

## Практическое задание 2.1

In [1]:

```
1 import numpy as np
```

1. Установить, какие произведения матриц  $AB$  и  $BA$  определены, и найти размерности полученных матриц:

а)  $A$  — матрица  $4 \times 2$ ,  $B$  — матрица  $4 \times 2$ ;

Вспомним, что матрицу  $A$  можно умножить не на всякую матрицу  $B$ : **необходимо, чтобы число столбцов матрицы  $A$  было равно числу строк матрицы  $B$ .**

По этому правилу видим, что произведение  $AB$  не определено и произведение  $BA$  также не определено

б)  $A$  — матрица  $2 \times 5$ ,  $B$  — матрица  $5 \times 3$

По этому правилу видим, что произведение  $AB$  определено (размер новой матрицы  $2 \times 3$ ), а произведение  $BA$  не определено

в)  $A$  — матрица  $8 \times 3$ ,  $B$  — матрица  $3 \times 8$

По этому правилу видим, что произведение  $AB$  определено (размер новой матрицы  $8 \times 8$ ) и произведение  $BA$  также определено (размер новой матрицы  $3 \times 3$ )

г)  $A$  — квадратная матрица  $4 \times 4$ ,  $B$  — квадратная матрица  $4 \times 4$

По этому правилу видим, что произведение  $AB$  определено и произведение  $BA$  также определено (размер обеих новых матриц  $4 \times 4$ )

2. Найти сумму и произведение матриц  $A = \begin{pmatrix} 1 & -2 \\ 3 & 0 \end{pmatrix}$  и  $B = \begin{pmatrix} 4 & -1 \\ 0 & 5 \end{pmatrix}$ .

Сумма данных матриц равна:  $C = \begin{pmatrix} 5 & -3 \\ 3 & 5 \end{pmatrix}$

Проверим

In [2]:

```
1 A = np.array([[1, -2], [3, 0]])
2 B = np.array([[4, -1], [0, 5]])
3
4 A + B
```

Out[2]:

```
array([[ 5, -3],
       [ 3,  5]])
```

Видим, что мои расчеты оказались верными

Произведение я посчитаю с помощью моей функции из 4 задания. Данная функция работает без numpy, а следовательно я разобрался в том, как перемножаются матрицы и здесь нет смысла переписывать это.

In [3]:

```
1 def product_matrices(A, B):
2     if A.shape[1] != B.shape[0]:
3         print(f'Матрицу с размером {A.shape} нельзя перемножить на матрицу с размером {B.shape}')
4     else:
5         C = [[0 for i in range(A.shape[0])] for i in range(B.T.shape[0])]
6
7         for i in range(A.shape[0]):
8             for j in range(B.T.shape[0]):
9                 for k in range(B.shape[0]):
10                    C[i][j] = C[i][j] + (A[i, k] * B.T[j, k])
11     return C
```

In [4]:

```
1 np.array(product_matrices(A, B))
```

Out[4]:

```
array([[ 4, -11],
       [12, -3]])
```

Проверим

In [5]:

```
1 A @ B
```

Out[5]:

```
array([[ 4, -11],
       [12, -3]])
```

Видим, что произведение посчитано правильно

3. Из закономерностей сложения и умножения матриц на число можно сделать вывод, что матрицы одного размера образуют линейное пространство. Вычислить линейную комбинацию  $3A - 2B + 4C$  для матриц  $A = \begin{pmatrix} 1 & 7 \\ 3 & -6 \end{pmatrix}$ ,  $B = \begin{pmatrix} 0 & 5 \\ 2 & -1 \end{pmatrix}$ ,  $C = \begin{pmatrix} 2 & -4 \\ 1 & 1 \end{pmatrix}$ .

Зададим матрицы

In [6]:

```
1 A = np.array([[1, 7], [3, -6]])
2 B = np.array([[0, 5], [2, -1]])
3 C = np.array([[2, -4], [1, 1]])
```

Воспользуемся для расчетов функцией произведения матрицы на число без использования numpy.

In [7]:

```
1 def product_matric_num(A, B):
2     for i in range(B.shape[0]):
3         for j in range(B.shape[1]):
4             B[i][j] = A * B[i][j]
5
6     return B
```

In [8]:

```
1 product_matric_num(3, A) - product_matric_num(2, B) + product_matric_num(4, C)
```

Out[8]:

```
array([[ 11,  -5],
       [  9, -12]])
```

Проверим расчеты

In [9]:

```
1 A = np.array([[1, 7], [3, -6]])
2 B = np.array([[0, 5], [2, -1]])
3 C = np.array([[2, -4], [1, 1]])
4 3 * A - 2 * B + 4 * C
```

Out[9]:

```
array([[ 11,  -5],
       [  9, -12]])
```

Видим, что расчеты прведены верно

4\*. Написать на Python функцию для перемножения двух произвольных матриц, не используя NumPy.

In [10]:

```
1 A = np.array([[1, 0], [2, 1], [10, 5]])
2 B = np.array([[2, 0, 0], [0, 0, 1]])
```

In [11]:

```
1 def product_matrices(A, B):
2     if A.shape[1] != B.shape[0]:
3         print(f'Матрицу с размером {A.shape} нельзя перемножить на матрицу с размером {B.shape}')
4     else:
5         C = [[0 for i in range(A.shape[0])] for i in range(B.T.shape[0])]
6
7         for i in range(A.shape[0]):
8             for j in range(B.T.shape[0]):
9                 for k in range(B.shape[0]):
10                    C[i][j] = C[i][j] + (A[i, k] * B.T[j, k])
11     return C
```

In [12]:

```
1 product_matrices(A, B)
```

Out[12]:

```
[[2, 0, 0], [4, 0, 1], [20, 0, 5]]
```

В более привычном виде:

In [13]:

```
1 np.array(product_matrices(A, B))
```

Out[13]:

```
array([[ 2,  0,  0],
       [ 4,  0,  1],
       [20,  0,  5]])
```

Проверим правильность перемножения с помощью средств numpy

In [14]:

```
1 A @ B
```

Out[14]:

```
array([[ 2,  0,  0],
       [ 4,  0,  1],
       [20,  0,  5]])
```

Видим, что все считается правильно