

# Лекция №1. Типы данных. Лекция №1

## Виды целых типов:

- Знаковые/signed (- ... 0 ... +)
- Беззнаковые/unsigned (0 ...+).

Объем	Наименование типа	Диапазон
8 байт	Int64 / long long	-9 223 372 036 854 775 808 до 9 223 372 036 854 775 807
4 байт	int	-2 147 483 648 до 2 147 483 647 (В зависимости от архитектуры)
2 байт	short int	-32768 до 32767
1 байт	char	-128 до 127
8 байт	double	$\pm 1.7 * 10^{\pm 308}$ (15 символов)
4 байт	float	$\pm 3.8 * 10^{\pm 38}$ (7 символов)

Для того, чтобы создать беззнаковый тип, перед именем знакового указывается слово **unsigned**.

```
unsigned int a;  
unsigned char b;
```

**Указатель** (модификатор типа) — это адрес памяти, по которому размещены какие-либо данные. Для объявления переменной указателя, используется модификатор типа "\*". На 32-разрядных платформах размер адресов равен 4 байтам, а на 64-разрядных - 8 байтам

```
double x = 4.9;  
double *p;
```

Для указателей определены две операции:

& — идентификатор типа переменной (результат = адрес этой переменной).

\* — разыменование (результат = данные по этому адресу).

```
double x = 4.9;
double *p = &x, *p2;    // в p записываем ссылку на x
p2 = p;                  // в p2 кидаем ссылку на p ⇒ ссылку на x
*prz = x * 2;            // по ссылке кладем в x значени x * 4.9 = 9.8

// result: x = 9.8
```

---

[Перечисление](#) (перечисляемый тип). С точки зрения компилятора перечисляемые объекты это целые числа. Применяется для улучшения читаемости программы. Программист имеет возможность задавать элементам перечисления произвольные целочисленные значения. При этом действует правило, что по умолчанию элемент перечисления имеет значение на единицу больше предыдущего.

```
enum [ТИП] {
    список_перечисление
} [переменные];

// Имена переменных принято писать капсом
```

---

```
enum RGB { RED, GREEN, BLUE } c1;

c1 = RED;

RGB c2; // C++
enum RGB c2; // так в чистом C
```