# **ФГБОУ ВО «Московский политехнический университет»**

## **Лабораторная работа №5**

Двумерные массивы и функции

**Задание 1 Вариант №11**

По дисциплине:

Программирование

Выполнил

студент 1 курса

группы 201-321

Колобов К.В.

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_ Никишина И.Н.

# **МОСКВА 2020**

**Цель работы**

Получить практический навык в написании программ обработки одномерных

массивов: поиск максимумов и минимумов, сортировка.

**Постановка задачи**

Уплотнить заданную матрицу, удаляя из неё заполненные нулями строки и столбцы.

Найти номер первой из строк, содержащих хотя бы один положительный элемент.

**Теоретическая часть**

В Python реализовать массив можно через вложенные списки. В следующем

листинге программы показан прием формирования двумерного списка и способы

инициализации его элементов. В этой программе следует обратить внимание на то, что

элемент списка, ранее инициированный числом, получает значение строкового типа и

преобразование делается по умолчанию.

**Описание программы**

Программа написана на Python 3.8, реализована в среде ОС GNU/Linux Ubuntu 20.04.

**Описание алгоритма**

1. Получаем на вход одно или два натуральных числа – размерность матрицы. Если введено одно число N, то матрица будет квадратной порядка N. Если введено два числа M и N – прямоугольной MxN. Печатаем сообщение об ошибке, если что-то пошло не так.
2. Создаём матрицу, заполняем её случайными целыми числами из диапазона.
3. Печатаем матрицу.
4. Удаляем из матрицы нулевые строки.
5. Печаем матрицу.
6. Удаляем из матрицы нулевые столбцы.
7. Печатаем матрицу.

**Описание входных и выходных данных**

Программа получает на вход одно или два натуральных числа, выводит целые числа и символы.

**Листинг программы**

Листинг приведён в Приложении 1.

Блок-схема приведена в Приложении 2.

**Результат работы программы**

>>> 2

Matrix 2 \* 2

[0, -1]

[0, -1]

There is no positive element

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Matrix 2 \* 2

[0, -1]

[0, -1]

There is no positive element

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Matrix 2 \* 1

[-1]

[-1]

There is no positive element

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Список используемой литературы**

Документация языка Python: <https://docs.python.org/3/>

**Приложение 1**

import random

import sys

from typing import Any, List

RANGE\_START = -1

RANGE\_STOP = 1

class Matrix:

def \_\_init\_\_(self, rows: int, cols: int = None) -> None:

self.\_\_rows, self.\_\_cols = rows, cols or rows

self.\_\_matrix = Matrix.create\_matrix(self.rows, self.cols)

@property

def rows(self) -> int:

return self.\_\_rows

@property

def cols(self) -> int:

return self.\_\_cols

@classmethod

def create\_matrix(cls, rows: int, cols: int) -> Any:

matrix = [

[random.randint(RANGE\_START, RANGE\_STOP) for \_ in range(cols)]

for \_ in range(rows)

]

return matrix

def remove\_zeroes\_rows(self) -> None:

row\_index = 0

for \_ in range(self.rows):

if all(row == 0 for row in self[row\_index]):

del self[row\_index]

row\_index -= 1

row\_index += 1

self.\_\_rows = len(self)

def remove\_zeroes\_cols(self) -> None:

col\_index = 0

for \_ in range(self.cols):

if all(row[col\_index] == 0 for row in self):

for row in range(self.rows):

del self[row][col\_index]

col\_index -= 1

col\_index += 1

try:

self.\_\_cols = len(self[0])

except IndexError:

self.\_\_cols = 0

def num\_of\_positive\_row(self) -> int:

for num, row in enumerate(self, 1):

if any(item > 0 for item in row):

return num

return -1

def \_\_len\_\_(self) -> int:

return len(self.\_\_matrix)

def \_\_getitem\_\_(self, index: int) -> List[int]:

return self.\_\_matrix[index]

def \_\_setitem\_\_(self, index: int, new\_row: List[int]) -> None:

try:

self.\_\_matrix[index] = new\_row

except IndexError:

raise IndexError(f"Wrong row index: {index}")

def \_\_delitem\_\_(self, index: int) -> None:

try:

del self.\_\_matrix[index]

except IndexError:

raise IndexError(f"Wrong row index: {index}")

def \_\_str\_\_(self) -> str:

positive\_row = self.num\_of\_positive\_row()

if positive\_row == -1:

positive\_row = "There is no positive element"

else:

positive\_row = f"Num of the first row " \

f"with a positive item: {positive\_row}"

info = f"Matrix {self.rows} \* {self.cols}"

matrix = '\n'.join(str(row) for row in self)

return f"{info}\n{matrix}\n{positive\_row}"

def main():

size = input("Enter the size of the matrix: ").split()

try:

size = list(map(int, size))

assert 0 < len(size) <= 2

assert all(i > 0 for i in size)

except ValueError:

print("Wrong input", file=sys.stderr)

exit(-1)

except AssertionError:

print("One of two natural values were expected", file=sys.stderr)

exit(-1)

matrix = Matrix(\*size)

print(matrix)

print('\_' \* 20)

matrix.remove\_zeroes\_rows()

print(matrix)

print('\_' \* 20)

matrix.remove\_zeroes\_cols()

print(matrix)

print('\_' \* 20)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Приложение 2**

