Домашнее задание №4

NumPy

Формулировка задания:

Научиться работать с фреймворком NumPy

План работы:

1) Создать ноутбук в Google Colab
2) Решить предложенные математические задачи с помощью NumPy (1 и 2 задачи обязательны к выполнению, остальные желательны)
3) Открыть доступ для чтения ноутбука по ссылке
4) Прикрепить ссылку на ноутбук в качестве ответа на домашнее задание на платформе learn.innopolis.university

Перечень инструментов, необходимых для реализации деятельности:

Google Colab https://colab.research.google.com/

```
# подключение библиотек
import numpy as np
```

Задача 1

1. Выполнить операции над матрицами:

   Даны матрицы

$$A = \begin{pmatrix} 1 & 2 & 1 \\ -2 & 1 & 1 \end{pmatrix}, B = \begin{pmatrix} 3 & 1 \\ -1 & 0 \\ 2 & 4 \end{pmatrix}, C = \begin{pmatrix} -2 & 2 \\ 1 & -1 \\ 1 & 2 \end{pmatrix}$$

Найти

A + 2B

B + 2C

```
A = np.matrix([[ 1,  2,  1],
               [-2,  1,  1]])
B = np.matrix([[ 3,  1],
               [-1,  0],
               [ 2,  4]])
C = np.matrix([[-2,  2],
               [ 1, -1],
               [ 1,  2]])
```

```
# матрицу A траспонируем т.к. Складывать допускается только матрицы одинаковой размерности.
A.transpose() + 2 * B
```

```
matrix([[7, 0],
        [0, 1],
        [5, 9]])
```

```
B + 2 * C
```

```
matrix([[-1,  5],
        [ 1, -2],
        [ 4,  8]])
```

Задача 2

1. Создать матрицу размером 8x8 элементов, состоящую из нулей. Заполнить эту матрицу значениями, расположенными в шахматном порядке.

   Задачу решить через срезы массива NumPy (должно получиться буквально двумя командами).

   Должно получиться так:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

2. Вывести полученную матрицу на экран.

```
matrix = np.zeros((8,8), dtype='int32')
matrix[::2, 1::2] = 1
matrix[1::2, ::2] = 1
print(matrix)
```

```
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
```

Задача 3*

1. Сформировать два вектора: первый длиной 15 элементов со значениями от 0 до 1,4 с шагом 0,1; второй – длиной 6 элементов со значениями от 1 до 6.
2. Преобразовать первый вектор в матрицу 5x3, а второй – в матрицу 3x2. Выполнить перемножение этих матриц. Сами матрицы и результат их перемножения вывести на экран.

```
vec01 = np.arange(start = 0, stop = 1.5, step = 0.1, dtype="float")
```

```
print(vec01)
vec02 = np.linspace(start = 1, stop = 6, num = 6, dtype="int")
print(vec02)
```

```
[0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.  1.1 1.2 1.3 1.4]
[1 2 3 4 5 6]
```

```
matrix53 = vec01.reshape(5, 3)
print(matrix53)
matrix32 = vec02.reshape(3, 2)
print(matrix32)
print(np.dot(matrix53, matrix32))
```

```
[[0.  0.1 0.2]
 [0.3 0.4 0.5]
 [0.6 0.7 0.8]
 [0.9 1.  1.1]
 [1.2 1.3 1.4]]
[[1 2]
 [3 4]
 [5 6]]
[[ 1.3  1.6]
 [ 4.   5.2]
 [ 6.7  8.8]
 [ 9.4 12.4]
 [12.1 16. ]]
```

Задача 4*

1. Сформировать вектор из целых чисел (тип элементов должен быть int32) размерностью 36 и значениями 1, 3, 5, 7, … и так далее.
2. Выделить из этого вектора все элементы со значениями кратными 3. Вывести результат на экран (исходный вектор и с кратными числами).

```
vec36 = np.arange(start = 1, stop = 72, step = 2, dtype="int32")
vec36mod3 = vec36[vec36 % 3 == 0]
print(vec36)
print(vec36mod3)
```

```
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
 49 51 53 55 57 59 61 63 65 67 69 71]
[ 3  9 15 21 27 33 39 45 51 57 63 69]
```