

```

1 """Урок 7 Задание 1
2 Реализовать класс Matrix (матрица). Обеспечить
3 перегрузку конструктора класса (метод
4 __init__()), который должен принимать данные (список
5 списков) для формирования матрицы.
6 Подсказка: матрица – система некоторых математических
7 величин, расположенных в виде
8 прямоугольной схемы.
9 Следующий шаг – реализовать перегрузку метода __str__
10 ( ) для вывода матрицы в
11 привычном виде.
12 Далее реализовать перегрузку метода __add__() для
13 реализации операции сложения двух
14 объектов класса Matrix (двух матриц). Результатом
15 сложения должна быть новая матрица.
16 Подсказка: сложение элементов матриц выполнять
17 поэлементно – первый элемент первой
18 строки первой матрицы складываем с первым элементом
19 первой строки второй матрицы и т.д.
20 """
21
22
23 m_1 = [[3, 5, 32], [2, 4, 6], [-1, 64, -8]]
24 m_2 = [[5, 7, 23], [9, 23, -54], [12, 3, 16]]
25
26
27 class Matrix_1:
28     def __init__(self, data):
29         self.data = data
30
31     def __str__(self):
32         return '\n'.join('\t'.join([str(itm) for itm
33 in line]) for line in self.data)
34
35     def __add__(self, other):
36         try:
37             add = [[int(self.data[line][itm]) + int(
38 other.data[line][itm]) for itm in range(len(self.data
39 [line]))]
40
41                     for line in range(len(self.data))]
42             return Matrix_1(add)
43         except IndexError:
44             return f'Data error'
45
46
47
48
49
50
51
52
53

```

```

34 mtx_1 = Matrix_1(m_1)
35 mtx_2 = Matrix_1(m_2)
36 mtx_3 = mtx_1 + mtx_2
37
38 print(f'---first way---')
39 print(f'{mtx_1}\n{"*" * 15}')
40 print(f'{mtx_2}\n{"*" * 15}')
41 print(f'{mtx_3}')
```

42

43

44 *# another way*

```

45 class Matrix_2:
46     def __init__(self, data):
47         self.data = data
48
49     def __str__(self):
50         return '\n'.join(map(str, self.data))
51
52     def __add__(self, other):
53         add = []
54         for el_i in range(len(self.data)):
55             add.append([])
56             for el_j in range(len(self.data[0])):
57                 add[el_i].append(self.data[el_i][el_j]
58 ] + other.data[el_i][el_j])
59         return '\n'.join(map(str, add))
60
61 mtx_4 = Matrix_2(m_1)
62 mtx_5 = Matrix_2(m_2)
63 mtx_6 = mtx_4 + mtx_5
64 print(f'\n---second way---')
65 print(f'{mtx_4}\n{"*" * 15}')
66 print(f'{mtx_5}\n{"*" * 15}')
67 print(f'{mtx_6}')
```

68

69

70 *# another way*

```

71 class Matrix_3:
72     def __init__(self, data):
73         self.data = data
74
75     def __str__(self):
76         return '\n'.join(map(lambda r: '\t'.join(map(
```

```
76 str, r)), self.data)) + '\n'
77
78     def __add__(self, other):
79         return Matrix_3(map(lambda r_1, r_2: map(
            lambda x, y: x + y, r_1, r_2), self.data, other.data
        ))
80
81 mtx_7 = Matrix_2(m_1)
82 mtx_8 = Matrix_2(m_2)
83 mtx_9 = mtx_7 + mtx_8
84 print(f'\n---third way---')
85 print(f'{mtx_7}\n{"*" * 15}')
86 print(f'{mtx_8}\n{"*" * 15}')
87 print(f'{mtx_9}')
```