Линейные алгоритмы обработки целочисленных данных

Основные алгоритмические структуры

Повторение

Повторение

В языке Python используется динамическая типизация.

Это означает, что в программах на нём переменные объявляются автоматически при первом использовании, а также в ходе исполнения программы тип переменных может изменяться в зависимости от того, какое значение присваивается переменной.

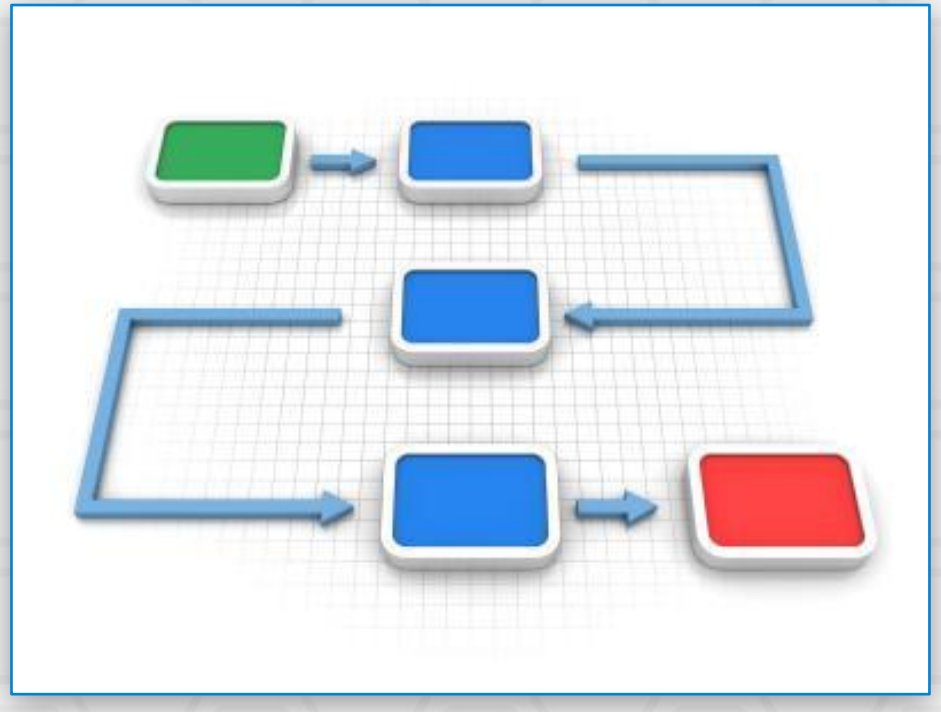
Вопросы к изучению

1. Линейные алгоритмы.
2. Правила записи арифметических выражений.
3. Инструменты обработки целых чисел.

Линейные алгоритмы

Линейные алгоритмы –

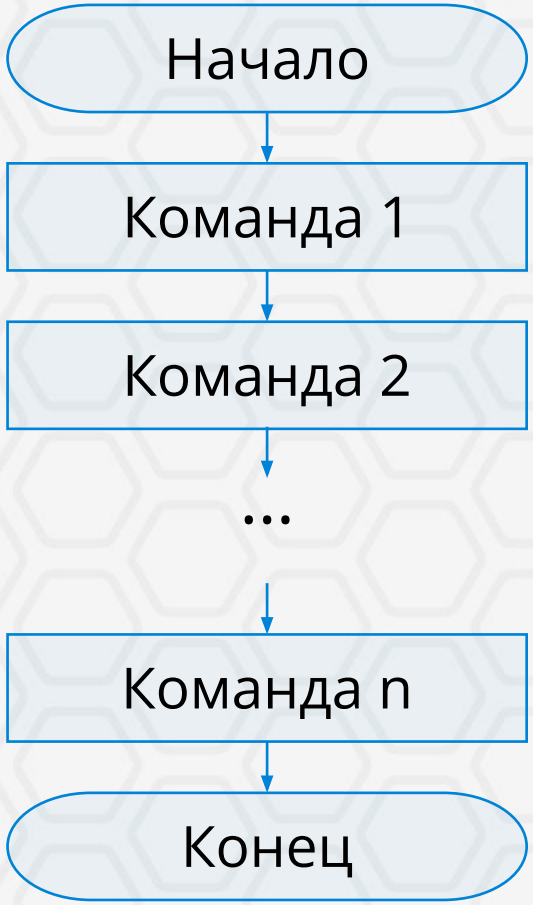
это алгоритмы, в которых команды выполняются последовательно, в том порядке, в котором они записаны.



Линейные алгоритмы

Линейные алгоритмы –

это алгоритмы, в которых команды выполняются последовательно, в том порядке, в котором они записаны.



Запись арифметических выражений в языке Python

Арифметические выражения могут содержать:

- ✔ числа;
- ✔ скобки;
- ✔ знаки арифметических операций;
- ✔ имена переменных;
- ✔ вызовы функций.

Запись арифметических выражений в языке Python

Арифметические выражения могут содержать:

- ✔ числа;
- ✔ скобки;
- ✔ знаки арифметических операций;
- ✔ имена переменных;
- ✔ вызовы функций.

Знак переноса в арифметических выражениях:

<Часть 1>\
<Часть 2>

Пример использования:

```
a = (75 - 7) * \
6 * 8 - 1
```

Запись арифметических выражений в языке Python

Приоритет выполнения арифметических операций:

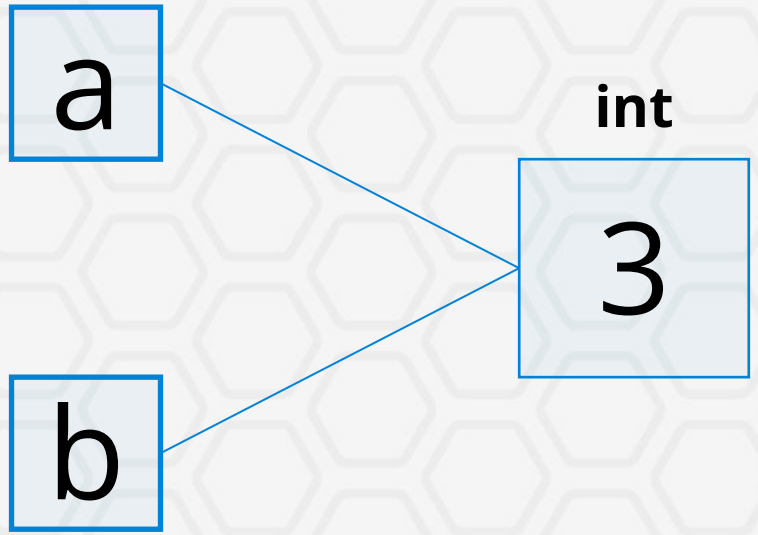
1. Выражения в скобках.
2. Операции возведения в степень.
3. Операции умножения и деления.
4. Операции сложения и вычитания.

Множественное присваивание

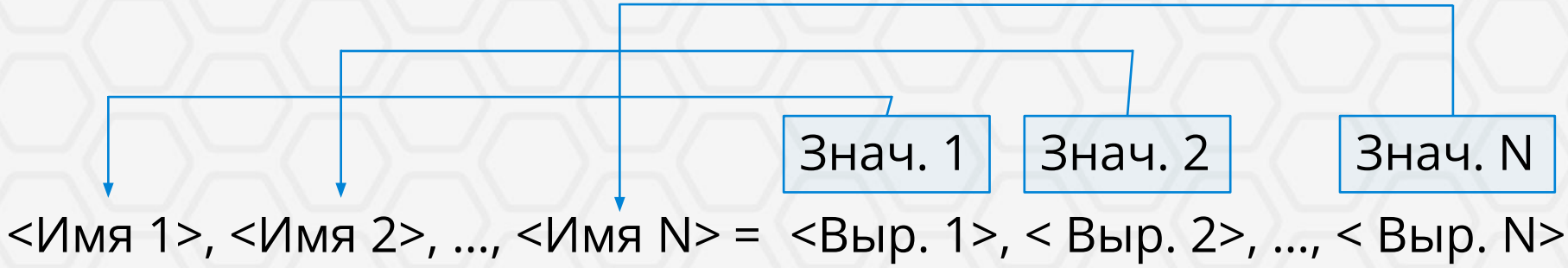
Пример:

```
b = a = 3
```

```
a = 3
b = a
```



Множественное присваивание



Важно:

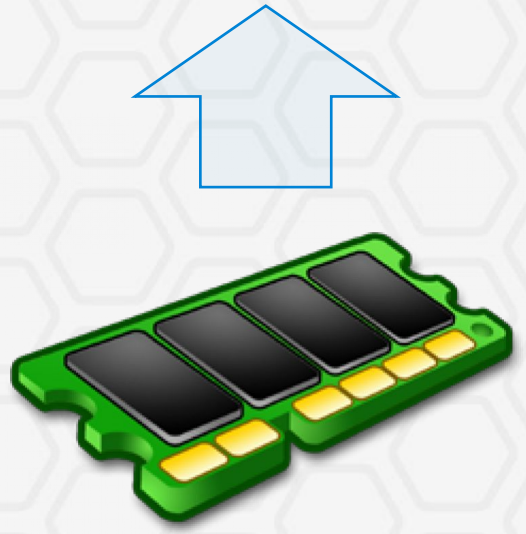
количество и порядок
следования имён переменных
и присваиваемых значений
должны совпадать.

Пример:

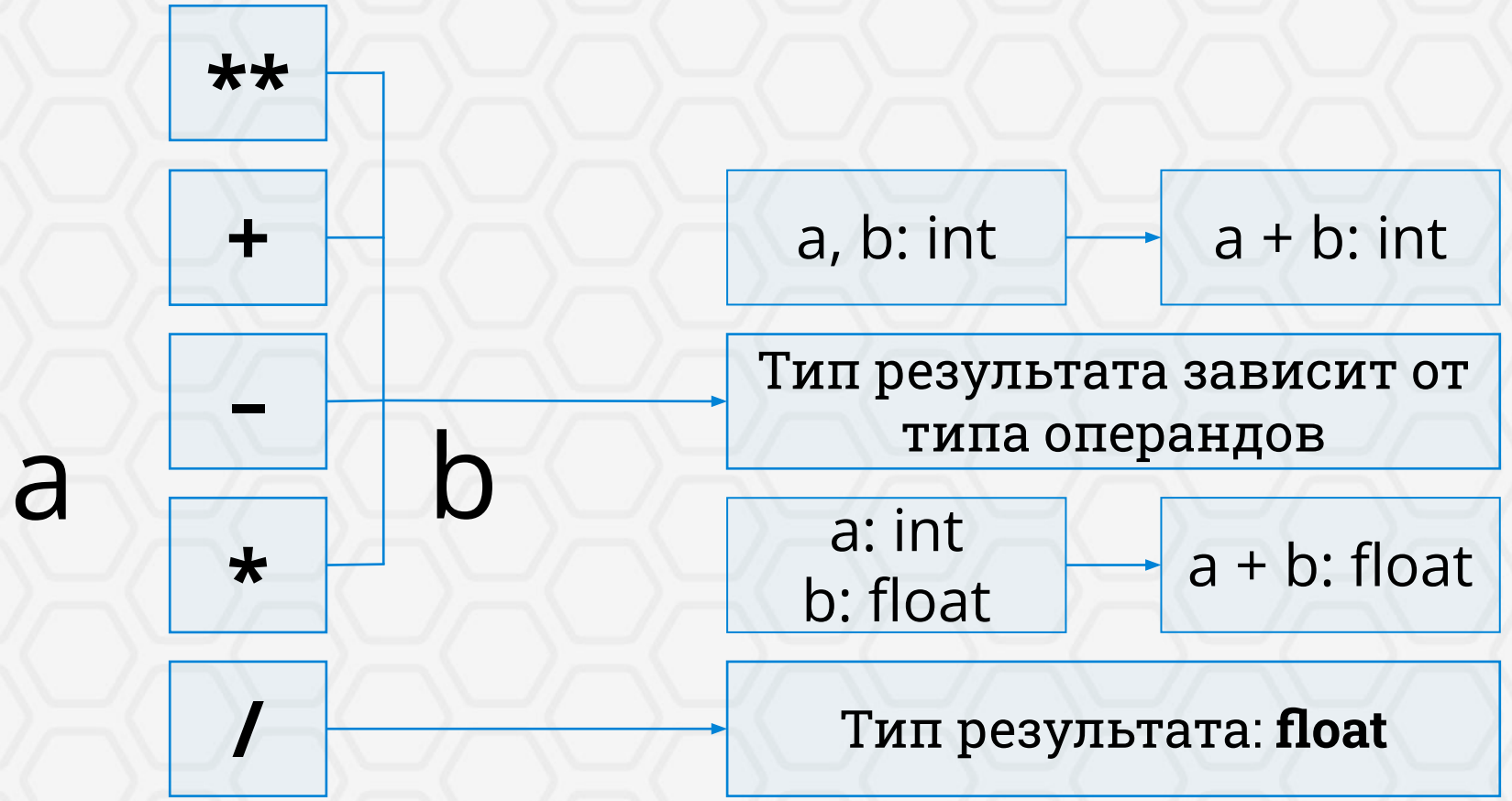
```
a, b = b, a
```

Целочисленный тип данных (int)

205 112 782 345 981 400



Операции обработки числовых данных



Функция модуля числа

Запись:

abs (<аргумент>)

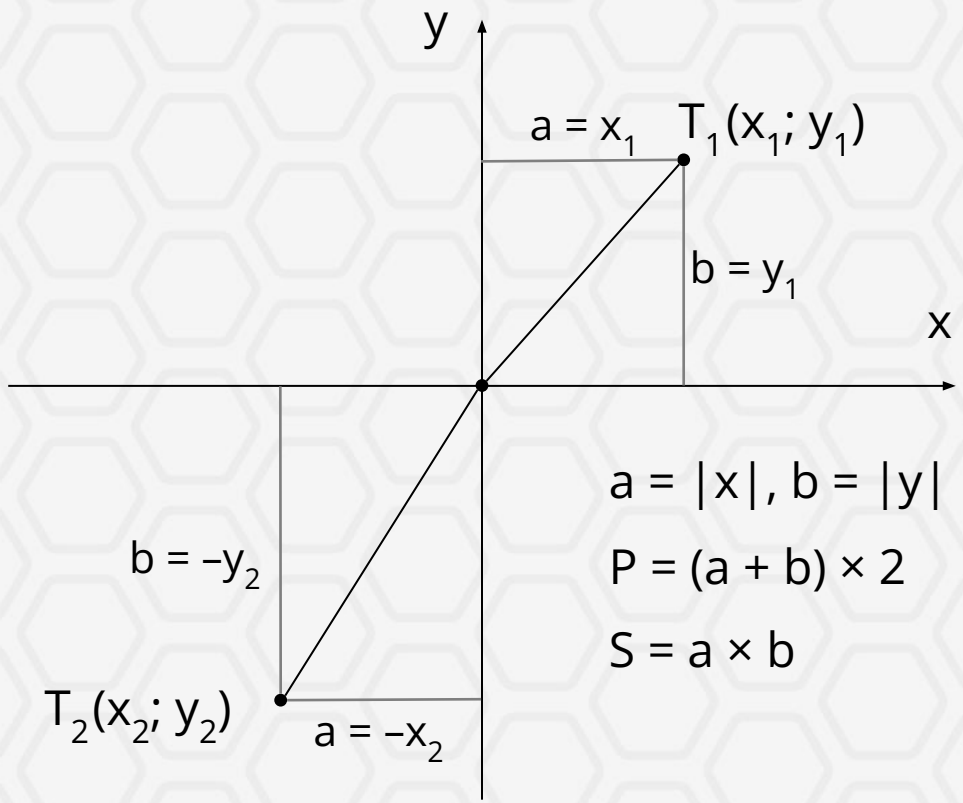
Модуль числа –

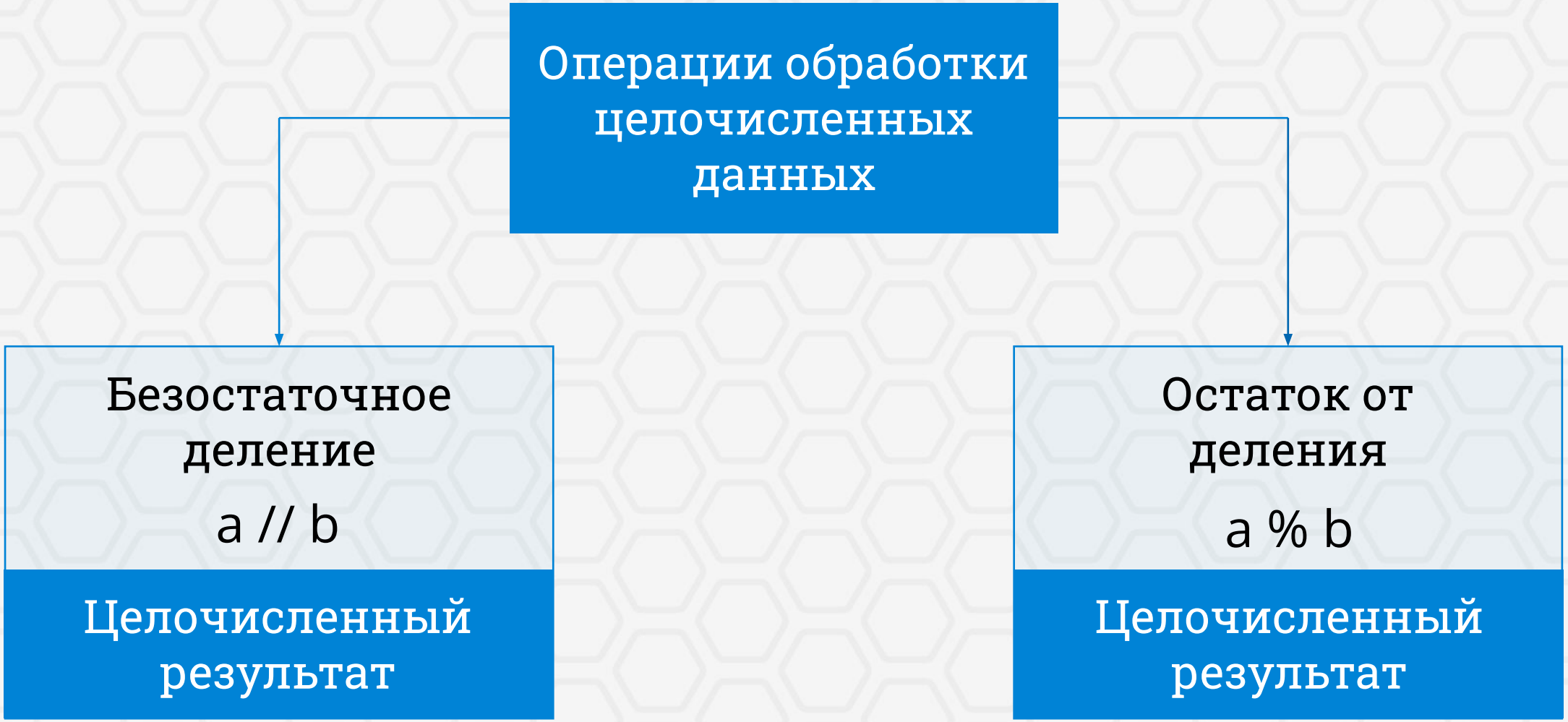
это расстояние на числовой оси от точки начала отсчёта до точки, соответствующей этому числу.

При $a \geq 0$, $|a| = a$,
при $a < 0$, $|a| = -a$.

Задача

Даны целочисленные ненулевые координаты точки T на плоскости. Из начала отсчёта в точку T провели отрезок, который является диагональю прямоугольника. Вычислить площадь и периметр этого прямоугольника.





Случайные числа

$$R_в[0{,}5(n_ц - 1)] = R_в[0{,}5(n_ц + 1)] = -0{,}5\left[1 + 2\sum_{n_т=1}^{0{,}5n_ц - 1{,}5} R_о(n_т)\right]$$

Случайные числа

Подключение модуля:

```
import random
```

Фукция randint:

random.randint (<нач.>, <кон.>)

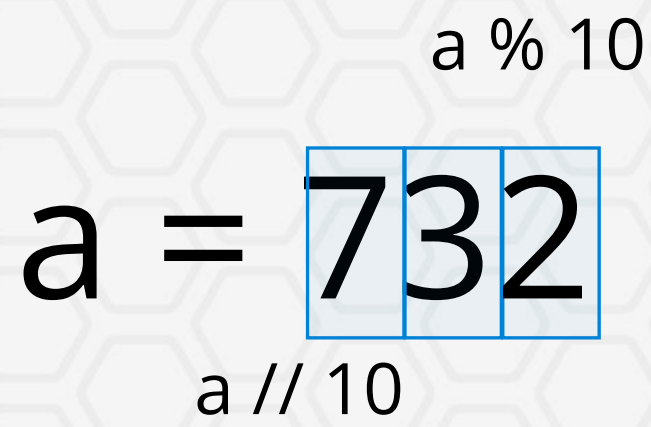
Точечный вызов

Пример использования:

```
import random
t = random.randint (a, b)
```

Задача

Написать модуль, который генерирует случайное целое трёхзначное число и вычисляет сумму его цифр. На экран должно быть выведено полученное случайное число и сумма его цифр.



Линейные алгоритмы обработки целочисленных данных

Линейные алгоритмы –

это алгоритмы, в которых команды выполняются последовательно, в том порядке, в котором они записаны.

Операции обработки целочисленных данных:

- ✔ безостаточное деление – a // b;
- ✔ остаток от деление – a % b.

Функция модуля числа:

abs (a)

Функция генерации целого случайного числа:

random.randint (a, b)