

Язык программирования С++ Введение

Лекции по курсу
Языки программирования
семестр 3, ФИИТ
Михалкович С.С.

Язык C++

1972 г. Язык программирования C (Керниган, Ричи)

1980 г. Язык программирования C++ (Бьярн Страуструп)

Ревизии языка C++

C++98

C++11

C++14

C++17

Философия C++

- Мультипарадигменный (процедурный, ОО, обобщенный)
- Переносимый
- Модульный
- Что не используем, за то не платим
- Пользовательские типы по эффективности не должны уступать встроенным
- Максимальное быстродействие в ущерб безопасности (нет контроля выхода за границы массива. Вообще)

Рейтинг ТЮВЕ (2018)

Sep 2018	Sep 2017	Change	Programming Language	Ratings
1	1		Java	17.436%
2	2		C	15.447%
3	5	↑	Python	7.653%
4	3	↓	C++	7.394%
5	8	↑	Visual Basic .NET	5.308%
6	4	↓	C#	3.295%
7	6	↓	PHP	2.775%
8	7	↓	JavaScript	2.131%
9	-	↑↑	SQL	2.062%
10	18	↑↑	Objective-C	1.509%

Простейшая программа 1

```
#include <iostream>

int main()
{
    setlocale(LC_ALL, "Russian");
    std::cout << "Привет мир!\n";
}
```

Подключение заголовочного файла

Escape-
последовательность
перехода на новую строку

Простейшая программа 2

```
#include <iostream>
```

Подключение пространства имен std



```
using namespace std;
```

```
const double Pi = 3.14159265;
```

```
int main()
```

```
{
```

```
    setlocale(LC_ALL, "");
```

```
    double r;
```


```
    cin >> r;
```

```
    double S = Pi * r*r;
```

```
    cout << "Площадь круга = " << S << endl;
```

```
}
```

Манипулятор перехода на следующую строку



Явное указание пространств имен и `using` объявления

```
std::cin >> r;  
std::cout << Pi * r*r << std::endl;
```

или

```
using std::cout;  
cout << S << std::endl;
```

Основные типы

```
int           (4)
bool          (1)
char          (1)
float         (4)
double        (8)
short int     (2) // или short
unsigned int  (4) // или unsigned
unsigned char (1) // это byte
```

Инициализация и неявные преобразования типов

```
double d = 3.14;  
auto s = "C++";
```

```
int i = 3.7; // i == 3 - warning: сужающее преобразование  
char c = 128; // warning: сужающее преобразование  
c = 'z' - 2; // Жуть  
bool b = -1; // true - warning: сужающее преобразование  
int j = true; // j == 1
```

инициализация : преобразование "double" в "int", возможна потеря данных

инициализация : усечение константного значения

инициализация : усечение из "int" в "bool"

Инициализация в стиле C++11

```
double d { 3.14 };  
auto s { "C++" };  
int i { 3.7 };    // error!  
char c { 128 };  // error!  
bool b { -1 };   // error!  
int j { true };  // OK, j == 1
```

Это предпочтительный способ

Основные операции

```
i << 3    // i shl 3
i >> 2    // i shr 2
a += 2;   // a = a+2;
b = a++;  // t = a; a++; b = t;
b = ++a;  // a++; b = a;
a = b+++--c;
7 / 3     // 7 div 3
7 % 3     // 7 mod 3
(i < 0 || i > 2) // или
(i >= 2 && i <= 3) // и
!(i>2)    // не
& | ^ ~ // побитовые and, or, xor, not
min = a<b ? a : b; // условная операция
a = b = c; // множественное присваивание
```

ФУНКЦИИ

```
void f(int i, int j)
{
    cout << i * j << endl;
}
```

```
int abs(int a)
{
    return a > 0 ? a : -a;
}
```

```
auto add(int x, int y)
{
    return x + y;
}
```

Ссылки

Определение. Ссылка – другое имя объекта

```
int a = 5;  
int& ra = a;  
ra = 3;
```

```
void swap(int& a, int& b) // Передача по ссылке  
{  
    auto v = a;  
    a = b;  
    b = v;  
}
```

```
int main()  
{  
    int c, d;  
    swap(c, d);  
}
```

Inline-функции (встраиваемые)

```
inline void sqr(int x)
{
    return x * x;
}
```

```
int main()
{
    int x = 5;
    auto a = sqr(x);
}
```

```
int main()
{
    int x = 5;
    auto a = x * x;
}
```

inline – рекомендация компилятору
сделать функцию встраиваемой

Генерация случайных чисел

```
#include <iostream>
#include <ctime>
#include <random>

int main()
{
    default_random_engine engine(time(0));
    uniform_int_distribution<> dist {1, 10};
    for (int i = 0; i < 100; i++)
        cout << dist(engine) << " ";
}
```

Структуры

```
struct complex
{
    double re, im;
};

struct Person
{
    string name;
    int age;
};

int main()
{
    complex c1 {0, 1}, c2 {1, 2};
    c1.re++;
    c1.im = 3;
    c2 = c1;

    Person p {"Иванов", 23};
    p = {"Попова", 21};
}
```

Кортежи

```
#include <tuple>

using namespace std;

int main()
{
    tuple<int, char, string> t {4, 'z', "C++"};
    auto x = make_tuple(4, 'z', "C++");
    auto s1 = get<2>(x);
    int i; char c; string s;
    tie(i, c, s) = t; // распаковка
}
```


Перечисления

```
int main()
{
    enum MyType { A, B, C }; // A=0 B=1 C=2
    enum YourType { D = 2, E, F = 0 }; // E=3
    MyType m = A;
}
```

Перечисления в стиле C++11

```
int main()
{
    enum class Color { Red, Green, Blue };
    Color c = Color::Blue;
}
```

Векторы

```
#include <vector>

using namespace std;

int main()
{
    vector<int> mm(5);
    vector<int> a {1, 2, 3, 7};

    // Цикл for по вектору
    for (int i=0; i < a.size(); i++)
        cout << a[i] << " ";

    // Цикл foreach по вектору
    for (auto x : a)
        cout << x << " ";
    for (int& x : a)
        x++;
}
```

Примеры использования векторов

```
bool is_symm(const vector<int>& a)
{
    int i = 0, j = a.size() - 1;
    while (i < j)
        if (a[i++] != a[j--])
            return false;
    return true;
}
```

```
bool redouble(vector<int>& a)
{
    for (int& x: a)
        x += 2;
}
```

```
int main()
{
    vector<int> a{ 1,2,3,4,5,5,4,3,2,1 };
    cout << is_symm(a) << endl;
    redouble(a);
}
```

Строки

```
#include <iostream>
#include <string>
```

```
using namespace std;
```

```
int main()
{
    string s = "C++", s1;
    s1 = s;
    s[0] = s[s.size() - 1];
    s = s + "17";
    cout << s << endl;
    cin >> s;           // ввод до первого пробела
    getline(cin, s1);  // ввод до конца строки
}
```

Преобразование строка <--> число

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s2 = to_string(123);
    try
    {
        int i = stoi("abc");
    }
    catch (const std::invalid_argument& ia)
    {
        cout << ia.what();
    }
}
```

Передача строки как параметра

```
#include <iostream>
#include <string>

using namespace std;

bool starts_with(const string& s, char c)
{
    return s[0] == c;
}

int main()
{
    string s = "C++";
    cout << starts_with(s, 'C');
}
```

initializer_list (C++11)

```
#include <iostream>
#include <initializer_list>

using namespace std;

void print(initializer_list<int> il)
{
    for (int x : il)
        cout << x << " ";
}

int main()
{
    print({2, 4, 6, 8});
}
```

Определение типов

```
using byte = unsigned char;  
using IntVec = vector<int>;
```

Определение типов (старый стиль)

```
typedef unsigned char byte;  
typedef vector<int> IntVec;
```


Q & A