**ЛАБОРАТОРНАЯ РАБОТА №5**

**Двумерные массивы. Использование подпрограмм при работе с двумерными массивами.**

Цель работы: приобрести практические навыки работы с двумерными массивами.

**Содержание работы**

1. Разработал алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.

2. Набрал текст программы и отладил её в среде разработки PyCharm.

3. Оформил отчет о проделанной работе.

**Выполнение работы**

1. Вариант и условия задачи

Вариант 8. Выполнить обработку элементов прямоугольной матрицы a, имеющей n строк и m столбцов. Определить, сколько отрицательных элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из n+1 строк и m+1

столбцов.

1. Код программы и результат ее работы

Функция append\_row\_of\_negative\_values(matrix):

Назначение - Добавить количество отрицательных элементов в ряду

Входые данные - матрица

Возвращаемые данные - ничего, так как матрица изменяется по ссылке

Функция get\_array\_of\_negative\_values\_in\_columns(matrix):

Назначение - Посчитать количество отрицательных элементов в столбцах и записать их в массив

Входые данные - матрица

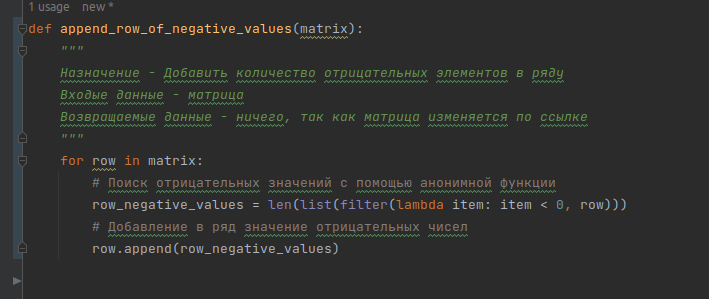
Возвращаемые данные - количество отрицательных элементов в столбцах в виде массива

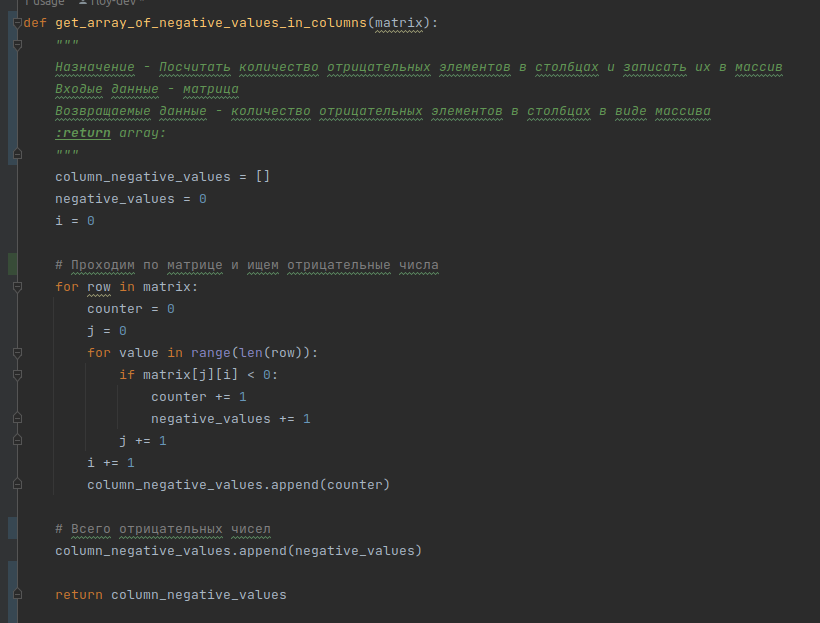
Функция get\_matrix\_filled\_by\_human(n, m):

Назначение - Заполнить и вывести матрицу по заданным параметрам n и m

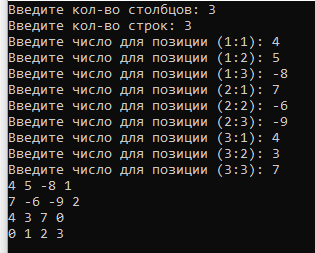
Входые данные - количества столбцов, количество строк

Возвращаемые данные - матрица заполненная человеком

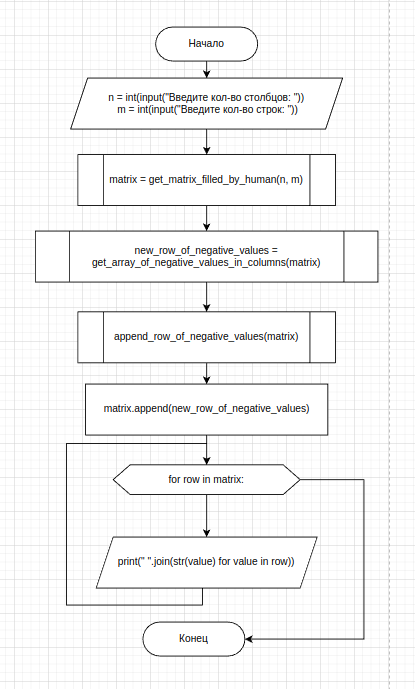


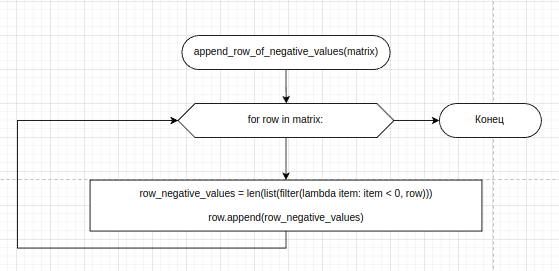


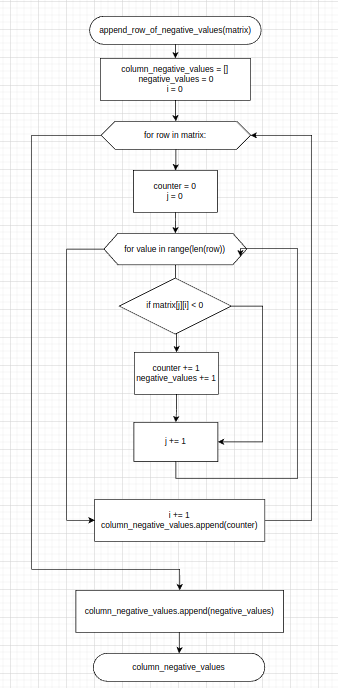


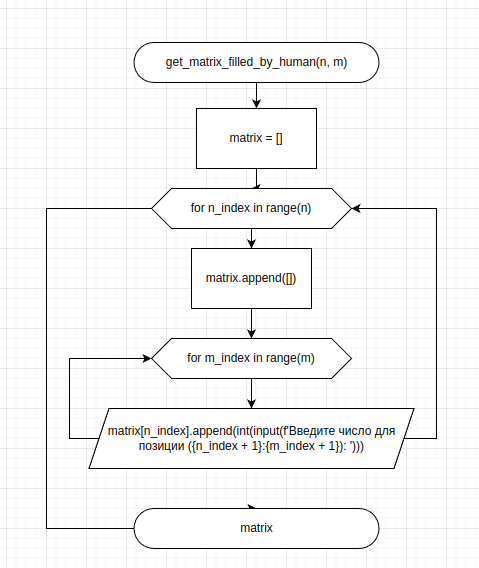


1. Блок схема









**Контрольные вопросы**

1. NumPy — библиотека с открытым исходным кодом для языка программирования Python. Возможности: поддержка многомерных массивов; поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами.
2. Один из способов формирования многомерного массива состоит в создании двумерного массива и соответствующего его расширения.
3. многомерные массивы хранятся в виде линейной последовательности своих компонент, и принципиальной разницы между одномерными и многомерными массивами в памяти ЭВМ нет.
4. Пример функции, генерирующей массив размера NxN

* np.eye(4) # матрица 4х4
* np.eye(4, 2) # матрица 4x2
* np.identity(5) # матрица 5x5
* np.zeros( (2, 3, 4) ) # нулевая матрица размерностью 2x3x4
* np.ones( [4, 3], dtype='int8') # матрица 4x3 из единиц и типом int8
* np.full((3, 2), -1) # матрица 3x2, состоящая из -1

1. С помощью цикла или рекурсии
2. np.random.sample()
3. Операции, которые можно производить над массивами

* len(l) — количество элементов в l;
* min(l) — наименьший элемент;
* max(l) — наибольший элемент;
* sum(l) — сумма чисел списка;
* for i in list() — перебирает элементы слева направо.
* index
* count
* append
* sort
* insert
* remove
* pop

Вывод: в ходе данной лабораторной работы получил практические навыки работы с двумерными массивами.