

ПРАКТИЧЕСКИЙ КУРС ДЛЯ НАЧИНАЮЩИХ

Боевой курс C++

Концентрированный курс по языку программирования C++: основные конструкции языка, работа с указателями и машинной памятью, объектно-ориентированное программирование, стандартная библиотека...

Переменные, типы данных и операции C++

Шамин Роман Вячеславович

доктор физико-математических наук,

директор Института перспективных технологий и индустриального программирования

МИРЭА – Российского технологического университета

Переменная – это основное понятия в программировании. Переменной называют именованную область в памяти, к которой можно обращаться (присваивать и считывать значения) по заданному имени.

Переменная в языке C++ имеет уникальное название (имя переменной), тип данных, которые хранятся в переменной и значение этой переменной.

Язык C++ относится к строго типизированным языкам программирования, т.е. перед использованием переменной ее нужно явно объявить. При этом указывается:

- имя переменной;
- тип данных;
- начальное значение (необязательно).

Вот примеры объявления переменных в C++

```
int a = 648;  
double x;  
char c = 'a';
```

Здесь мы переменным a и c присвоили начальное значение, а переменной x нет. Однотипные переменные можно объявлять одновременно:

```
int a, b = -4, c;
```

Заметим, что язык C++ является регистрозависимым, поэтому ABC, Abc, abc – это будут различные переменные:

```
double ABC, Abc, abc;
```

Имя переменной может включать в себя латинские буквы, цифры и знак подчеркивания. При этом первым символом должна быть буква или знак подчеркивания.



Каждая переменная в C++ должна иметь определенный тип данных. Основные типы данных:

`int` – целочисленные значения

`short` – небольшие целочисленные значения

`long` – большие целочисленные значения

`long long` – особенно большие значения

Перед каждым из этих типов может быть слово `unsigned`, что будет означать «число без знака». В этом случае диапазон переменной увеличивается в двое, но зато числа будут неотрицательные.

К сожалению, в языке C++ емкость переменных (минимальный и максимальный диапазоны) зависят от разрядности процессора, поэтому могут меняться. Если у Вас 64-х битный процессор, то `int` занимает 4 байта и, соответственно, переменная этого типа может хранить число в от -2 147 483 648 до 2 147 483 647. Гарантируется, что емкости типов данных: `short < int < long < long long`.

Рассмотрим теперь символьные типы данных.

`char` – один символ (один байт).

`unsigned char` – символ без знака.

`wchar_t` – для хранения «широких символов», использующих Unicode.

`char16_t`, `char32_t` – символы на 2 и 4 байта для Unicode.

Тип данных `char` часто используют для хранения чисел от -128 до 127, а `unsigned char` от 0 до 255.

Использование `wchar_t`, `char16_t` и `char32_t` не так просто, как кажется.



Для работы с вещественными числами (дробными) используются следующие типы данных:

float – небольшие дробные значения;

double – дробные значения;

long double – большие дробные значения.

Тип данных float занимает 4 байта, double – 8, а long double – 16 байт. Как правило, для расчетов вполне хватает типа данных double. Следует отметить, что вещественные числа в компьютере и в C++ представляются приближенно. Это нужно иметь в виду при сравнении чисел этого типа данных.

Еще одним важным типом данных в C++ является логический тип данных:

bool – принимает только два значения true или false.

Наконец, в C++ есть еще один особый тип данных:

void – пустой тип данных

Нельзя создать переменную типа void, но этот тип используется для того, чтобы указать, что функция не возвращает значения, а также для объявления нетипизированных указателей.

Для работы со строками в C++ как и в C нет типа данных. Но если в C строки представляют собой массивы типа char, в которых конец строки – это нулевой символ. Для работы со строками в стиле C приходится вызывать функции даже для сравнения двух строк и т.д. Стандартная библиотека C++ включает в себя класс string для работы со строками «по человечески».

Если перед типом данных написать слово const, то вместо переменной мы получим константу, значение которой уже нельзя будет изменить:

```
const int max = 100;
```

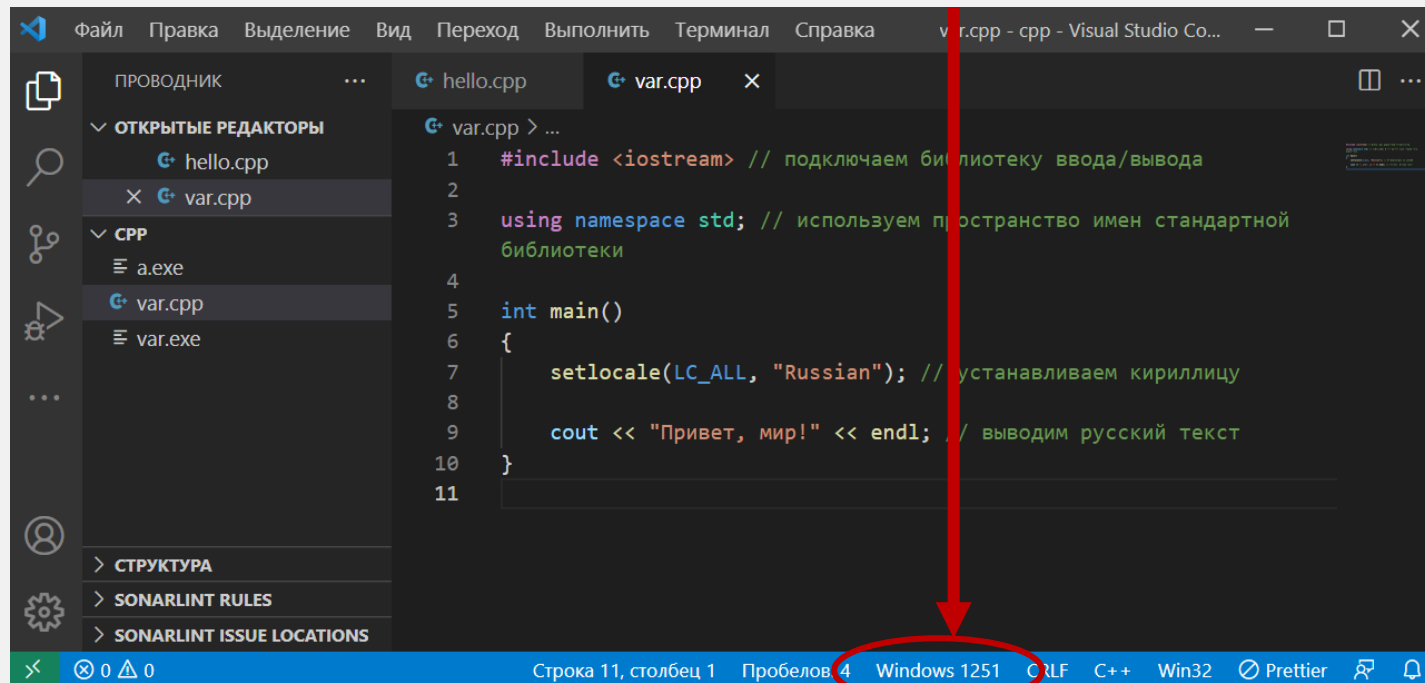


Подключаем кириллицу

Напишем программу для работы с переменными. Но в самом начале давайте установим возможность использования русских букв в C++. Для этого используем функцию `setlocale()` как в следующем коде:

```
#include <iostream> // подключаем библиотеку ввода/вывода
using namespace std; // используем пространство имен стандартной библиотеки
int main()
{
    setlocale(LC_ALL, "Russian"); // устанавливаем кириллицу
    cout << "Привет, мир!" << endl; // выводим русский текст
}
```

Убедитесь, что текст Вашей программы сохранен в кодировке Windows-1251.



Объявим переменные различного типа:

```
int a = 648; // объявляем и инициализируем переменную целого типа
double x = 3.14; // вещественный тип
double y = 1E-6; // экспоненциальная запись y = 0.000001;
char c = 'w'; // символ
char b = 75; // символ с кодом 75
char t = '\t'; // символ табуляции - это один символ
bool Flag = true; // логическая переменная
Flag = 0; // так тоже можно, 0 - false, 1 - true
```

Старайтесь не использовать числовые значения для логических переменных, а пользуйтесь true и false. Для ранее объявленной переменной можно менять значение в любой момент.

Для константы изменить значение будет уже нельзя:

```
const int max = 100; // объявляем константу
max = 120; // ошибка! код не будет компилироваться
```

В C++ можно числа записывать в различных системах исчисления:

```
int h = 0xFF; // шестнадцатеричное число h = 15
int g = 0b10101; // двоичное число g = 21
int o = 010; // восьмеричное число o = 8
```

Будьте внимательны, не начинайте число с нуля, если не хотите записать число в восьмеричной системе исчисления (кто-нибудь помнит, когда это было нужно?☺).



С числовыми переменными можно выполнять арифметические операции $+$, $-$, $*$, $/$, которые мы не будем подробно рассматривать в виду их очевидности. Рассмотрим некоторые особенности.

При делении целых чисел деление происходит с отбрасыванием дробной части:

```
int n = 5; // объявляем целочисленные переменные
int m = 2;
double k = n / m; // объявляем вещественную переменную и присваиваем результат деления
cout << n << " / " << m << " = " << k << endl; // выводим на печать
```

В результате будет напечатана строка: "5 / 2 = 2". При том, что переменная k имеет вещественный тип. Чтобы деление было с дробной частью нужно, чтобы числитель или знаменатель были вещественными:

```
cout << 5 / 2 << endl; // будет напечатано 2
cout << 5 / 2.0 << endl; // будет напечатано 2.5
```

Чтобы получить остаток от деления нужно воспользоваться операцией $\%$:

```
cout << 100 % 7 << endl; // будет напечатано 2
```

Операции $+=$, $-=$, $*=$, $/=$ означают следующее:

```
a += 5; // эквивалентно a = a + 5
```

остальные аналогично.

Для инкрементирования и декрементирования используем следующие операции

```
a++; // эквивалентно a = a + 1
a--; // эквивалентно a = a - 1
```



Числовые переменные можно сравнивать с помощью операций

== – равно; != – не равно; < – меньше; > – больше; <= – меньше или равно; >= – больше или равно.

Обратите внимание, что операция = – это операция присвоения, но не сравнения.

В C++ есть стандартные логические операции:

! – логическое «НЕ»;

&& – логическое «И»;

|| – логическое «ИЛИ».

Поразрядные операции:

& – побитовая конъюнкция; | – побитовая дизъюнкция; ^ – побитовое исключающее или; ~ – побитовая инверсия.

Еще в C++ иногда используют операции сдвига (не используйте их без особой надобности!):

<< побитовый сдвиг влево;

>> побитовый сдвиг вправо.

Например, побитовый сдвиг влево на k приведет к тому, что число будет умножено на 2^k :

```
a = 3; // a = 3 = 0b11
a = a << 2; // сдвигаем на 2, в результате a = 0b1100 = 12
cout << a << endl; // будет напечатано 12
```

