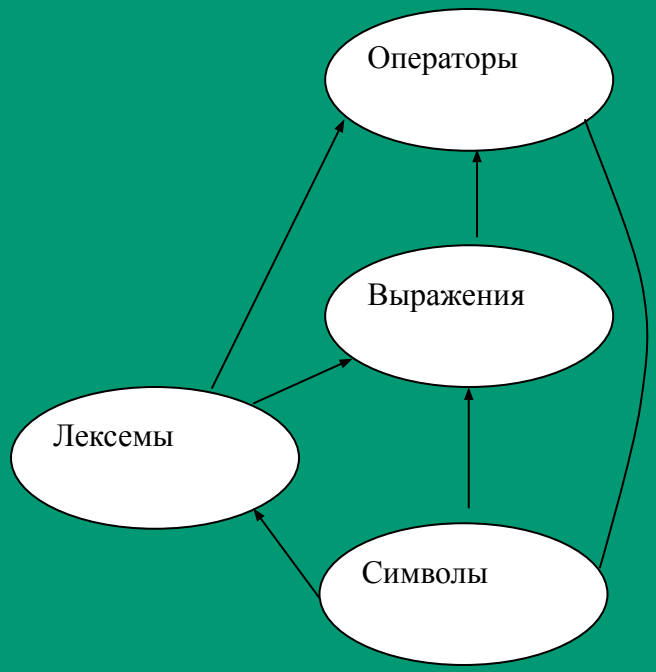
```
program prim;
uses crt;
begin
  Writeln ('Hello!');
end.
```

1. ***Из каких частей состоит программа на С++?***
2. ***Чем определение отличается от объявления?***
3. ***Перечислить этапы создания исполняемой программы на языке С++.***
4. ***Что такое препроцессор?***
5. ***Что такое директива препроцессора?***

Базовые средства языка СИ++

Слова называют лексемами (элементарными конструкциями), словосочетания – выражениями, предложения – операторами.

Лексемы образуются из символов, выражения из лексем и символов, операторы из символов выражений и лексем



СИ++

Элементы алгоритмического языка

Алфавит языка СИ++:

- прописные и строчные латинские буквы и знак подчеркивания;
- арабские цифры от 0 до 9;
- специальные знаки “{},| []()+-/%*.\’:;&?<>=!#^
- пробельные символы (пробел, символ табуляции, символы перехода на новую строку).

Лексемы языка:

- ***Идентификаторы***

Например:

PROG1, prog1 и Prog1 – три различных идентификатора.

- ***Ключевые***

- ***Знаки операций***

- ***Константы***

- ***Разделители – скобки, точка, запятая пробельные символы.***

Константы в Си++

Константы делятся на 5 групп:

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

Целые константы

Десятичная константа (примеры: 8, 0, 192345).
Восьмеричная константа – (примеры: 016).
Шестнадцатеричные константы – (примеры: 0xA, 0X00F).

Вещественные константы.

Вид константы с фиксированной точкой:[цифры].[цифры] (примеры: 5.7, .0001, 41.).

Вид константы с плавающей точкой: [цифры][.][цифры]E|e[+|-][цифры] (примеры:0.5e5, .11e-5, 5E3).

Перечислимые константы

вводятся с помощью ключевого слова ***enum***

Примеры:

enum { one=1, two=2, three=3,four=4};

enum {zero,one,two,three}

enum { ten=10, three=3, four, five, six};

enum {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday} ;

Символьные константы

– это один или два символа, заключенные в апострофы.

- ***Символьные константы, состоящие из одного символа, имеют тип char и занимают в памяти один байт.***
- ***Символьные константы, состоящие из двух символов, имеют тип int и занимают два байта.***

СИ++

Управляющие последовательности

1. ***Для представления символов, не имеющих графического отображения, например:***

\a *– звуковой сигнал,*

\b *– возврат на один шаг,*

\n *– перевод строки,*

\t *– горизонтальная табуляция.*

2. Для представления символов: *\ , ’, ? , ” (\\, \’,\? ,\”).*

3.Для представления символов с помощью шестнадцатеричных или восьмеричных кодов *(\073, \0xF5).*

Строковая константа – это последовательность символов, заключенная в кавычки. Внутри строк также могут использоваться управляющие символы. Например: “\nНовая строка”,

“\n\”Алгоритмические языки программирования высокого уровня \””.

СИ++

Типы данных в Си++

В Си++ определено 6 простых типов данных:

int (целый)
целочисленные
char (символьный)
wchar_t (расширенный символьный)
bool (логический)
с плавающей точкой (число=мантисса х $10^к$)
float(вещественный)
double (вещественный с двойной точностью)

целочисленные

с плавающей точкой
(число=мантисса х $10^к$)

Существует 4 спецификатора типа, уточняющих внутреннее представление и диапазон стандартных типов:

short (короткий)
long (длинный)
signed (знаковый)
unsigned (беззнаковый)

Тип *int*

short int *- занимает 2 байта, имеет диапазон* ***–32768 ..+32767;***

long int *– занимает 4 байта, имеет диапазон* ***–2 147 483 648..+2 147 483 647***

unsigned short int *- занимает 2 байта, имеет диапазон* ***0 ..65536;***

unsigned long int *– занимает 4 байта, имеет диапазон* ***0..+4 294 967 295.***

Тип *char*

В данных типа ***signed char*** *можно хранить значения в диапазоне от* ***–128 до 127****.*

При использовании типа ***unsigned char*** *значения могут находиться в диапазоне от* ***0 до 255***

Тип *wchar_t*

Размер этого типа, как правило, соответствует типу ***short.***

Строковые константы такого типа записываются с префиксом ***L:*** *Например:* ***L“String #1”.***

Тип *bool*

Внутренняя форма представления ***false – 0,*** *любое другое значение интерпретируется как* ***true.***

Типы *с плавающей точкой.*

*Внутреннее представление вещественного числа состоит из 2 частей****: мантиссы и порядка***

Тип *void*

К основным типам также относится тип void Множество значений этого типа – ***пусто.***

СИ++

Переменные

Переменная в СИ++ - *именованная область памяти, в которой хранятся данные определенного типа* : ***int a; float x;***
Общий вид оператора описания:
[класс памяти][const]тип имя [инициализатор];

Класс памяти определяет <u>время жизни и область видимости переменной</u>

<u>Класс памяти</u> может принимать значения: auto, extern, static, register

СИ++

Переменные

Const *– показывает, что эту переменную нельзя изменять* ***(именованная константа).***

Инициализатор можно записывать в двух формах — со знаком равенства или в круглых скобках:

= значение

(значение)

Примеры описания переменных:

```
short int a = 1;
char s, symbol = 'f'; /* инициализация относится только к symbol */
char t = 54;
float c = 0.22, x, sum;
const char C = 'C';
```

Пример

```
int a; //глобальная переменная
void main()
{
int b;//локальная переменная
extern int x;//переменная x определена в другом месте
static int c;//локальная статическая переменная
a=1;//присваивание глобальной переменной
int a;//локальная переменная a
a=2;//присваивание локальной переменной
::a=3;//присваивание глобальной переменной
}
int x=4;//определение и инициализация x
```

Унарные операции в Си++

СИ++

&	получение адреса операнда
*	Обращение по адресу (разыменование)
-	унарный минус, меняет знак арифметического операнда
~	поразрядное инвертирование внутреннего двоичного кода целочисленного операнда (побитовое отрицание)
!	логическое отрицание (НЕ). В качестве логических значений используется 0 - ложь и не 0 - истина, отрицанием 0 будет 1, отрицанием любого ненулевого числа будет 0.

++	Увеличение на единицу:
	префиксная операция - увеличивает операнд до его использования,
	постфиксная операция увеличивает операнд после его использования.
	int m=1,n=2;
	int a=(m++)+n; // a=4,m=2,n=2
	int b=m+(++n);//a=3,m=1,n=3

- -	уменьшение на единицу: префиксная операция - уменьшает операнд до его использования, постфиксная операция уменьшает операнд после его использования.

sizeof	вычисление размера (в байтах) для объекта того типа, который имеет операнд
	имеет две формы
	sizeof выражение
	sizeof (тип)
	Примеры:
	sizeof(float)//4
	sizeof(1.0)//8, т. к. вещественные константы по умолчанию имеют тип double

СИ++

Унарные операции в Си++

~	поразрядное отрицание
!	логическое отрицание
–	арифметическое отрицание (унарный минус)
+	унарный плюс
new	выделение памяти
delete	освобождение памяти
(type)	преобразование типа

Бинарные операции в Си++

*	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

СИ++

Бинарные операции в Си++

? :	условная операция
=	присваивание
*=	умножение с присваиванием
/=	деление с присваиванием
%=	остаток отделения 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

СИ++

Операции в СИ++

Операции присваивания

*=, +=, -=, *= и т.д.*

Формат операции простого присваивания:
операнд1=операнд2

Условная операция.

Выражение1 ? Выражение2 : Выражение3;

Первым вычисляется значение выражения1. Если оно истинно, то вычисляется значение выражения2, которое становится результатом. Если при вычислении выражения1 получится 0, то в качестве результата берется значение выражения3.

Например:

x<0 ? -x : x ; //вычисляется абсолютное значение x.

СИ++

Выражения

Примеры выражений:

(a + 0.12)/6

x && y || !z

(t * sin(x)-1.05e4)/((2 * k + 2) * (2 * k + 3))

СИ++

Приоритеты операций в выражениях

Ранг	Операции
1	( ) [ ] -> .
2	! ~ - ++ -- & * (тип) sizeof тип( )
3	* / % (мультипликативные бинарные)
4	+ - (аддитивные бинарные)
5	<< >> (поразрядного сдвига)
6	< > <= >= (отношения)
7	== != (отношения)
8	& (поразрядная конъюнкция «И»)
9	^ (поразрядное исключающее «ИЛИ»)
10	| (поразрядная дизъюнкция «ИЛИ»)
11	&& (конъюнкция «И»)
12	|| (дизъюнкция «ИЛИ»)
13	?: (условная операция)
14	= *= /= %= -= &= ^= |= <<= >>= (операция присваивания)
15	, (операция запятая)

СИ++

Контрольные вопросы

1. Из каких элементов состоит естественный язык? Что является аналогами этих элементов в C++?
2. Что такое лексема? Привести примеры лексем в языке C++.
3. Что такое идентификатор? Правила записи идентификаторов.
4. Что такое константа? Как константа обрабатывается компилятором?
5. Какие типы констант существуют в C++. Привести примеры констант разных типов.
6. К какому типу относятся константы 192345, 0x56, 0xCB, 016, 0.7865, .0045, ‘c’, “x”, one, “one”, 5 , 5.?
7. Что такое тип данных?
8. Чем отличаются типы данных: float и double, char и wchar_t, int и short int?
9. Чем отличаются типы данных int и unsigned int?
10. Перечислить все типы данных, которые существуют в C++. Сколько места в памяти занимают данные каждого типа?
11. На что влияет количество памяти, выделяемое для данных определенного типа?
12. Что такое переменная? Чем объявление переменой отличается от ее определения? Привести примеры определений и объявлений.
13. Что такое класс памяти? Какие классы памяти существуют в C++? Привести примеры объявлений и определений переменных разных классов памяти.
14. Что такое выражение? Из чего состоит выражение?
15. Что такое операнд?

СИ++

Контрольные вопросы

16. Какие операции можно применять к целочисленным данным? К вещественным данным? К символьным данным?
17. Что такое отношение?
18. В каком случае отношение считается ложным, а в каком – истинным?
19. Какие операции называются унарными? Привести примеры.
20. Какие операции называются бинарными? Привести примеры.
21. Что такое тернарная операция? Привести пример.
22. Какая разница между постфиксной и префиксной операцией инкремента (декремента)?
23. Какие операции присваивания существуют в С++?
24. Привести примеры выражений, содержащих операции присваивания, операции инкремента (декремента), аддитивные и мультипликативные операции. Пояснить, как они будут выполняться.
25. Что такое леводопустимое значение? Привести пример.
26. Чему будет равно значение выражений:
    int z=x/y++; если int x=1, y=2;
    int w=x%++y, если int x=1, y=2;
    int a=++m+n++*sizeof(int); если int m=1, n=2;
    float a=4*m/0.3*n; если float m=1.5; int n=5;
    int ok=int(0.5*y)<short(x)++; если int x=10, y=3;

СИ++

Ввод и вывод данных

В языке Си++ нет встроенных средств ввода и вывода – он осуществляется с помощью функций, типов и объектов, которые находятся в стандартных библиотеках. Существует два основных способа: функции унаследованные из Си и объекты Си++.

Для ввода/вывода данных в стиле Си используются функции, которые описываются в библиотечном файле **stdio.h.**

1) *printf* (форматная строка, список аргументов);

```
printf ( “Значение числа Пи равно  %f\n”, pi);
```

Форматная строка может содержать

1) ***символы печатаемые текстуально;***
2) ***спецификации преобразования;***
3) ***управляющие символы.***

Каждому аргументу соответствует своя спецификация преобразования:
***%d, %i -** десятичное целое число;*
%f -** число с плавающей точкой**;
***%e,%E –** число с плавающей точкой в экспоненциальной форме;*
***%u –** десятичное число в беззнаковой форме;*
***%c -** символ;*
***%s -** строка.*

В форматную строку также могут входить управляющие символы:
***\n** - управляющий символ новая строка;*
***\t** – табуляция;*
***\a** – звуковой сигнал и др.*

Модификаторы формата.

%[-]m[.p]C, где

- - задает выравнивание по левому краю,

m – минимальная ширина поля,

p – количество цифр после запятой для чисел с плавающей точкой и минимальное количество выводимых цифр для целых чисел (если цифр в числе меньше, чем значение p, то выводятся начальные нули),

C- спецификация формата вывода.

Пример

printf("\nСпецификацииформата:\n%10.5d-целое, \n%10.5f - с плавающей точкой, \n%10.5e – в экспоненциальной форме, \n%10s - строка",10,10.0,10.0,"10");

Будет выведено:

Спецификации формата:

…..00010 – целое

..10.00000 – с плавающей точкой

1.00000e+001 - в экспоненциальной форме

……..10 – строка.

СИ++

2) *scanf (форматная строка, список аргументов);*

В качестве аргументов используются адреса переменных.

Например:

scanf(“ %d%f ”, &x,&y);

int i;

scanf(“%d”, &i);

Спецификаторы формата

```
int i;
scanf ("%d", &i); // %d - прочитать целое число
int i;
scanf ("%o", &i); // %o - прочитать восьмеричное число
int i;
scanf ("%x", &i); // %x - прочитать шестнадцатеричное число
float t;
scanf ("%f", &t); // %e(%f) - прочитать вещественное число
char ch;
scanf ("%c", &ch); // %c - прочитать символ
char *str;;
scanf ("%s", str); // %s - прочитать строку
```

При использовании библиотеки классов Си++ подключается библиотечный файл ***iostream.h,*** в котором определены стандартные потоки ввода данных от клавиатуры cin и вывода данных на экран дисплея cout, а также соответствующие операции

1) **<<** *- операция записи данных в поток;*

2) **>>** *- операция чтения данных из потока.*

Например:

```
#include <iostream.h>;
. . . . . .
cout << “\nВведите количество элементов: ”;
 cin >> n;
```

СИ++

Контрольные вопросы

1. Что такое форматная строка? Что содержит форматная строка функции printf? Что содержит форматная строка функции scanf?
2. Что такое спецификация преобразования? Привести примеры спецификаций преобразования для различных типов данных.
3. Что будет выведено функцией printf("\nСреднее арифметическое последовательности чисел равно: %10.5f \nКоличество четных элементов последовательности равно%10.5d ",S/n,k);
4. Как записать вывод результатов из вопроса 3 с помощью операции << ?

Как выполнить ввод переменных x и y, где x типа long int, а y типа double с помощью функции scanf? С помощью операции >> ?