

Глава 2 «Базовые концепции»

§2.2 Переменные

Объявление и инициализация переменных

Создание любой переменной резервирует некоторое количество памяти в зависимости от её типа. При объявлении переменной компилятору необходимо указать её **тип и идентификатор**. C++ предлагает большой ассортимент встроенных типов данных.

Идентификатор – это имя для переменной, он начинается с буквы (A-Z или a-z) или с нижнего подчеркивания (_), с последующими буквами, нижними подчёркиваниями и цифрами от 0 до 9.

Инициализация переменной – это присвоение начального значения переменной.

Для этого после идентификатора нужно прописать **оператор присвоения =**, а затем **нужное значение**.

Пример объявления и инициализации переменной

Тип переменной (Целочисленный) Значение переменной,
равное 10



```
int myVariable = 10;
```

Идентификатор (Имя
переменной)

Типы переменных

1) Целочисленный:

Тип	Размер в байтах	Диапазон значений
int	4	От -2147483648 (-2^{31}) до 2147483647 ($2^{31}-1$)
short	2	От -32768 (-2^{15}) до 32767 ($2^{15}-1$)
longint	8	От -9223372036854775808 (-2^{63}) до 9223372036854775807 ($2^{63}-1$)
Unsigned int/short/longint	4/2/8	От 0 до 4294967295 ($2^{32}-1$) / 65535 ($2^{16}-1$) / 18446744073709551615 ($2^{64}-1$)

Типы переменных

2) Вещественный (с плавающей точкой) :

Тип	Размер в байтах	Диапазон значений
float	4	От 1.5E-45 до 3.4E+38
double	8	От 5.0E-324 до 1.7E+308
long double	10	От 1.9E-4951 до 1.1E+4932

Типы переменных

3) Символьный:

Тип	Размер в байтах	Значения
char	1	256 различных значений символов (ASCII)
wchar_t	2	65536 различных символов (UNICODE)

4) Логический:

Тип	Размер в байтах	Значения
bool	1	true (истина), false (ложь)

Примеры

```
1. int var1 = 5;  
2. short var2 = -34;  
3. double var3 = 36.6;  
4. char var4 = 'a';  
5. bool var5 = true;
```

1. Целочисленный тип **int**, идентификатор **var1**, значение **5**.
2. Целочисленный тип **short**, идентификатор **var2**, значение **-34**.
3. Вещественный тип **double**, идентификатор **var3**, значение **36.6**.
4. Символьный тип **char**, идентификатор **var4**, значение **a**.
5. Логический тип **bool**, идентификатор **var5**, значение **true**.

Примеры

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

```
int main()
{
    double var1 = 10.6;
    cout << var1;
    return 0;
}
```

Пример программы,
которая создает
переменную типа double,
присваивает ей значение
10.6 и выводит его на
экран.

Вывод

10.6

Основные операции в C++

Основные операции:

- 1) Арифметические: $+$ (сложение), $-$ (вычитание), $*$ (умножение), $/$ (деление), $\%$ (остаток от деления).
- 2) Инкремент/декремент: $++$ (добавить единицу), $--$ (отнять единицу)
 1. Присваивания: $=$ (присвоить значение).
 2. $+=$, $*=$, $-=$, $/=$, $\%=$ (сокращенная запись арифметических операций + присваивание).

Основные операции в C++, примеры

```
int c = a + b;
```

с - это сумма а
и b

```
int c = a * b;
```

с - это
произведение а
и b

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

с - это разность
а и b

```
int c = a % b;
```

с - это остаток от
деления а на b

```
double c = a / b;
```

с - это частное
от деления а на
b

```
a++;
```

```
a--;
```

```
a*=3;
```

```
a-=2;
```

Увеличение а на 1

Уменьшение а на 1

Увеличение а в 3

раза

Уменьшение а на 2

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

```
int main()
```

```
{
```

```
    int a = 10;
```

```
    int b = 20;
```

```
    int c = 30;
```

```
    int d = a + b * c;
```

```
    cout << d;
```

```
    return 0;
```

```
}
```

Порядок выполнения операций
такой же, как и в математике
(сначала умножение и
деление, а затем сложение и
вычитание. Получим $10 + 20 * 30 = 610$)

Вывод

610

Неявное преобразование типов

Самая частая ошибка – это неявное преобразование типов, например:

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

int main()
{
    int a = 2;
    int b = 5;
    double c = a / b;
    cout << c;
    return 0;
}
```

Вывод

0

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

int main()
{
    double a = 2;
    double b = 5;
    double c = a / b;
    cout << c;
    return 0;
}
```

Вывод

0.4

На первом примере переменным a и b присвоен целочисленный тип int, поэтому при их делении происходит неявное преобразование переменной c в целочисленный тип. Поэтому берется целая часть от числа 0.4, то есть 0.

Чтобы это исправить, нужно изначально присвоить переменным a и b вещественный тип double.

Основные ошибки с типами

данных

Еще одна распространенная ошибка – это выход за пределы допустимых значений, например:

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

int main()
{
    short a = 20000;
    short b = 30000;
    cout << a + b;
    return 0;
}
```

Вывод

-17232

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

int main()
{
    int a = 20000;
    int b = 30000;
    cout << a + b;
    return 0;
}
```

Вывод

50000

Тип переменной short имеет диапазон до 32767, но при сложении a и b получаем 50000, что больше. Из-за этого мы выходим из диапазона.

Ошибка исправляется выбором другого типа переменной с большим диапазоном.