

Добавьте в предыдущий класс следующие методы:

- `__add__` принимающий вторую матрицу того же размера и возвращающий сумму матриц
- `__mul__` принимающий число типа `int` или `float` и возвращающий матрицу, умноженную на скаляр
- `__rmul__` делающий то же самое, что и `__mul__`. Этот метод будет вызван в том случае, аргумент находится справа. Можно написать `__rmul__ = __mul__`

Например:

В этом случае вызовется `__mul__`: `Matrix([[0, 1], [1, 0]]) * 10`

В этом случае вызовется `__rmul__` (так как у `int` не определен `__mul__` для матрицы справа): `10 * Matrix([[0, 1], [1, 0]])`

Разумеется, данные методы не должны менять содержимое матрицы.

Пример 1

### Ввод

```
# Task 2 check 1

m = Matrix([[10, 10], [0, 0], [1, 1]])

print(m.size())
```

### Вывод

```
(3, 2)
```

Пример 2

### Ввод

```
# Task 2 check 2

m1 = Matrix([[1, 0, 0], [0, 1, 0], [0, 0, 1]])

m2 = Matrix([[0, 1, 0], [20, 0, -1], [-1, -2, 0]])

print(m1 + m2)
```

### Вывод

```
1      1      0
20     1     -1
-1     -2     1
```

## Пример 3

### Ввод

```
# Task 2 check 3

m = Matrix([[1, 1, 0], [0, 2, 10], [10, 15, 30]])

alpha = 15

print(m * alpha)

print(alpha * m)
```

### Вывод

15	15	0
0	30	150
150	225	450
15	15	0
0	30	150
150	225	450