

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Основы работы с библиотекой NumPy»

**Отчет по лабораторной работе № 2 по дисциплине «Технологии
распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1

Гребенкин Е. А. « » 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____ (подпись)

Цель работы: исследовать базовые возможности библиотеки NumPy языка программирования Python.

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).
3. Выполните клонирование созданного репозитория на рабочий компьютер.
4. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.
6. Проработать примеры лабораторной работы.
7. Решить задания в ноутбуках, выданных преподавателем.
8. Создать ноутбук, в котором выполнить решение индивидуального задания. Ноутбук должен содержать условие индивидуального задания. При решении индивидуального задания не должны быть использованы условный оператор if, а также операторы циклов while и for, а только средства библиотеки NumPy. Привести в ноутбуке обоснование принятых решений. Номер варианта индивидуального задания необходимо уточнить у преподавателя.
9. Зафиксируйте сделанные изменения в репозитории.
10. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики,

статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем.

11. Зафиксируйте сделанные изменения в репозитории.

12. Выполните слияние ветки для разработки с веткой main (master).

13. Отправьте сделанные изменения на сервер GitHub.

Проработка примеров:

```
In [88]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

In [89]: m = np.matrix('1, 2, 3, 4; 5, 6, 7, 8; 9, 1, 5, 7')

In [90]: m
Out[90]: matrix([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 1, 5, 7]])
```

Рисунок 1 – Пример 1

```
In [91]: m[1, 0]
Out[91]: 5

In [92]: m[1, :]
Out[92]: matrix([[5, 6, 7, 8]])
```

Рисунок 2 – Пример 2

```
In [93]: m[:, 2]
Out[93]: matrix([[3],
                [7],
                [5]])

In [94]: m[1, 2:]
Out[94]: matrix([[7, 8]])

In [95]: m[0:2, 1]
Out[95]: matrix([[2],
                [6]])
```

Рисунок 3 – Пример 3

```
In [96]: m[0:2, 1:3]
Out[96]: matrix([[2, 3],
                [6, 7]])

In [97]: cols = [0, 1, 3]
         m[:, cols]
Out[97]: matrix([[1, 2, 4],
                [5, 6, 8],
                [9, 1, 7]])

In [98]: type(m)
Out[98]: numpy.matrix
```

Рисунок 4 – Пример 4

```
In [99]: m = np.array(m)
In [100]: m
Out[100]: array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 1, 5, 7]])

In [101]: m.shape
Out[101]: (3, 4)
```

Рисунок 5 – Пример 5

```
In [110]: nums = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
In [111]: letters = np.array(['a', 'b', 'c', 'd', 'e', 'a', 'e', 'b'])
In [112]: less_than_5 = nums < 5
In [113]: less_than_5
Out[113]: array([ True,  True,  True,  True, False, False, False, False, False,
                False])
```

Рисунок 6 – Пример 6

```
In [114]: pos_a = letters == 'a'
In [115]: pos_a
Out[115]: array([ True, False, False, False, False,  True, False, False])

In [116]: nums = nums[less_than_5]
In [117]: nums
Out[117]: array([1, 2, 3, 4])

In [118]: mod_m = np.logical_and(m >= 3, m <= 7)
In [119]: mod_m
Out[119]: array([[False, False,  True,  True],
                [ True,  True,  True, False],
                [False, False,  True,  True]])
```

Рисунок 7 – Пример 7

```

In [125]: m[m > 7] = 25
In [126]: m
Out[126]: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7, 25],
                 [25,  1,  5,  7]])

In [127]: np.arange(10)
Out[127]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [128]: np.arange(5, 12)
Out[128]: array([ 5,  6,  7,  8,  9, 10, 11])

```

Рисунок 8 – Пример 8

Индивидуальные задания:

Индивидуальное задание №1

4. Дана целочисленная квадратная матрица. Определить:

- произведение элементов в тех строках, которые не содержат отрицательных элементов;
- максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

```
In [64]: import numpy as np
```

Дана целочисленная квадратная матрица. Определить: произведение элементов в тех строках, которые не содержат отрицательных элементов:

```
In [57]: N = 5
```

```
In [58]: array = (np.random.rand(N, N) * 10).astype(int)
```

```
In [59]: array
```

```
Out[59]: array([[1, 8, 5, 2, 8],
                [4, 2, 5, 7, 9],
                [6, 1, 4, 2, 6],
                [0, 2, 6, 1, 1],
                [6, 4, 1, 4, 4]])
```

```
In [60]: array[0][0] = 0
```

```
In [61]: array
```

```
Out[61]: array([[0, 8, 5, 2, 8],
                [4, 2, 5, 7, 9],
                [6, 1, 4, 2, 6],
                [0, 2, 6, 1, 1],
                [6, 4, 1, 4, 4]])
```

```
In [62]: array[(array > 0).all(axis=1)].prod(axis=1)
```

```
Out[62]: array([2520, 288, 384])
```

Максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы:

```
In [63]: max([np.diagonal(array, offset=i).sum() for i in range(-N + 1, N)])
```

```
Out[63]: 18
```

Рисунок 9 – Решение задач

Контрольные вопросы

1. Каково назначение библиотеки NumPy?

Библиотека NumPy – это библиотека ЯП Python, добавляющая поддержку больших многомерных массивов и матриц, свойства и методы для работы с ними.

2. Что такое массивы ndarray?

Ndarray – основной элемент библиотеки NumPy. Означает N-мерный массив. Этот объект является однородным с заранее известным количеством элементов.

3. Как осуществляется доступ к частям многомерного массива?

Предположим есть массив $m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

*[1,2,3,]
[4,5,6,]
[7,8,9,]*

- *Для выделения одного элемента нужно написать $m[i, j]$, например $m[1,2] = 6$*
- *Для выделения строки нужно написать $m[i, :]$, например, $m[1, :] = [[4,5,6]]$*
- *Для выделения колонки нужно написать $m[:, j]$, например, $m[:, 2] = [[3], [6], [9]]$*
- *Для выделения части матрицы нужно написать $m[i, j:]/m[i:,j]/m[i:,j:]$, например $m[1, 1:] = [[5, 6]]$*

4. Как осуществляется расчёт статистик по данным?

Для этих целей используют следующие методы:

- *argmax - Индексы элементов с максимальным значением (по осям)*
- *argmin - Индексы элементов с минимальным значением (по осям)*
- *max - Максимальные значения элементов (по осям)*
- *min - Минимальные значения элементов (по осям)*
- *mean - Средние значения элементов (по осям)*
- *prod - Произведение всех элементов (по осям)*
- *std - Стандартное отклонение (по осям)*
- *sum - Сумма всех элементов (по осям)*
- *var - Дисперсия (по осям)*

5. Как выполняется выборка данных из массива ndarray?

Используя Boolean выражение, которое используется для индексации массивов, например

```
m = np.matrix([[0.555, 0.324, 0.123], [0.987, 0.213, 0.543], [0.5435, 0.346, 0.324]])  
m[m > 0.5]  
>>> matrix([[0.555, 0.987, 0.543, 0.5435]])  
# Так же можно модифицировать матрицу, используя where  
np.where(m > 0.5, True, False)  
>>> matrix([[True, False, False, True, False, True, True, False]])
```