

Лекция 3. Типы данных и переменные Основные типы. целые, вещественные, символьные, логические. Константы, литералы, ключевые слова.

Типы данных и переменные

1. Введение

Любая программа на языке C++ работает с **данными**: числами, текстом, логическими значениями и т. д.

Для того чтобы компьютер понимал, **какого рода данные** нужно хранить и как с ними работать, в C++ используются **типы данных (data types)**.

Тип данных определяет:

- сколько памяти займёт значение;
 - какие операции с ним допустимы;
 - как интерпретировать хранимую информацию.
-

2. Переменные

2.1. Определение

Переменная (variable) — это именованная область памяти, предназначенная для хранения данных определённого типа.

Переменные можно изменять во время выполнения программы.

Общий формат объявления:

тип_данных имя_переменной [= значение];

Примеры:

```
int age = 20;           // целое число
double weight = 72.5;  // вещественное число
char grade = 'A';      // символ
bool isActive = true;  // логическое значение
```

2.2. Правила именования переменных

- Имя может содержать буквы, цифры и символ `_` (подчёркивание).
- Имя не может начинаться с цифры.
- Регистр имеет значение: `Age` и `age` — разные переменные.

- Нельзя использовать **ключевые слова** языка C++ как имена переменных (например, `int`, `for`, `class`).

Хорошие примеры:

```
int studentCount;  
float averageScore;  
bool isConnected;
```

Плохие примеры:

```
int 1value;    // ошибка: имя не может начинаться с  
цифры  
int double;    // ошибка: ключевое слово
```

3. Основные типы данных

C++ имеет богатую систему типов.

Базовые типы делятся на несколько категорий:

3.1. Целые типы (integer types)

Используются для хранения целых чисел (без дробной части).

Тип	Размер (байт)	Диапазон (примерно)	Пример
short	2	от -32,768 до 32,767	short s = 100;
int	4	от -2,147,483,648 до 2,147,483,647	int age = 25;
long	4 или 8	зависит от системы	long distance = 123456L;
long long	8	от -9e18 до 9e18	long long big = 1234567890123LL;
unsigned int	4	от 0 до 4,294,967,295	unsigned int count = 50;

Примечание:

`unsigned` означает, что число не может быть отрицательным, и весь диапазон используется для положительных значений.

3.2. Вещественные типы (floating-point types)

Используются для хранения чисел с дробной частью.

Тип	Размер	Точность	Пример
float	4 байта	~7 знаков после запятой	float pi = 3.14f;
double	8 байт	~15 знаков	double price = 125.456;
long double	10–16 байт	высокая точность	long double big = 2.718281828459;

Рекомендация:

Для научных и инженерных вычислений чаще используется double.

3.3. Символьный тип (character type)

Тип char используется для хранения **одного символа** (буквы, цифры, знака).

```
char letter = 'A';  
char digit = '7';
```

- Символы заключаются в **одинарные кавычки** ' '.
- На самом деле символ хранится как **число (код ASCII)**.

Пример:

```
char c = 'A';  
cout << int(c);    // выведет 65 — код ASCII
```

3.4. Логический тип (boolean type)

Тип bool используется для хранения **истинностных значений**: true или false.

```
bool isReady = true;  
bool hasError = false;
```

При выводе:

- true обычно отображается как 1
 - false — как 0
-

4. Размеры типов данных

Размер типов может отличаться в зависимости от компилятора и платформы. Для проверки можно использовать оператор `sizeof()`:

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

int main() {
    cout << "int: " << sizeof(int) << " байт" << endl;
    cout << "double: " << sizeof(double) << " байт" <<
endl;
    cout << "char: " << sizeof(char) << " байт" <<
endl;
    cout << "bool: " << sizeof(bool) << " байт" <<
endl;
}
```

Вывод (пример для x64):

```
int: 4 байт
double: 8 байт
char: 1 байт
bool: 1 байт
```

5. Константы (const)

Константа (constant) — это переменная, значение которой **нельзя изменить** после инициализации.

Объявляется с помощью ключевого слова `const`:

```
const double PI = 3.14159;
const int DAYS_IN_WEEK = 7;
```

Попытка изменить значение константы вызывает ошибку:

```
PI = 3.15; // Ошибка!
```

5.1. Препроцессорные константы

Можно также использовать директиву препроцессора (решотка) `define`:

```
(решотка) define MAX_VALUE 100
```

Разница:

- `const` учитывается компилятором, имеет тип и область видимости.
 - (решотка) `define` просто текстовая подстановка до компиляции.
-

6. Литералы (литеральные значения)

Литерал — это непосредственно записанное в коде значение.

Тип литерала	Примеры
Целые	10, -25, 0xFF, 075
Вещественные	3.14, 2.5e6
Символьные	'A', '\n', '%'
Строковые	"Hello", "C++ is fun!"
Булевы	true, false

Примечание:

Символьные литералы в одиночных кавычках ' ', строковые — в двойных " ".

7. Ключевые слова языка C++

Ключевые слова (keywords) — это зарезервированные слова, которые имеют особое значение в языке и не могут использоваться как имена переменных или функций.

Примеры ключевых слов C++:

```
int, float, double, char, bool,  
if, else, for, while, switch,  
return, class, struct, public, private,  
new, delete, const, static, void,  
namespace, using, try, catch, throw
```

Всего в C++ более 90 ключевых слов (в зависимости от стандарта).

8. Инициализация переменных

Переменную можно инициализировать несколькими способами:

```
int a = 10;           // Классическая форма
int b(20);            // Конструкторная инициализация
int c{30};            // Универсальная (C++11)
```

Рекомендуется использовать фигурные скобки `{ }` — они предотвращают ошибки преобразования типов.

9. Типизация и преобразование типов

C++ — **строго типизированный язык**, но допускает явное и неявное преобразование типов.

9.1. Неявное (автоматическое) преобразование

Происходит, если типы совместимы:

```
int a = 5;
double b = a;    // int → double автоматически
```

9.2. Явное (ручное) преобразование (type casting)

Программист указывает тип явно:

```
double d = 9.8;
int x = (int)d;           // старый стиль
int y = static_cast<int>(d); // современный стиль
```

10. Пример программы

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

int main() {
    int age = 21;
    double height = 1.82;
    char grade = 'A';
    bool isStudent = true;
    const double PI = 3.14159;

    cout << "Возраст: " << age << endl;
    cout << "Рост: " << height << endl;
    cout << "Оценка: " << grade << endl;
```

```
    cout << "Студент: " << isStudent << endl;  
    cout << "Число Пи: " << PI << endl;  
}
```

Вывод:

Возраст: 21
Рост: 1.82
Оценка: A
Студент: 1
Число Пи: 3.14159

11. Заключение

В C++ тип данных — это фундамент любого выражения и операции. От правильного выбора типа зависит:

- точность вычислений;
- использование памяти;
- корректность программы.

Переменные и константы обеспечивают хранение и изменение данных, а строгая типизация делает программы надёжными и предсказуемыми.

12. Вопросы для самопроверки

1. Что такое переменная и как она объявляется в C++?
2. Перечислите основные категории типов данных.
3. Чем отличается `float` от `double`?
4. Что такое константа и как она объявляется?
5. В чём разница между `const` и (решотка) `define`?
6. Что такое литералы? Приведите примеры.
7. Почему нельзя использовать ключевые слова в качестве имён переменных?
8. Что делает оператор `sizeof()`?
9. Что такое явное и неявное преобразование типов?
10. Почему важно выбирать правильный тип данных при программировании?