

# Дополнительные математические функции:

1. Функция `fmod(x,y)` - остаток деления  $x$  на  $y$ ;
  - проверка кратности;
  - проверка четности / нечетности числа.

Проверка кратности (для типа `double`):

```
fmod(x, y) // #include <math.h>
```

```
if (fmod(x, y) == 0)
```

Условие:  $x$  кратно  $y$

Проверка кратности (для типа `int`):

операция `%`

```
if (x % y == 0)
```

Условие:  $x$  кратно  $y$

# Дополнительные математические функции:

Пример кратности:

Условие: z кратно 7

`fmod (z, 7) == 0`

Проверка четности числа:

`#include <math.h>`

`fmod (x, 2) == 0`

Проверка нечетности числа:

`#include <math.h>`

`fmod (x, 2) != 0`    или    `fmod (x, 2) == 1`

# Дополнительные математические функции:

## 2. Функция $\text{int}(x)$

- целая часть числа;
- отбрасывается дробная часть.

- дробная часть числа:

$$x - \text{int}(x)$$

$$\text{fmod}(x, 1)$$

```
double a = 452.554;
```

```
int b;
```

```
b = a;
```

автоматическое приведение типов. В b попадет значение a без дробной части. Но компилятор может выдавать предупреждение о возможной потере данных

```
b = int(a);
```

тоже самое, что и выше, но предупреждений со стороны компилятора не будет.

```
#include <math.h>
```

```
double c;
```

```
c = floor(a); // округление вниз до ближайшего целого
```

```
c = ceil(a); // округление вверх до ближайшего целого
```

При этом результат все равно останется типа с плавающей точкой, который можно привести к целому типу, как выше