

# Содержание

<b>переменные и литералы</b> .....	3
<b>Переменные и их базовые типы</b> .....	3
Модификаторы signed и unsigned .....	3
Модификатор const .....	4
<b>Числовые и символьные литералы</b> .....	4



# переменные и литералы

Переменные бывают глобальными, локальными, статическими и внешними. См [классы хранения / scope](#)

## Переменные и их базовые типы

```
int a; // объявление (случайное значение)
int a = 10; инициализация при объявлении
```

имена идентификаторов регистрозависимы и могут состоять из следующих символов: a-z, A-Z, 0-9, `_`. См [идентификаторы](#)

Тип данных	Для 16-битных	Для 32-битных	Для 64-битных	Описание	Диапазон
char	1	1	1	Символьный и целочисленный	[0; 255] или [-128; 127]
short [int]	2	2	2	Целочисленный	[-32768; 32767]
int	2	4	4	Целочисленный	[-32768; 32767] или [-2147483648; 2147483647]
long [int]	4	4	8	Целочисленный	зависит от платформы (см. столбец)
long long [int]	8	8	8	Целочисленный	зависит от реализации (обычно 64-битный, см. столбец)
float	4	4	4	Вещественный	см. IEEE-754 single (примерно $\pm 3.4e38$ , ~6-9 значащих цифр)
double	8	8	8	Вещественный	см. IEEE-754 double (примерно $\pm 1.7e308$ , ~15-17 значащих цифр)
long double	10	10	10	Вещественный	платформа-зависимо (часто 80-битное расширение на x86, иначе совпадает с double)

для получения размера переменных используется оператор `sizeof()`, возвращающий размер в байтах типа `size_t`. Он так же может принимать тип в качестве аргумента.



Тип результата оператора `sizeof` - `size_t` фактически является псевдонимом для типа `unsigned long long`

## Модификаторы `signed` и `unsigned`

- `unsigned` - беззнаковый;

- `signed` – знаковый (появился в стандарте ANSI C90).



Обратите внимание, когда я говорю о модификаторах `signed` и `unsigned`, то отмечаю именно целочисленные типы данных. Дело в том, что вещественные типы `float` и `double` всегда знаковые и применять к ним эти модификаторы нельзя

## Модификатор `const`

Объект с модификатором `const` становится константным, тоесть неизменяемым.

```
const float PI = 3.14;
```

## Числовые и символьные литералы

```
int dec, hex, oct;
dec = 100;
hex = 0x1FA;
oct = 0123;

double d1, d2, d3, d4;
d1 = 10.0;
d2 = -7.;
d3 = 1e2; // <число>e<степень десятки>
d4 = 5e-3;
```

При желании мы можем явно указать компилятору тип числового литерала. Для этого используются следующие суффиксы:

- `U` или `u` – использование модификатора `unsigned` в определении литерала;
- `L` или `l` – использование типа `long` при определении литерала;
- `LL` или `ll` – использование типа `long long` при определении литерала.

```
int dec_i, dec_ui, dec_l, dec_ul;
float var_f;

dec_i = 100; // тип литерала int
dec_ui = 100U; // тип литерала unsigned int
dec_l = 100L; // тип литерала long
dec_ul = 100UL; // тип литерала unsigned long
var_f = 100.0f // тип литерала float
```

From:

<https://wiki.radi0.cc/> - **radi0wiki**

Permanent link:

[https://wiki.radi0.cc/c:c\\_ultimate\\_guide:variables](https://wiki.radi0.cc/c:c_ultimate_guide:variables)

Last update: **2025/11/09 12:07**

