

# Матрицы

Создать класс **Matrix**, который будет реализовывать следующие возможности.

Класс должен принимать список списков.

Например:

`m = Matrix([[ -1, 3], [0, 1], [ -2, 2]])`

$$\begin{pmatrix} -1 & 3 \\ 0 & 1 \\ -2 & 2 \end{pmatrix}$$

будет представлять матрицу

Пункты **синего цвета** означают методы, которые должны возвращать новый экземпляр класса **Matrix**.

(т.е. каждый такой метод должен возвращать **self**)

Пункты **зеленого цвета** — это методы основанные на `@classmethod`

- **Сложение матриц (только одинаковых размерностей):**

$$\begin{pmatrix} -1 & 3 \\ 0 & 1 \\ -2 & 2 \end{pmatrix} + \begin{pmatrix} 2 & 0 \\ -1 & 1 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} -1+2 & 3+0 \\ 0+(-1) & 1+1 \\ -2+3 & 2+(-2) \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ -1 & 2 \\ 1 & 0 \end{pmatrix}$$

- **Вычитание матриц (точно так же).**
- **Умножение матрицы на число.**
- **Транспонирование матрицы:**

Пример:

$$A = \begin{pmatrix} 12 & -1 \\ -5 & 0 \end{pmatrix}$$
$$A^T = \begin{pmatrix} 12 & -5 \\ -1 & 0 \end{pmatrix}$$

- Создает единичную матрицу размером  $m, n$
- Создает нулевую матрицу размером  $m, n$
- Создает диагональную матрицу из переданного списка

**Диагональная матрица** — квадратная матрица, все элементы которой, стоящие вне главной диагонали, равны нулю.

Диагональная матрица имеет вид:

$$A = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

- Возвращает размерность матрицы (кортеж)
- Возвращает кол-во элементов в матрице
- Возвращает сумму всех элементов матрицы
- Возвращает новую матрицу, где вместо отрицательных чисел стоят нули
- Возможность сравнения на равенство двух матриц
- Переопределить метод `__str__`