

Технологии программирования

Доц. каф. «Медиаменеджмента и медиапроизводства» Евич
Л.Н.

Лекция 2. Объявление переменных и типы данных в C++.

В Си++ переменные можно описывать в любой части программы, но перед непосредственным их использованием.

Основные типы данных языка C++

Тип данных	Размер, байт	Диапазон значений
bool	1	true, false
unsigned short int	2	0 — 65 535
short int	2	-32 768 — 32 767
int	4	-2 147 483 648 — 2 147 483 647
unsigned int	4	0 — 4 294 967 295
float	4	1.2e-38 — 3.4e38
double	8	2.2e-308 — 1.8e308
char	1	256 значений символов



Объявление переменных в C++

```
int a;    // объявление переменной a целого типа.  
float b,c; // объявление переменных b,c типа данных с плавающей  
запятой.
```

При объявлении переменной можно сразу ее инициализировать.

```
float b=3.2; // инициализация переменной b типа float.  
char ch = 'a'; // инициализация переменной ch типа char.  
bool f = true; // инициализация логической переменной f.
```

Имя переменной (идентификатор) может состоять только из латинских символов строчных и прописных, цифр от 0 до 9 и знака подчёркивания. При этом имя переменной не может начинаться с цифры.



Переполнение данных

```
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    unsigned short int a=65535;
    cout<<"a="<<a<<endl;
    a=a+2;
    cout<<"a+2="<<a<<endl;
    getchar();
}
```

Результат выполнения данной программы:

a=65535

a+2=1



Константные переменные, преобразование типов данных в C++.

Квалификатор `const`.

Если переменная объявлена с ключевым словом `const`, значит, она не должна меняться. После определения константной переменной нельзя изменить ее значение или передать ее в качестве аргумента функции, которая не гарантирует ее неизменности.

```
const float pi = 3.14;    // Целая константа
pi = 3.1;                // Нельзя, значение не должно меняться
const float pi;          // Нельзя, отсутствует начальное значение
```



Лекция 3. Константные переменные, преобразование типов данных в C++.

Директива `#define`

```
#define MY_PI 3.14 // строка не заканчивается точкой с запятой.
```

Отличия `const` и директивы `#define`: директива `#define` создает макроконстанту, и ее действие распространяется на весь файл, а с помощью `const` создается константная переменная, которая может иметь ограниченную область видимости.



Лекция 3. Константные переменные, преобразование типов данных в C++.

Квалификатор `volatile`

```
volatile int my_time;
```

Этот квалификатор используется в тех случаях, когда надо объявить переменную, которая может быть изменена в результате выполнения внешних действий не контролируемых программой, например, системными устройствами (таймером).

Этот квалификатор может быть также использован для описания объектов данных, используемых разными процессами в многозадачной среде.



Операторы в C++.

Побитовые операторы

Побитовое ИЛИ (|).

Оператор | записывает в бит результата единицу только в том случае, если хотя бы один из сравниваемых бит равен 1.

Этот оператор часто используют для установки отдельных битов числа.

Бит 0	Бит 1	Результат
0	0	0
0	1	1
1	0	1
1	1	1

```
01001010 | 10001101 // результат  
11001111
```



Операторы в C++.

Побитовые операторы

Исключающее ИЛИ (^).

Оператор \wedge записывает в бит результата единицу в том случае, если сравниваемые биты отличаются друг от друга. Этот оператор часто применяется при выводе изображений на экран, когда необходимо накладывать друг на друга несколько графических слоев.

Бит 0	Бит 1	Результат
0	0	0
0	1	1
1	0	1
1	1	0



Операторы в C++.

Побитовые операторы

Операторы сдвига.

сдвиг влево (<<) и сдвиг вправо (>>).

Первый сдвигает битовое представление целочисленной переменной, указанной слева от оператора, влево на количество битов, указанное справа от оператора. При этом освобождающиеся младшие биты заполняются нулями, а соответствующее количество старших битов теряется.



Операторы в C++.

Побитовые операторы

Пример

```
unsigned int value = 52 ; // младший байт: 0011 0100
value <<= 1 ;           // младший байт: 0110 1000
cout << value;         // будет выведено 104
```



Операторы в C++.

Побитовые операторы

Пример

```
unsigned int value = 15 ; // младший байт: 0000 1111
value >>= 1 ;           // младший байт: 0000 0111
cout << value;         // будет выведено 7
```



Операторы в C++.

Инкрементирование и декрементирование.

Инкрементирование (++) и декрементирование (--) являются соответственно операциями увеличения или уменьшения значения переменной на 1.

Пример.

```
a=a+1;
```

```
b=b-1;
```

Соответствуют

```
a++; // или ++a
```

```
b--; // или --b
```



Операторы в C++.

Инкрементирование и декрементирование.

В постфиксной записи `a++` сначала возвращается значение переменной, после чего она увеличивается на 1, в префиксной записи наоборот.

Пример

```
int i=3; j, k=0;
```

```
k=++i;      //i=4, k=4
```

```
k=i++;     // i=4, k=3
```

```
k=--i;     //i=2, k=2
```

```
k=i--;     //i=2, k=3
```

```
i=j=k--;   //i=0, j=0, k=-1
```



Основные математические операторы в C++

- + — сложение;
- — вычитание;
- * — умножение;
- / — деление или целочисленное деление (для целочисленных переменных);
- % — деление по модулю (для целочисленных переменных).



Основные математические операторы в C++

```
// Вычисление суммы и разности двух чисел
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    int a, b=6;
    cout<<"a=";
    cin>>a;
    getchar();
    cout<<a<<"+"<<b<<"="<<a+b<<endl;
    cout<<a<<"-"<<b<<"="<<a-b;
    getchar();
}
```



Операторы в C++.

Комбинированные операторы присваивания

Примеры.

```
a=a+b;  
c=c/b;  
b=b*4;  
a=a%c;
```

```
a+=b;  
c/=b;  
b*=4;  
a%=c;
```



Основные математические функции в C++.

Функция	Описание работы функции
double sqrt (double x)	Вычисляет и возвращает корень из положительного числа, принимая его в качестве аргумента x
double pow (double x , double y)	Возводит число, принимаемое в качестве аргумента x , в степень, принимаемую в качестве аргумента y
double fabs (double x)	Вычисляет абсолютное значение числа x (иными словами, его модуль)
double fmod (double x , double y)	Вычисляет остаток от деления x на y .
double ceil (double x)	Вычисляет наименьшее целое, значение которого не будет меньше, чем x . Например, функция ceil (4.68) вернет значение 5.00 .
double floor (double x)	В отличие от предыдущей функции эта функция вычисляет наибольшее целое, по значению не превосходящее x . Например, функция floor (4.68) вернет значение 4.00 .
double modf (double value , double *ptr)	Разбивает значение аргумента value на целую и дробные части. Целую часть функция сохраняет в объекте на который указывает указатель *ptr , а дробную возвращает.

Основные математические функции в C++.

Функция	Описание работы функции
double cos (double x)	Вычисляет косинус аргумента x , который задается в радианах
double sin (double x)	Вычисляет синус аргумента x , который задается в радианах
double tan (double x)	Вычисляет тангенс аргумента x , который задается в радианах
double acos (double x)	Вычисляет главное значение арккосинуса x . Аргумент x должен быть из интервала [-1;+1]. Функция возвращает значение в радианах из интервала [0; π]
double asin (double x)	Вычисляет главное значение арксинуса x . Аргумент x должен быть из интервала [-1;+1]. Функция возвращает значение в радианах из интервала [$-\pi/2$; $+\pi/2$]
double atan (double x)	Вычисляет главное значение арктангенса x .
double exp (double x)	Вычисляет значение показательной функции аргумента x .
double log (double x)	Вычисляет натуральный логарифм аргумента x .
double log10 (double x)	Вычисляет десятичный логарифм аргумента x .



Преобразование типов данных в C++.

При использовании в операции переменных различных типов данных Си может автоматически преобразовывать данные из одного типа в другой.

Автоматическое преобразование типов данных приводится к типу данных имеющего более высокий приоритет. Типы данных в порядке уменьшения приоритета:

double

float

long

int

short



Преобразование типов данных в C++.

Пример 2

```
int ival;  
float fval1=3.0, fval2=10.0;  
ival=fval1+fval1/fval2;
```

Результат выполнения операции $fval1/fval2$ равен 0,3;
 $fval1+fval1/fval2$ ($=3+0,3=3,3$).

Результат выполнения операций в левой части присваивается целочисленной переменной `ival`, значение этой переменной будет равно 3 (дробная часть отбрасывается).



Преобразование типов данных в C++.

Задание. Определите значение переменной `fval`, полученное в результате выполнения фрагмента программы

```
int ival1=3, ival2=10;  
float fval=3.5;  
fval=fval+ival1/ival2;
```



Преобразование типов данных в C++.

Явное преобразование типов

Если в программе необходимо временно изменить тип некоторой переменной, то перед ее именем в круглых скобках указывается название соответствующего типа данных.

Например,

```
int ival1=3, ival2=10;  
float fval=3.5;  
fval=fval+(float)ival1/ival2; //  
fval=fval+ival1/(float)ival2;
```



Задания для самостоятельной работы

3. Определите, что будет выведено на экран в результате работы следующей программы.

```
#include <iostream>
#include <stdio.h>
using namespace std;
void main() {
    cout<<'A'<<endl;
    cout<<(int)'A'<<endl;
    getchar();
}
```

Наберите текст программы и проверьте правильность ответа.



Задания для самостоятельной работы

4. Следующая программа должна была вывести результат деления числа **a** на **2**. Выясните, корректно ли работает данная программа. Исправьте ее для корректной работы.

```
#include <iostream>
#include <stdio.h>
using namespace std;
void main() {
    int a=25;
    cout<<"a="<<a<<endl;
    a=a/2;
    cout<<"a/2="<<a<<endl;
    getchar();
}
```



Задания для самостоятельной работы

5. Определите, что будет выведено на экран в результате работы следующей программы.

```
#include <iostream>;
#include <stdio.h>;
using namespace std;
void main() {
    int a=2534;
    c=a/100;
    a=a%100;
    c=a-c;
    cout<<"a="<<a<<endl;
    cout<<"c="<<c<<endl;
    getch();
}
```

Наберите текст программы и проверьте правильность ответа.



Задания для самостоятельной работы

6. Дан прямоугольный треугольник с катетами a и b . Составьте программу для вычисления гипотенузы c .
7. Треугольник задан координатами своих вершин $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Составьте программу для вычисления длин его сторон, периметра и площади.
8. Даны два вещественных числа. Составьте программу для нахождения коэффициентов приведенного квадратного уравнения, корнями которого являются эти числа.
9. Составьте программу для нахождения числа сотен и тысяч в заданном шестизначном натуральном числе.
10. Составьте программу для нахождения второй справа цифры в записи заданного четырехзначного целого числа.



Задания для самостоятельной работы

11. Даны четыре целых числа (hour, min, sec, time). Первые три из них (hour, min, sec) — это время (в часах, минутах и секундах) начала просмотра фильма, четвёртое (time) определяет продолжительность фильма в секундах. Составьте программу для нахождения времени окончания просмотра фильма. Считать, что фильм шёл непрерывно.
12. На первую клетку шахматной доски положили 1 зерно, а на каждую следующую клетку на n зёрен больше, чем на предыдущую. Составьте программу, которая вычисляет, сколько всего зёрен оказалось на шахматной доске.

