

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»**

**Факультет информационных систем и технологий**

**Кафедра: «Измерительно-вычислительные комплексы»**

**Дисциплина: «Методы искусственного интеллекта»**

**Отчет**

По лабораторной работе №2

по теме: **«Матрицы»**

Выполнил:  
студент гр. ИСТбд-42  
Тагашев И. И.

Проверил:  
к.т.н., доцент каф. ИВК  
Шишкин В. В.

Ульяновск  
2022 г.

## Лабораторная работа №2

**Цель работы:** изучение двумерных массивов в Python.

**Задание:**

С клавиатуры вводится два числа  $K$  и  $N$ . Квадратная матрица  $A(N,N)$ , состоящая из 4-х равных по размерам подматриц,  $B, C, D, E$  заполняется случайным образом целыми числами в интервале  $[-10,10]$ . Для отладки использовать не случайное заполнение, а целенаправленное. Вид матрицы  $A$ :

Для ИСТд-42:

E	B
D	C

Для простоты все индексы в подматрицах относительные. По сформированной матрице  $F$  (или ее частям) необходимо вывести не менее 3 разных графиков. Допускается использование библиотек `numpy` и `matplotlib`

### Вариант №17

Формируется матрица  $F$  следующим образом: скопировать в нее  $A$  и если в  $E$  количество нулей в нечетных столбцах, чем сумма чисел в нечетных строках, то поменять местами  $B$  и  $E$  симметрично, иначе  $C$  и  $E$  поменять местами несимметрично. При этом матрица  $A$  не меняется. После чего если определитель матрицы  $A$  больше суммы диагональных элементов матрицы  $F$ , то вычисляется выражение:  $A^{-1} * A^T - K * F^{-1}$ , иначе вычисляется выражение  $(A^{-1} + G - F^{-1}) * K$ , где  $G$ -нижняя треугольная матрица, полученная из  $A$ . Выводятся по мере формирования  $A$ ,  $F$  и все матричные операции последовательно.

### Теоретическая справка:

Матрицами называются массивы элементов, представленные в виде прямоугольных таблиц, для которых определены правила математических действий. Элементами матрицы могут являться числа, алгебраические символы или математические функции.

Для работы с матрицами в Python также используются списки. Каждый элемент списка-матрицы содержит вложенный список.

Таким образом, получается структура из вложенных списков, количество которых определяет количество столбцов матрицы, а число элементов внутри каждого вложенного списка указывает на количество строк в исходной матрице.

## Ход работы:

### 1. Листинг программы:

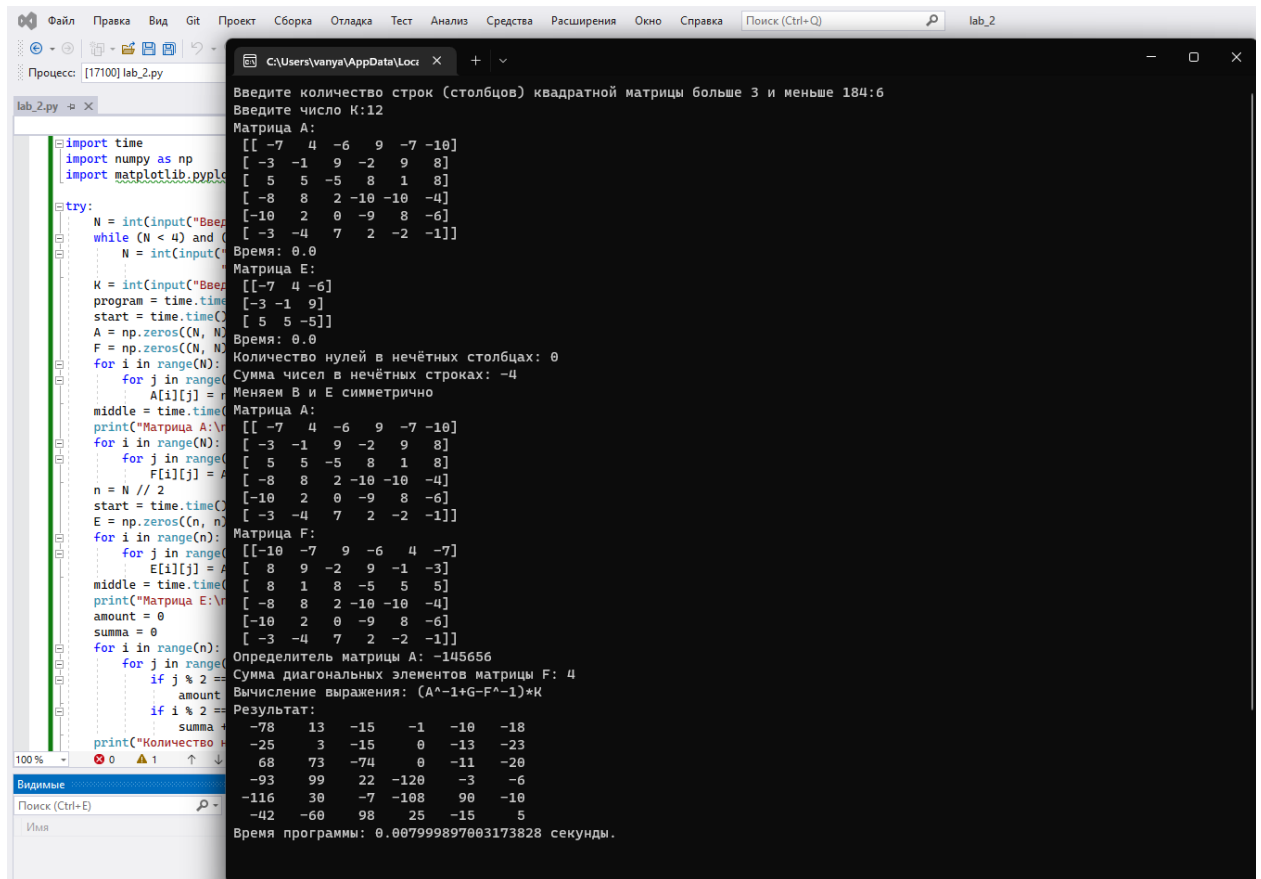
```
1. import time
2. import numpy as np
3. import matplotlib.pyplot as plt
4.
5. try:
6.     N = int(input("Введите количество строк (столбцов)
    квадратной матрицы больше 3 и меньше 184:"))
7.     while (N < 4) and (N > 183):
8.         N = int(input("Вы ввели неверное число. "
    "\nВведите количество строк (столбцов)
    квадратной матрицы больше 3 и меньше 184:"))
9.     K = int(input("Введите число K:"))
10.    program = time.time()
11.    start = time.time()
12.    A = np.zeros((N, N), dtype=int)
13.    F = np.zeros((N, N), dtype=int)
14.    for i in range(N):      # Формируем матрицу A
15.        for j in range(N):
16.            A[i][j] = np.random.randint(-10, 10)
17.    middle = time.time()
18.    print("Матрица A:\n", A, "\nВремя:", middle - start)
19.    for i in range(N):      # Формируем матрицу F, копируя из
    матрицы A
20.        for j in range(N):
21.            F[i][j] = A[i][j]
22.    n = N // 2              # Размерность подматрицы
23.    start = time.time()
24.    E = np.zeros((n, n), dtype=int)    # Формируем матрицу E
25.    for i in range(n):
26.        for j in range(n):
27.            E[i][j] = A[i][j]
28.    middle = time.time()
29.    print("Матрица E:\n", E, "\nВремя:", middle - start)
30.    amount = 0
31.    summa = 0
32.    for i in range(n):
33.        for j in range(n):
34.            if j % 2 == 0 and E[i][j] == 0:    # Количество 0 в
    нечетных столбцах
35.                amount += 1
36.            if i % 2 == 0:    # Сумма элементов в нечетных
    строках
37.                summa += E[i][j]
38.    print("Количество нулей в нечётных столбцах:", amount,
    "\nСумма чисел в нечётных строках:", summa)
39.    if amount > summa:
40.        print("Меняем B и E симметрично")
41.        for i in range(n):    # B и E симметрично
```

```

43.         for j in range(n):
44.             F[i][j] = A[i][N-j-1]
45.             F[i][N-j-1] = A[i][j]
46.     else:
47.         print("Меняем C и E несимметрично")
48.         for i in range(n):           # C и E несимметрично
49.             for j in range(n):
50.                 F[i][j] = A[n + i][n + j]
51.                 F[n + i][n + j] = A[i][j]
52.         print("Матрица A:\n", A, "\nМатрица F:\n", F)
53.         print("Определитель матрицы A:", round(np.linalg.det(A)),
54.               "\nCумма диагональных элементов матрицы F:", np.trace(F))
55.         if np.linalg.det(A) == 0 or np.linalg.det(F) == 0:
56.             print("Нельзя вычислить т.к. матрица A или F
57.               вырождена")
58.             elif np.linalg.det(A) > np.trace(F):
59.                 print("Вычисление выражения: A^-1*A^T-K*F^-1")
60.                 A = np.dot(np.linalg.inv(A), np.transpose(A)) -
61.                   (np.linalg.inv(F) * K) # A^-1*A^T-K*F^-1
62.             else:
63.                 print("Вычисление выражения: (A^-1+G-F^-1)*K")
64.                 A = (np.linalg.inv(A) + np.tril(A) - np.linalg.inv(F))
65.                 * K # (A^-1+G-F^-1)*K
66.             print("Результат:")
67.             for i in A:           # Вывод результата
68.                 for j in i:
69.                     print("%5d" % round(j), end=' ')
70.                 print()
71.             finish = time.time()
72.             result = finish - program
73.             print("Время программы: " + str(result) + " секунды.")\
74.
75.             plt.plot(F)           # График 1.
76.             plt.show()
77.
78.             for i in range(0, n):   # График 2.
79.                 for j in range(0, n):
80.                     plt.bar(i, F[i][j])
81.             plt.show()
82.
83.             x = np.arange(0, n, 1)   # График 3.
84.             f0 = F[0][0]
85.             a0 = A[0][0]
86.             labels = ["F[0]", "A[0]"]
87.             fig, ax = plt.subplots()
88.             ax.stackplot(x, f0, a0, labels=labels)
89.             ax.legend(loc='upper left')
90.             plt.show()
91.
92. except ValueError:
93.     print("\nЭто не число")

```

## 2. Тестирование программы:



The screenshot shows a Python IDE with a file named `lab_2.py`. The code defines a function `try:` that takes inputs `N` and `K`, creates matrices `A`, `E`, and `F`, and performs various calculations including matrix multiplication and determinant calculation. The output window shows the execution results for `N=12` and `K=12`.

```
Введите количество строк (столбцов) квадратной матрицы больше 3 и меньше 184:6
Введите число K:12
Матрица A:
[[ -7  4 -6  9 -7 -10]
 [ -3 -1  9 -2  9  8]
 [  5  5 -5  8  1  8]
 [ -8  8  2 -10 -10 -4]
 [-10  2  0 -9  8 -6]
 [ -3 -4  7  2 -2 -1]]
Время: 0.0
Матрица E:
[[ -7  4 -6]
 [ -3 -1  9]
 [  5  5 -5]]
Время: 0.0
Количество нулей в нечётных столбцах: 0
Сумма чисел в нечётных строках: -4
Меняем B и E симметрично
Матрица A:
[[ -7  4 -6  9 -7 -10]
 [ -3 -1  9 -2  9  8]
 [  5  5 -5  8  1  8]
 [ -8  8  2 -10 -10 -4]
 [-10  2  0 -9  8 -6]
 [ -3 -4  7  2 -2 -1]]
Матрица F:
[[ -10 -7  9 -6  4 -7]
 [  8  9 -2  9 -1 -3]
 [  8  1  8 -5  5  5]
 [ -8  8  2 -10 -10 -4]
 [-10  2  0 -9  8 -6]
 [ -3 -4  7  2 -2 -1]]
Определитель матрицы A: -145656
Сумма диагональных элементов матрицы F: 4
Вычисление выражения: (A^2-1+G-F^2-1)*K
Результат:
-78  13 -15 -1 -10 -18
-25  3 -15  0 -13 -23
 68 73 -74  0 -11 -20
-93 99  22 -120 -3 -6
-116 30 -7 -108 90 -10
-42 -60 98  25 -15  5
Время программы: 0.007999897003173828 секунды.
```

Рис. 1 — «Ввод исходных данных N и K»

**Matplotlib** — это библиотека на языке Python для визуализации данных. В ней можно построить двумерные (плоские) и трехмерные графики.

**NumPy** это open-source модуль для python, который предоставляет общие математические и числовые операции в виде пре-компилированных, быстрых функций.

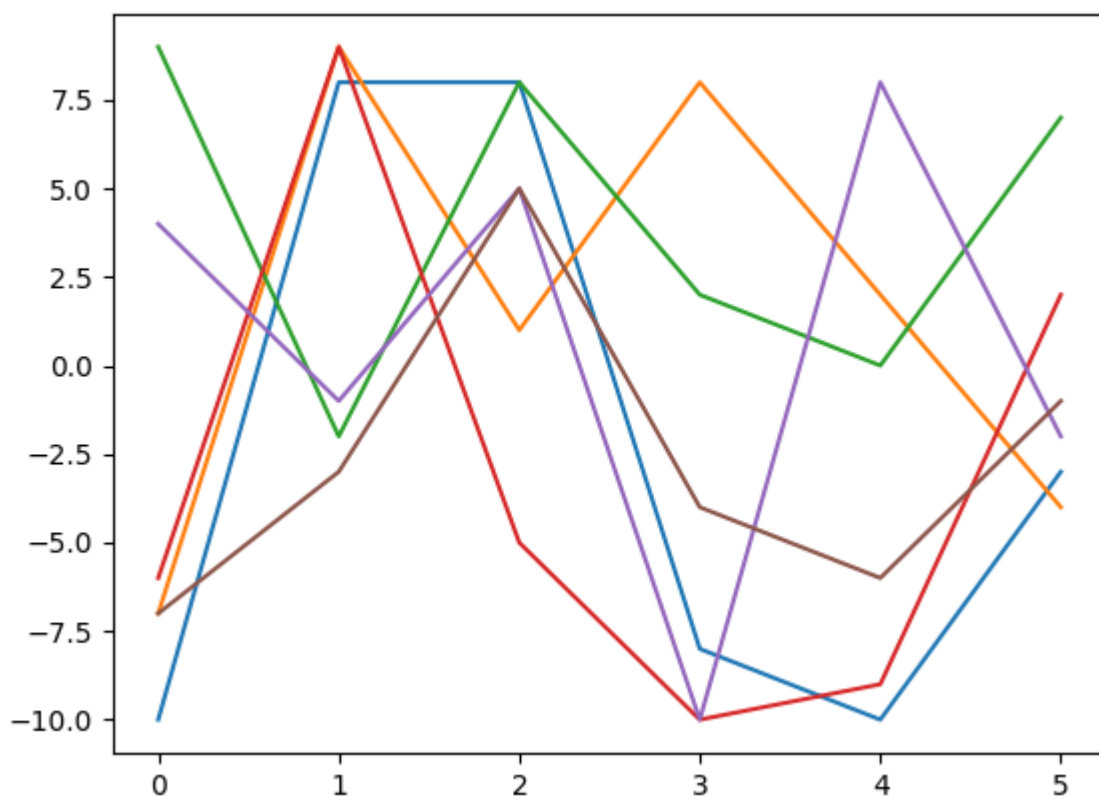


Рис. 2 – «Построение линейного графика»

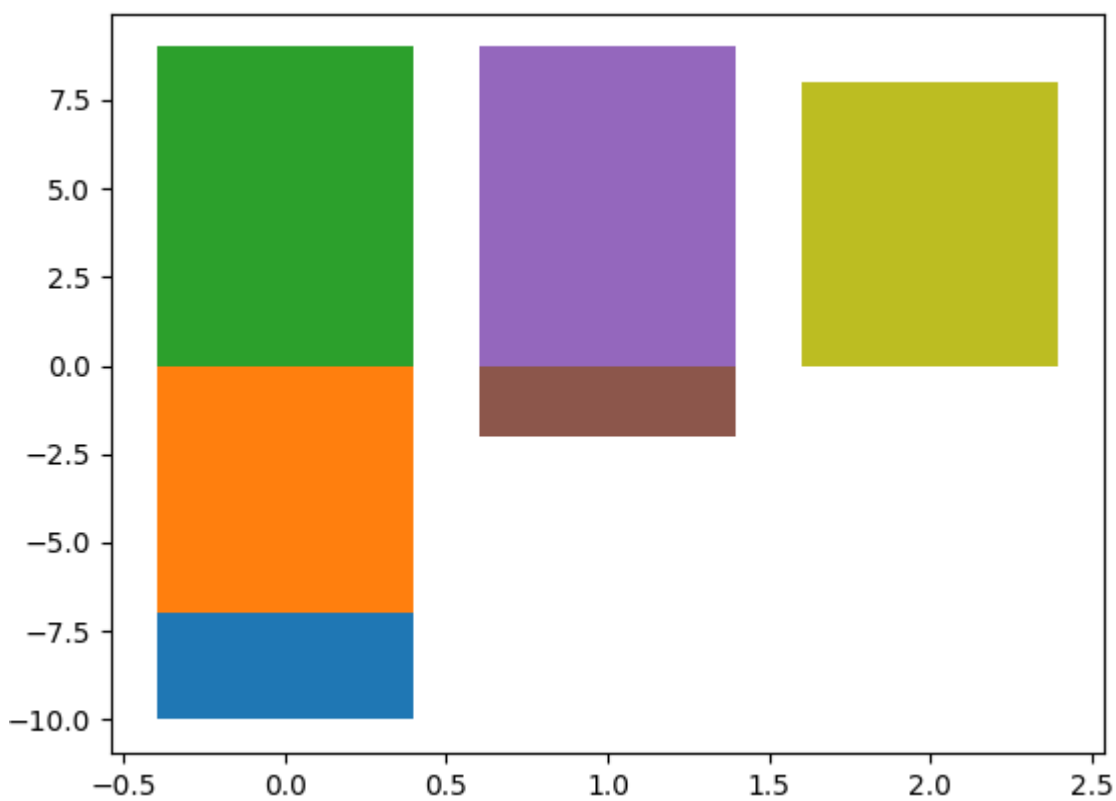


Рис. 3 – «Построение диаграммы для категориальных данных»

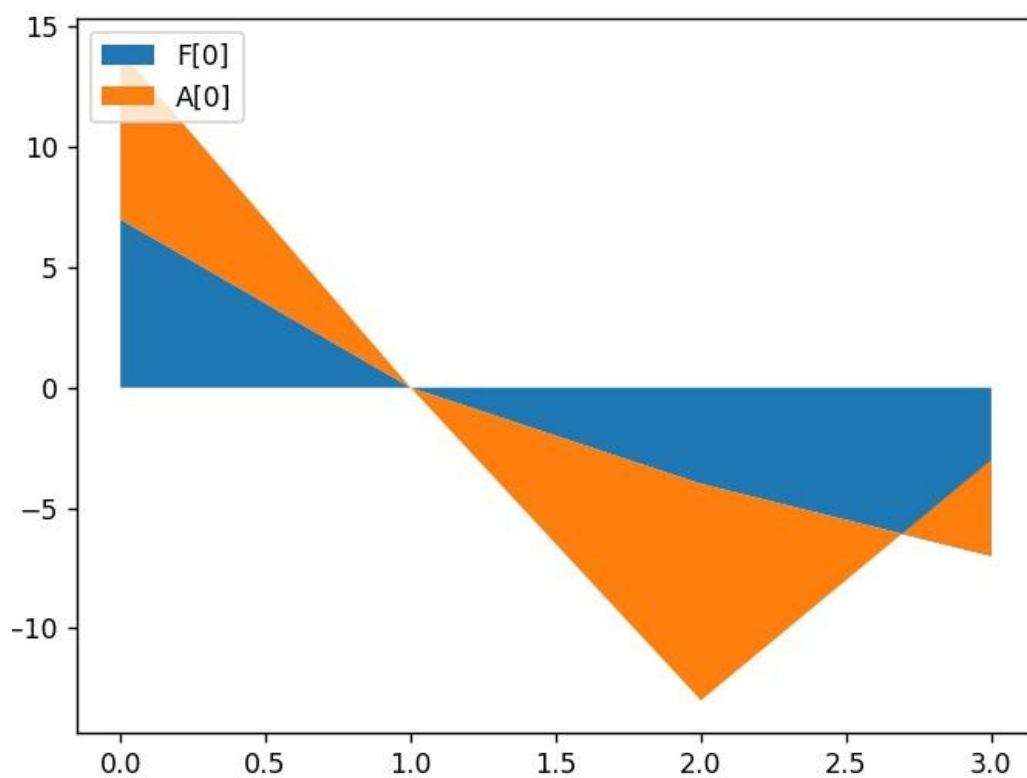


Рис. 4 – «Построение стекового графика»

**Вывод по работе:** в ходе выполнения лабораторной работы №2 «Массивы», были получены навыки работы с двумерными массивами.