Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №2**

Дисциплина: «Разработка профессиональных приложений»

Двумерные структуры данных

Вариант №8

Выполнил:

студент группы ИВТАСбд-22

Кириллов Д.А.

Проверил:

преподаватель кафедры

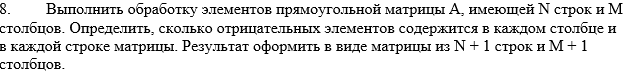
«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Постановка задачи**

Написать программу на языке python, выполняющую обработку данных по варианту



Ввод элементов списка должен быть доступен путем автоматической генерации. Необходимо использовать библиотеку numpy. Результаты выполнения должны сохраняться в файл (исходные данные и результат обработки).

**Описание реализации**

Создание матрицы и ее заполнение осуществляется в отдельной функции. Заполнение происходит путем автоматической генерации чисел. Используется библиотека numpy для работы с различными матрицами. В программе будет реализовано три матрицы: начальная, матрица с нулями, куда будет добавлена начальная матрица, и исходная матрица, в которую будет добавлена вторая матрица с добавленными строкой и столбцом, где будут написано количество отрицательных элементов в каждой строке и столбце. Вывод осуществляется в текстовый файл в отдельной функции.

**Описание возникших трудностей**

Возникали проблемы с выходом за границы массива, когда нынешний алгоритм еще не был реализован, как сейчас. Также было не понятно, как правильно заполнять строчку и столбец числом отрицательных чисел.

**Описание альтернативных способов решения**

Можно было попробовать реализовать программу путем задания списка, а не матрицы. И уже в дальнейшем выполнять с ним другие функции.

**Приложение**

import numpy as np  
  
  
# создаем матрицу случайных чисел  
def create\_matrix():  
 random\_matrix = np.matrix(np.random.randint(-100, 100, size=(rows, cols)))  
 return random\_matrix  
  
  
# создаем результирующую матрицу  
def create\_result(random\_matrix):  
 # создаем матрицу из нулей, куда копируем random\_matrix  
 b = np.append(random\_matrix, np.zeros([len(random\_matrix), 1], dtype=int), axis=1)  
 # c1 = np.hstack((random\_matrix, np.zeros((random\_matrix.shape[0], 1))))  
 result = np.r\_[b, [np.array([0] \* (cols + 1))]]  
  
 for i in range(rows):  
 for j in range(cols):  
 if random\_matrix[i, j] < 0:  
 result[i, cols] += 1  
 result[rows, j] += 1  
  
 return result  
  
  
def pr():  
 # выводим исходные данные и результат в файл  
 random\_matrix = create\_matrix()  
 result\_matrix = create\_result(random\_matrix)  
 with open("lab2.txt", "w") as f:  
 f.write('Исходные данные' + '\n\r' + str(random\_matrix) + '\n\r' + 'Результат обработки' + '\n\r' + str(result\_matrix))  
 # print(create\_matrix())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 rows = int(input('rows='))  
 cols = int(input('cols='))  
 pr()