

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное  
учреждение высшего образования**  
**«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**  
**Кафедра инфокоммуникаций**  
**Институт цифрового развития**

**ОТЧЁТ**

по лабораторной работе №3.2

Дисциплина: «Программирование на Python»

Тема: «Основы работы с библиотекой NumPy»

Выполнила: студент 2 курса,  
группы ИВТ-б-о-21-1  
Богдан Александр Анатольевич

Ставрополь 2023

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

**Практическая часть:**

1. Создал 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).

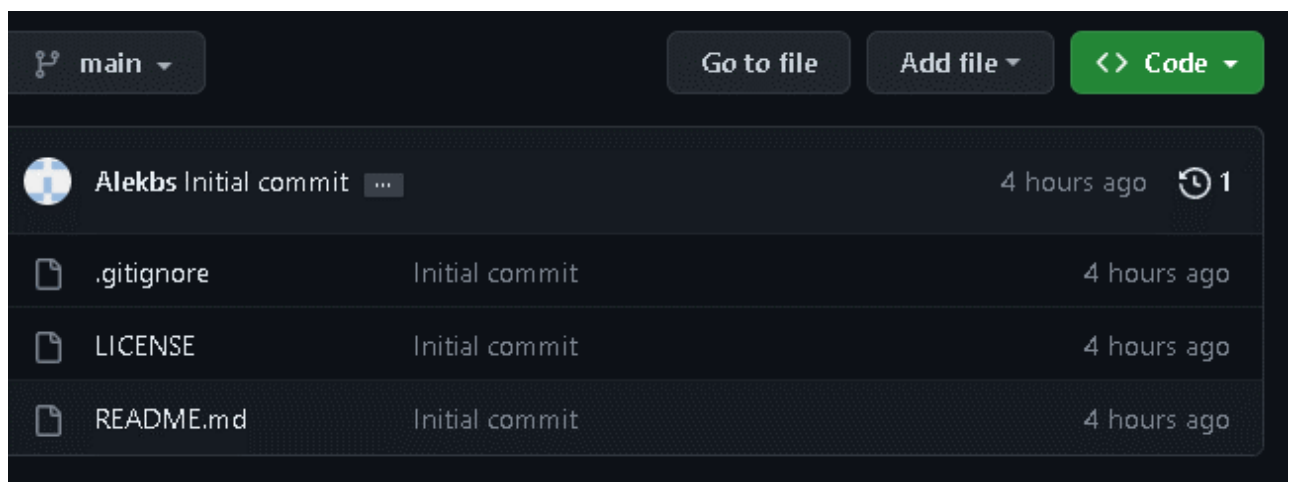


Рисунок 1. Создание репозитория

2. Выполнил клонирование созданного репозитория на рабочий компьютер.

3. Организовал свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\student-09-331\Downloads\Lab_3.2>git flow init
Which branch should be used for bringing forth production releases?
- develop
- main
Branch name for production releases: [main] main
Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop
How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] pre
Hooks and filters directory? [C:/Users/student-09-331/Downloads/Lab_3.2/.git/hooks] hook
```

Рисунок 2. Организация репозитория в соответствии с git-flow

4. Дополнил файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
pip-wheel-metadata/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
```

Рисунок 3. Изменение .gitignore

5. Проработал примеры лабораторной работы.

## 6. Решить задания в ноутбуках, выданных преподавателем.

```
In [6]: # основная структура данных - массив
a = np.array([1, 2, 3, 4, 5])
b = np.array([0.1, 0.2, 0.3, 0.4, 0.5])

print("a =", a)
print("b =", b)
```

```
a = [1 2 3 4 5]
b = [0.1 0.2 0.3 0.4 0.5]
```

Создайте массив с 5 любыми числами:

```
In [9]: a = np.array([1, 2, 3, 4, 5])

print(a)
```

```
[1 2 3 4 5]
```

Арифметические операции, в отличие от операций над списками, применяются поэлементно:

```
In [8]: list1 = [1, 2, 3]
array1 = np.array([1, 2, 3])

print("list1:", list1)
print('\tlist1 * 3:', list1 * 3)
print('\tlist1 + [1]:', list1 + [1])

print('array1:', array1)
print('\tarray1 * 3:', array1 * 3)
print('\tarray1 + 1:', array1 + 1)
```

```
list1: [1, 2, 3]
list1 * 3: [1, 2, 3, 1, 2, 3, 1, 2, 3]
list1 + [1]: [1, 2, 3, 1]
array1: [1 2 3]
array1 * 3: [3 6 9]
array1 + 1: [2 3 4]
```

Создайте массив из 5 чисел. Возведите каждый элемент массива в степень 3

## Рисунок 4. Решение ноутбуков преподавателя

возвращаемые значения -- массивы, содержащие результаты вычисления для каждого элемента (True -- да или False -- нет).

```
In [14]: print("a =", a)
print("\ta > 1: ", a > 1)
print("\nb =", b)
print("\tb < 0.5: ", b < 0.5)

print("\nОдновременная проверка условий:")
print("\t(a > 1) & (b < 0.5): ", (a > 1) & (b < 0.5))
print("\tА вот это проверяет, что a > 1 ИЛИ b < 0.5: ", (a > 1) | (b < 0.5))
```

```
a = [10 20 30 40 50]
a > 1: [ True  True  True  True  True]

b = [ 1  5 10  8  5]
b < 0.5: [False False False False False]
```

Одновременная проверка условий:  
(a > 1) & (b < 0.5): [False False False False False]  
А вот это проверяет, что a > 1 ИЛИ b < 0.5: [ True True True True True]

Создайте 2 массива из 5 элементов. Проверьте условие "Элементы первого массива меньше 6, элементы второго массива делятся на 3"

```
In [18]: mas1 = np.array([1, 3, 5, 7, 8])
mas2 = np.array([1, 3, 6, 8, 10])

print("mas1 =", mas1)
print("\tmas1 > 6: ", mas1 > 6)
print("mas1 =", mas1)
print("\tmas2 % 3 == 0: ", mas2 % 3 == 0)

mas1 = [1 3 5 7 8]
mas1 > 6: [False False False  True  True]
mas1 = [1 3 5 7 8]
mas2 % 3 == 0: [False  True  True False False]
```

Теперь проверьте условие "Элементы первого массива делятся на 2 или элементы второго массива больше 2"

## Рисунок 5. Решение ноутбуков преподавателя

```
In [7]: import numpy as np

m1 = np.array([2, 4, 6, 8, 10, 12])
m2 = np.array([ 7, 11, 15, 18, 23, 29])

print((m1+m2)**2)

[ 81 225 441 676 1089 1681]

2. Выведите все элементы из первого массива, индексы которых соответствуют индексам тех элементов второго массива, которые больше 12 и дают остаток 3 при делении на 5.

In [21]: print(m1[np.where((m2 > 12) & (m2 % 5 == 3))])

[ 8 10]

3. Проверьте условие "Элементы первого массива делятся на 4, элементы второго массива меньше 14". (Подсказка: в результате должен получиться массив с True и False)

In [23]: print((m1 % 4 == 0) & (m2 < 14))

[False  True  False  False  False  False]
```

## Рисунок 6. Решение ноутбуков преподавателя

### Задание №2

- Найдите интересный для вас датасет. Например, можно выбрать датасет тут: <http://data.un.org/Explorer.aspx> (выбираете датасет, жмете на view data, потом download, выбирайте csv формат)
- Рассчитайте подходящие описательные статистики для признаков объектов в выбранном датасете
- Проанализируйте и прокомментируйте содержательно получившиеся результаты
- Все комментарии оформляйте строго в ячейках формата markdown

```
In [23]: 1 import csv
          2 import numpy as np

In [24]: 1 with open ('organizations_gdp_hist.csv', 'r', newline='') as f:
          2     data=csv.reader(f, delimiter=',')
          3     total_gdp=[]
          4     gdp_variation=[]
          5     for row in data:
          6         if row[2]=="Miembros OCDE":
          7             total_gdp.append(float(row[6]))
          8             gdp_variation.append(float(row[8]))

In [25]: 1
          2 print(f"Среднее значение ВВП в Организации экономического сотрудничества и развития: {np.mean(total_gdp)}")
          3 print(f"Среднее значение изменения ВВП: {np.mean(gdp_variation)}")
          4 print(f"Среднее отклонение ВВП в Организации экономического сотрудничества и развития: {np.std(total_gdp)}")
          5 print(f"Среднее отклонение изменения ВВП: {np.std(gdp_variation)}")
          6 print(f"Коэффициент парной корреляции: {np.corrcoef(total_gdp, gdp_variation)}")

Среднее значение ВВП в Организации экономического сотрудничества и развития: 22238256040665.2
Среднее значение изменения ВВП: 2.9227095605094546
Среднее отклонение ВВП в Организации экономического сотрудничества и развития: 18335569068062.613
Среднее отклонение изменения ВВП: 2.00698391724925
Коэффициент парной корреляции: [[ 1.          -0.47053856]
 [-0.47053856  1.          ]]
```

## Рисунок 7. Решение ноутбуков преподавателя

7. 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 for, а только средства библиотеки NumPy. Привести в ноутбуке обоснование принятых решений. Номер варианта индивидуального задания необходимо уточнить у преподавателя.

## Индивидуальное задание:

Дана целочисленная квадратная матрица. Определить:

- сумму элементов в тех столбцах, которые не содержат отрицательных элементов;
- минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

### Индивидуальное задание

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

```
In [11]: 1 import numpy as np
2
3 m = int(input("Введите размерность матрицы:\nm = "))
4 n = int(input("n = "))
5 matrix = np.random.randint(-10, 20, size=(m, n))
```

Введите размерность матрицы:  
m = 5  
n = 3

```
In [12]: 1 print("Матрица:\n", matrix)
```

Матрица:  
[[-3 -4 3]  
 [-1 5 8]  
 [-3 -7 -5]  
 [-4 -6 -7]  
 [ 5 8 16]]

```
In [14]: 1
2 non_zero_columns = np.count_nonzero(np.prod(matrix, axis=0))
3 print("Количество столбцов без нулевых элементов:", non_zero_columns)
4
5 row_characteristics = np.sum(matrix * (matrix > 0) * (matrix % 2 == 0), axis=1)
6
7 sorted_indices = np.argsort(row_characteristics)
8
9 sorted_matrix = matrix[sorted_indices]
10 print("Отсортированная матрица:")
11 print(sorted_matrix)
12
```

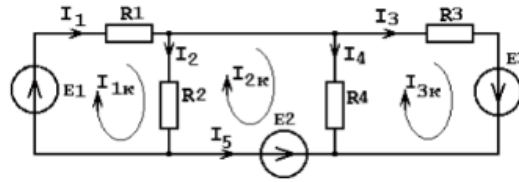
Количество столбцов без нулевых элементов: 3  
Отсортированная матрица:  
[[-3 -4 3]  
 [-3 -7 -5]  
 [-4 -6 -7]  
 [-1 5 8]  
 [ 5 8 16]]

Рисунок 8. Решение индивидуального задания 1

8. 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.

## Индивидуальное задание 2

Рассчитать методом контурных токов токи в ветвях для заданной электрической цепи с источниками ЭДС



```
In [4]: 1 import numpy as np
```

```
In [5]: 1 r1 = 100  
2 r2 = 300  
3 r3 = 150  
4 r4 = 250  
5 e1 = 10  
6 e2 = 20  
7 e3 = 15
```

Рисунок 9. Решение индивидуального задания 2

### Контрольные вопросы:

1. Каково назначение библиотеки NumPy?

numpy – это библиотека для языка программирования Python, которая предоставляет в распоряжение разработчика инструменты для эффективной работы с многомерными массивами и высокопроизводительные вычислительные алгоритмы.

2. Что такое массивы ndarray?

Объект ndarray (что значит N-размерный массив). Этот объект является многомерным однородным массивом с заранее заданным количеством элементов. Однородный — потому что практически все объекты в нем одного размера или типа. На самом деле, тип данных определен другим объектом NumPy, который называется dtype (тип-данных).

Каждый ndarray ассоциирован только с одним типом dtype.

### 3. Как осуществляется доступ к частям многомерного массива?

Иногда возникает задача взять не все элементы строки, а только часть: рассмотрим пример, когда нам из второй строки нужно извлечь все элементы, начиная с третьего.

```
>>> m[1, 2:]  
  
matrix([[7, 8]])
```

Запись 2: означает, что начиная с третьего столбца включительно (т.к. нумерация начинается с 0, то третий элемент имеет индекс 2) взять все оставшиеся в ряду элементы.

Аналогично предыдущему примеру, можно извлечь только часть столбца матрицы.

```
\>>> m[0:2, 1:3]  
  
matrix([[2, 3],  
[6, 7]])
```

Numpy позволяет извлекать произвольный набор столбцов или строк матрицы, строки (столбцы) которые нужно извлечь передаются в виде списка.

### 4. Как осуществляется расчет статистик по данным?

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)

#### 4. Как выполняется выборка данных из массивов ndarray?

Самым замечательным в использовании boolean массивов при работе с ndarray является то, что их можно применять для построения выборок.

Например:

```
>>> less_then_5 = nums < 5
```

```
>>> less_then_5
```

```
array([ True,  True,  True,  True, False, False, False, False, False, False])
```

Если мы переменную less\_then\_5 передадим в качестве списка индексов для nums, то получим массив, в котором будут содержаться элементы из nums с индексами равными индексам True позиций массива less\_then\_5.

**Вывод:** в результате выполнения работы были исследованы базовые возможности библиотеки NumPy языка программирования Python.