

Лекция 2 Базовые элементы языка C++

О.С. Трушин

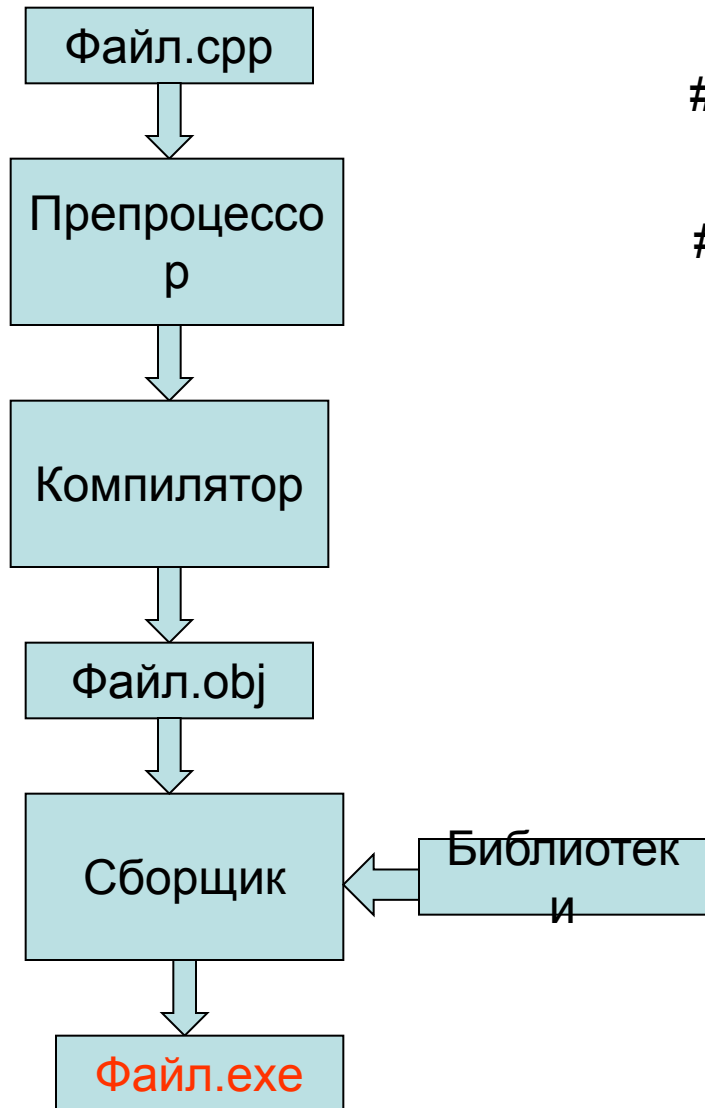
Зав. лаб. ЯФ ФТИАН РАН,

Доцент кафедры нанотехнологии в
электронике

План

- Повторение пройденного.
- Директивы препроцессора.
- Функции. Главная функция `main`.
- Переменные. Константы.
- Арифметические операторы.
Приоритеты.
- Ввод вывод в потоке.
- Массивы.

Директивы препроцессора



`#include` – включение текста в файл

`#define` – назначение значения

`#if`
`#ifdef`
`#ifndef`
`#else`
`#endif`

Условная компиляция

Директива #include

**Назначение – включение копии файла
в данное место программы**

Синтаксис

```
#include < имя файла >  
#include “имя файла”
```

// пример использования

```
#include <iostream>  
#include <math.h>
```

```
void main()  
{
```

```
float x = sin(3.1415);  
cout << “ x=” << x;  
}
```

Директива #define

Назначение – создание символических констант и макросов

Синтаксис

`#define` идентификатор замещающий текст

Все что справа от идентификатора замещающий текст !!!

`#define PI = 3.1415`

Пример использования

```
#define pi_2 1.57  
#define scale 3.0
```

```
void main()  
{
```

```
float y=scale*sin(pi_2);  
cout << " y=" << y;
```

```
}
```

Директива #define создание макросов

Макрос - это символическое имя некоторой операции

Пример

```
#define CIRCLE_AREA(x) ( PI * (x) * (x) )
```

```
void main()  
{  
    area=CIRCLE_AREA(4);  
}
```

Условная компиляция I

Назначение – управление выполнением директив препроцессора и компиляцией

Пример

```
#if !defined(NULL)
```

```
#define NULL 0
```

```
#endif
```

defined(NULL) дает 1
если NULL определена

и

0 если нет

#ifdef эквивалентно #if defined(имя)
#ifndef эквивалентно #if !defined(имя)

Условная компиляция II

Пример выключения кусков кода при компиляции

```
#ifdef LOGGING_ENABLED  
  
    cout << " LOGGING is enabled";  
  
#else  
  
    cout << "" LOGGING is disabled";  
  
#endif
```


Комментарии

**Комментарии - пояснения разработчиков,
включаемые в исходный код.
Они игнорируются компилятором.**

Пример1

```
//-----текст----1 строка-----
```

Пример2

```
/*  
-----многострочный-----  
-----ТЕКСТ-----  
*/
```

Идентификаторы

Идентификаторы – это последовательности символов, используемые в качестве имен переменных, функций, аргументов и т.д.

Идентификаторы могут состоять из букв, цифр, символов подчеркивания.

Они не могут начинаться с цифр!!!

- Компилятор различает буквы верхнего и нижнего регистра
- Нельзя использовать системные слова C++
- Начинающиеся с символа подчеркивания – системные
- Имеются ограничения по длине (зависит от компилятора)

Базовые типы данных

Идентификатор типа	Тип данных
bool	логический
char	символьный
int	целый
float	вещественный
double	вещественный двойной точности
void	значение не возвращается

ФУНКЦИИ

Тип Имя(Аргументы, ...)

```
{  
    Тело  
}
```

Пример1:

```
float fun()  
{  
    return sin(x)*sin(x)+cos(x)*cos(x);  
}
```

Главная функция - main

Пример2:

```
void main()  
{  
    cout << "Hello";  
}
```

Арифметические операторы. Приоритеты

+ сложить
- вычесть
* умножить
/ разделить

$$X=1+2*3;$$

$$Y=(1+2)*3$$

Ввод-вывод в потоке (с клавиатуры и на экран)

// пример

```
#include <iostream>
```

```
void main()
```

```
{
```

```
Int x,y;
```

```
cin >> x;
```

```
y=x+5;
```

```
cout << " x=" << x <<" y="<<y;
```

```
}
```

Массивы

Синтаксис

Тип Имя[размер];

int x[10]; - одномерный массив целых
float y[5][7]; - двух-мерный массив вещественных
char a[15]; - одномерный массив символьных

Максимальная размерность массивов - 7

Массивы нумеруются начиная с нулевого элемента.

Циклы for

Синтаксис

```
for( стартовая точка; конечная точка; изменение счетчика на 1 шаге)
{
Тело цикла
}
```

Пример1 (целый счетчик):

```
for( int i=0; i<20; i++)
{
    cout << " i=" << i;
}
```

Пример2 (вещественный счетчик):

```
float x;
for( x=0; x<20.0; x=x+0.1)
{
    cout << " i=" << i;
}
```


Условный оператор if

Синтаксис

```
if( условное выражение)
{
    тело оператора
}
```

Пример 1

```
if( x>0)
{
    y=sin(x);
    z=y+2;
}
```

Пример 2

```
if( x>0 && y<2)z=x+y;
```

Конструкция if-else

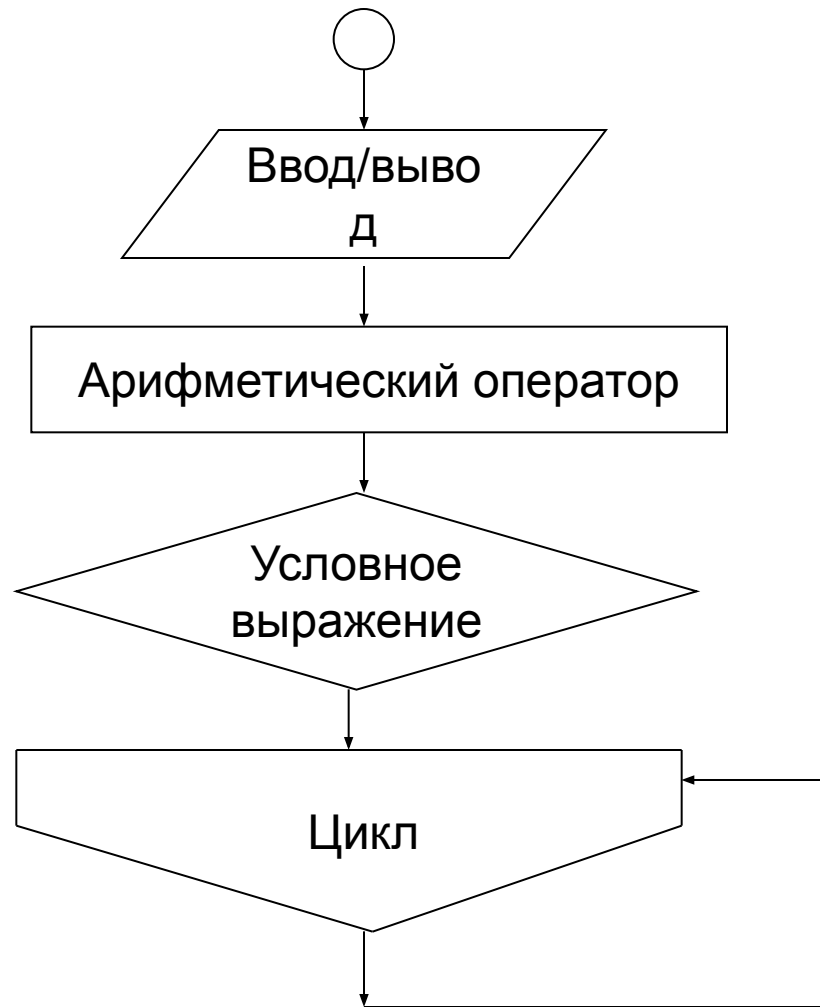
Синтаксис

```
if( условное выражение)
{
    тело1
}
else
{
    тело2
}
```

Пример

```
if( x==0)
{
    y=1;
}
else
{
    y=2;
}
```

Блок схема программы



Пример блок-схемы



ПОТОКОВЫЙ ВВОД/ВЫВОД В ФАЙЛ

```
#include <fstream>
using namespace std;

void main()
{
    ifstream file1("test.in");

    ofstream file2("test.out");

    int x;
    file1 >> x;

    int y= x+1;
    file2 << " y=" << y;

}
```

Локальные и глобальные переменные

Пример1 Локальные

```
#include <iostream>
```

```
float fun( float x)
{
    float s=2.0;
    return s*x;
}
```

```
void main()
{
    float x=0.1;
    int s=5;
    float y=fun(x);
    cout << y;
}
```

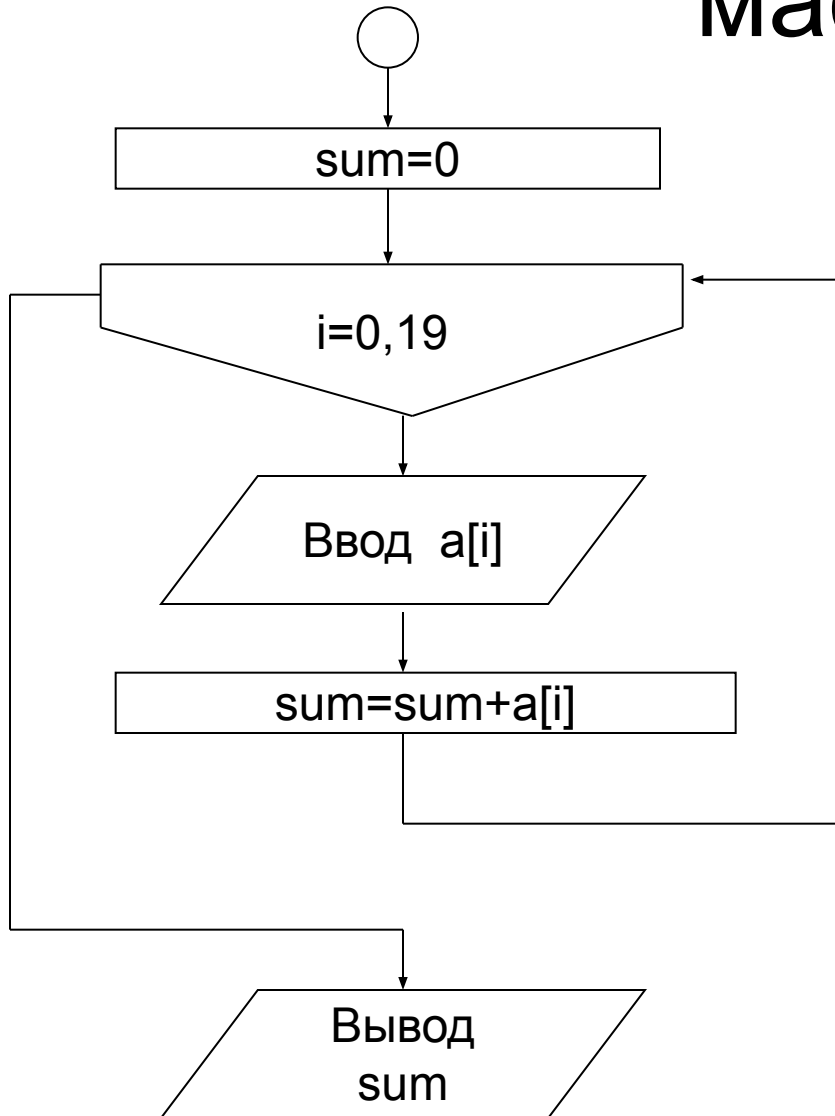
Пример1 Глобальные

```
#include <iostream>
int s;
```

```
float fun( float x)
{
    return s*x;
}
```

```
void main()
{
    float x=0.1;
    int s=5;
    float y=fun(x);
    cout << y;
}
```

Пример 1: Вычисление суммы массива



```
#include <fstream>
#include <iostream>
using namespace std;
```

```
void main()
{
    ifstream file1("test.in");
    int a[20];

    int sum=0;
    for(int i=0; i<20; i++)
    {
        file1 >> a[i];
        sum=sum+a[i];
    }

    cout << " sum=" << sum;

    system("pause");
}
```

Пример2: Поиск минимального

```
#include <fstream>
#include <iostream>
using namespace std;
int minimum(int a[])
{
    int result=1000;
    for(int i=0; i<20; i++)
        if(a[i] < result)result=a[i];
    return result;
}
void main()
{
    ifstream file1("test.in");
    int a[20];

    for(int i=0; i<20; i++)
        file1 >> a[i];

    cout << " mininum=" << minimum(a);

    system("pause");
}
```


Пример3: Вычисление определителя 3*3

```
#include <fstream>
#include <iostream>
using namespace std;
int det(int a[][3])
{
    int result=0;
    result=result + a[1][1]*a[2][2]*a[3][3] + a[2][1]*a[3][2]*a[1][3] + a[1][2]*a[2][3]*a[3][1];
    result=result - a[3][1]*a[2][2]*a[1][3] - a[2][1]*a[1][2]*a[3][3] - a[3][2]*a[2][3]*a[1][1];
    return result;
}
void main()
{
    ifstream file1("test.in");
    int a[3][3];

    for(int i=0; i<3; i++)
        for(int j=0; j<3; j++)
            file1 >> a[i][j];

    cout << " det=" << det(a);

    system("pause");
}
```