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 mathplotlib

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:

```
import time
import numpy as np
import matplotlib.pyplot as plt

try:
    N = int(input("Введите количество строк (столбцов) квадратной матрицы больше 3 и меньше 184:"))
    while (N < 4) and (N > 183):
        N = int(input("Вы ввели неверное число. "
                      "\nВведите количество строк (столбцов) квадратной матрицы больше 3 и меньше 184:"))
    K = int(input("Введите число K:"))
    program = time.time()
    start = time.time()
    A = np.zeros((N, N), dtype=int)
    F = np.zeros((N, N), dtype=int)
    for i in range(N):     # Формируем матрицу A
        for j in range(N):
            A[i][j] = np.random.randint(-10, 10)
    middle = time.time()
    print("Матрица A:\n", A, "\nВремя:", middle - start)
    for i in range(N):      # Формируем матрицу F, копируя из матрицы A
        for j in range(N):
            F[i][j] = A[i][j]
    n = N // 2          # Размерность подматрицы
    start = time.time()
    E = np.zeros((n, n), dtype=int)   # Формируем матрицу E
    for i in range(n):
        for j in range(n):
            E[i][j] = A[i][j]
    middle = time.time()
    print("Матрица E:\n", E, "\nВремя:", middle - start)
    amount = 0
    summa = 0
    for i in range(n):
        for j in range(n):
            if j % 2 == 0 and E[i][j] == 0:   # Количество 0 в нечетных столбцах
                amount += 1
            if i % 2 == 0:    # Сумма элементов в нечетных строках
                summa += E[i][j]
    print("Количество нулей в нечётных столбцах:", amount, "\nСумма чисел в нечётных строках:", summa)
    if amount > summa:
        print("Меняем B и E симметрично")
        for i in range(n):       # B и E симметрично
```

```
            for j in range(n):
                F[i][j] = A[i][N-j-1]
                F[i][N-j-1] = A[i][j]
    else:
        print("Меняем C и E несимметрично")
        for i in range(n):      # C и E несимметрично
            for j in range(n):
                F[i][j] = A[n + i][n + j]
                F[n + i][n + j] = A[i][j]
    print("Матрица A:\n", A, "\nМатрица F:\n", F)
    print("Определитель матрицы A:", round(np.linalg.det(A)),
"\nСумма диагональных элементов матрицы F:", np.trace(F))
    if np.linalg.det(A) == 0 or np.linalg.det(F) == 0:
        print("Нельзя вычислить т.к. матрица A или F
вырождена")
    elif np.linalg.det(A) > np.trace(F):
        print("Вычисление выражения: A^-1*A^T-K*F^-1")
        A = np.dot(np.linalg.inv(A), np.transpose(A)) -
(np.linalg.inv(F) * K)  # A^-1*A^T-K*F^-1
    else:
        print("Вычисление выражения: (A^-1+G-F^-1)*K")
        A = (np.linalg.inv(A) + np.tril(A) - np.linalg.inv(F))
* K   # (A^-1+G-F^-1)*K
    print("Результат:")
    for i in A:          # Вывод результата
        for j in i:
            print("%5d" % round(j), end=' ')
        print()
    finish = time.time()
    result = finish - program
    print("Время программы: " + str(result) + " секунды.")\

    plt.plot(F)                 # График 1.
    plt.show()

    for i in range(0, n):       # График 2.
        for j in range(0, n):
            plt.bar(i, F[i][j])
    plt.show()

    x = np.arange(0, n, 1)      # График 3.
    f0 = F[0][0]
    a0 = A[0][0]
    labels = ["F[0]", "A[0]"]
    fig, ax = plt.subplots()
    ax.stackplot(x, f0, a0, labels=labels)
    ax.legend(loc='upper left')
    plt.show()

except ValueError:
    print("\nЭто не число")
```

2. *Тестирование программы:*

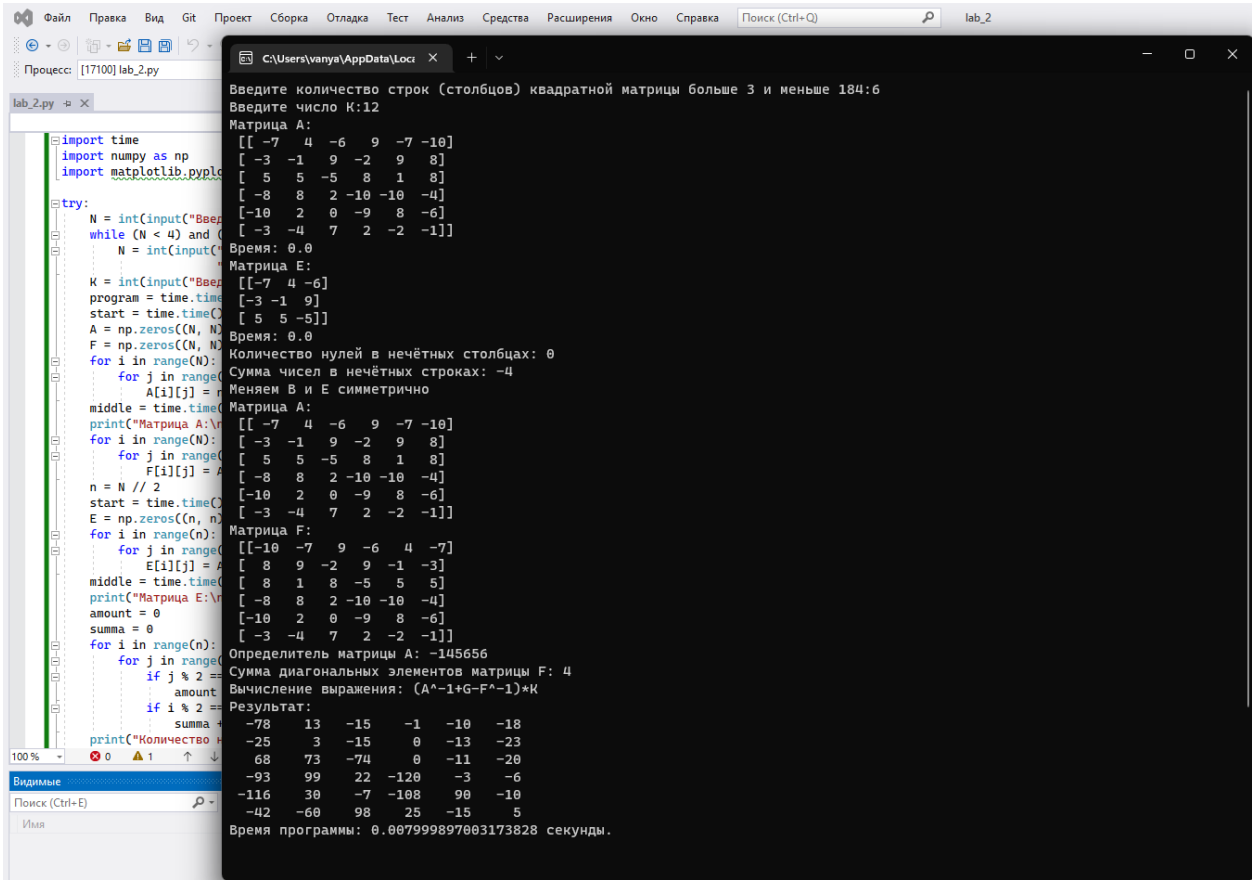

Рис. 1 – «Ввод исходных данных N и K»

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-скомпилированных, быстрых функций.

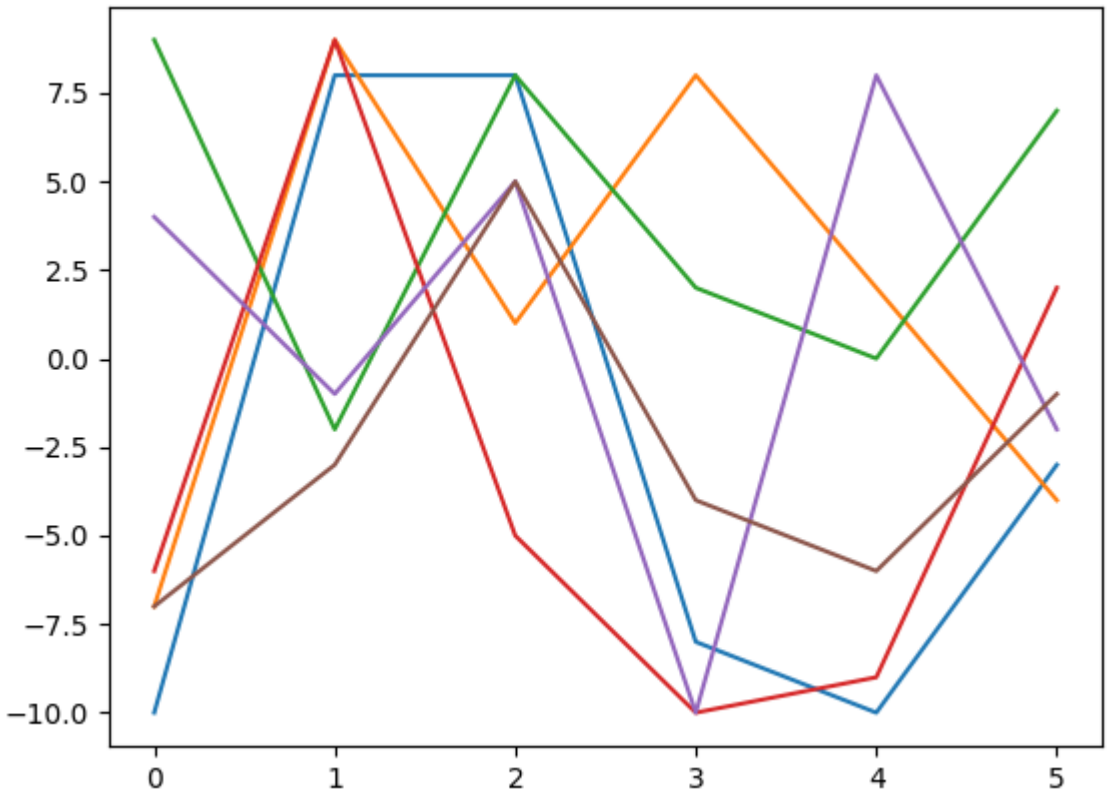

Рис. 2 – «Построение линейного графика»

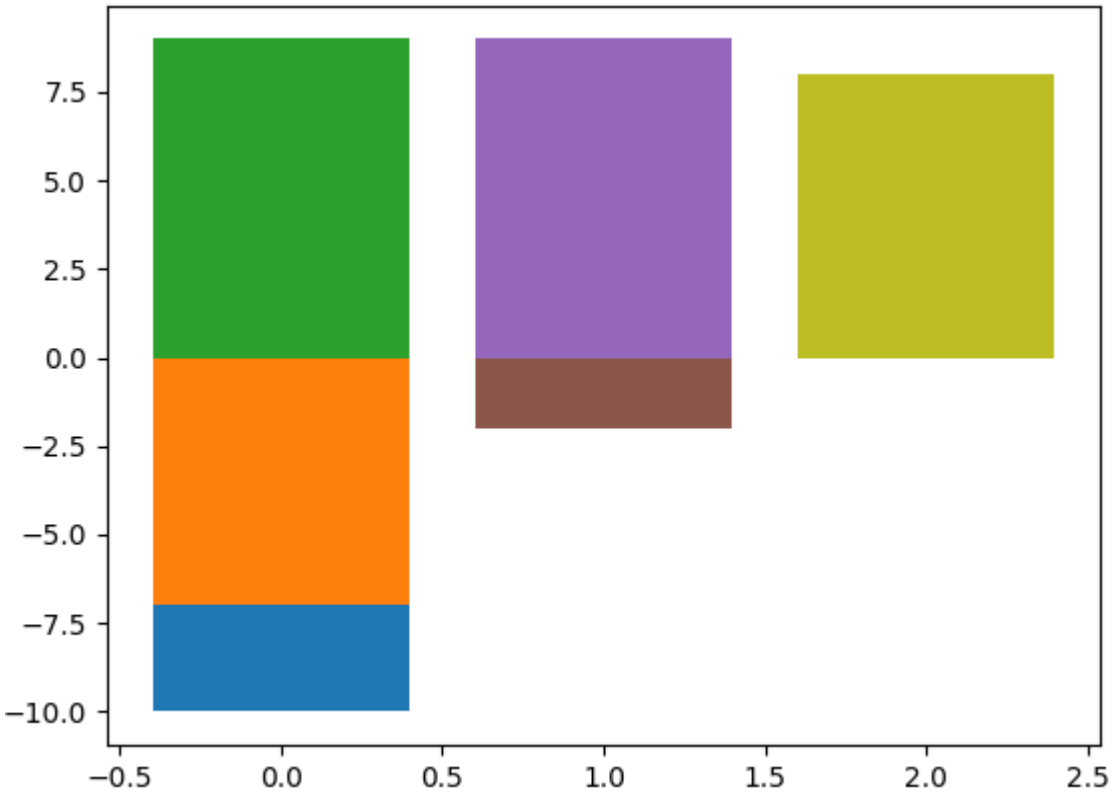

Рис. 3 – «Построение диаграммы для категориальных данных»

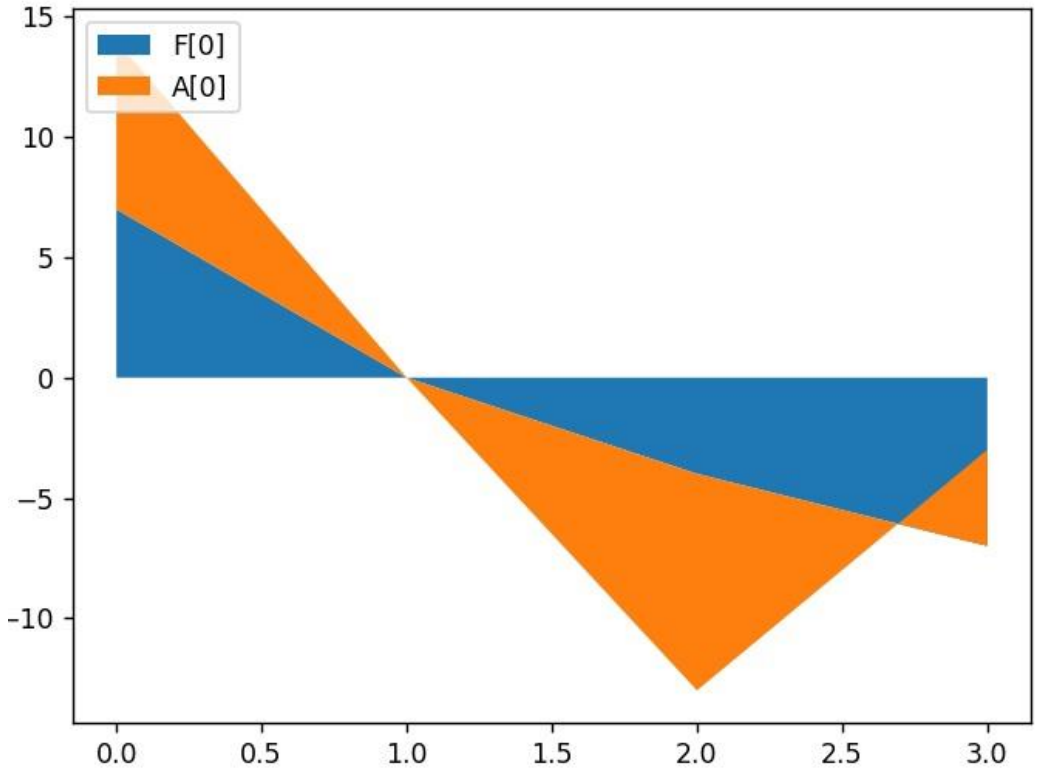

Рис. 4 – «Построение стекового графика»

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