ГУАП  
КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | А.В. Аграновский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАБОТА С МАССИВАМИ |
| по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | М023 |  |  |  | Д.А. Трегуб |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

ПОСТАНОВКА ЗАДАЧИ

Задание № 1  
Заданы две матрицы. Найдите произведение этих матриц.

Требования к программе:  
Цель программы: вывод сообщения о результате вычислений или сообщения об ошибке. На вход подаётся список списков. Выходными данными является список списков или сообщения по ошибке.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Классический случай перемножения матриц заключается в трёх вложенных циклах последовательного умножения и сложения для получения результата.

ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ

Таблица 1 - Описание переменных

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип переменной | Назначение |
| first | List | Хранение первого массива |
| second | List | Хранение второго массива |
| result | List | Хранение результирующего массива |
| n | Int | Хранение количества строк первого массива |
| countСolumn | Int | Хранение столбцов первого массива |
| key | Str | Ключ для продолжения работы программы |

На вход подаётся строка два списка со списками – first и second.

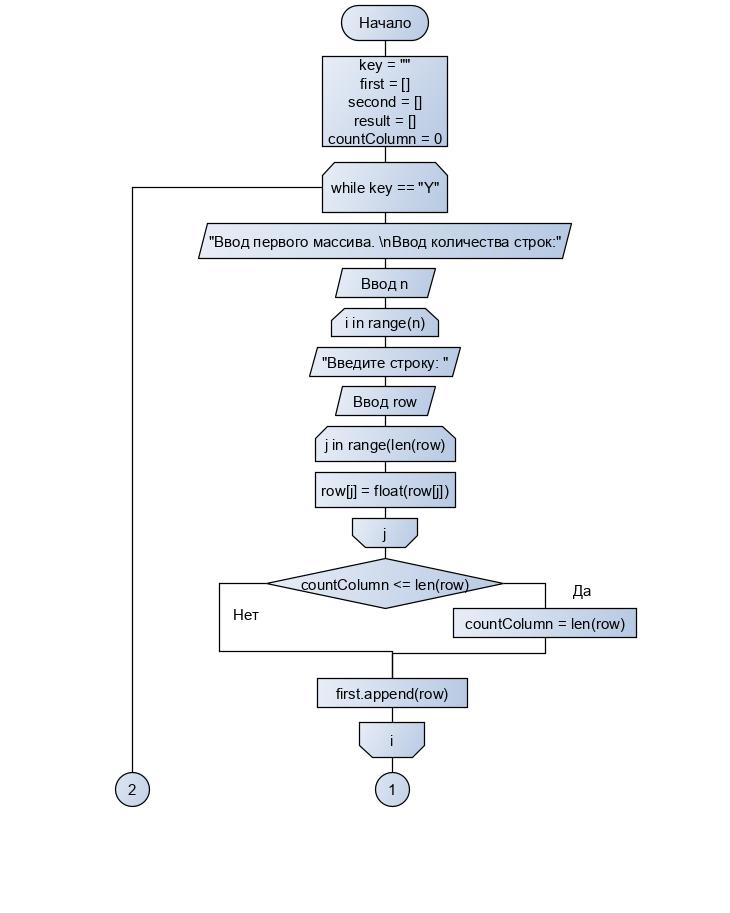
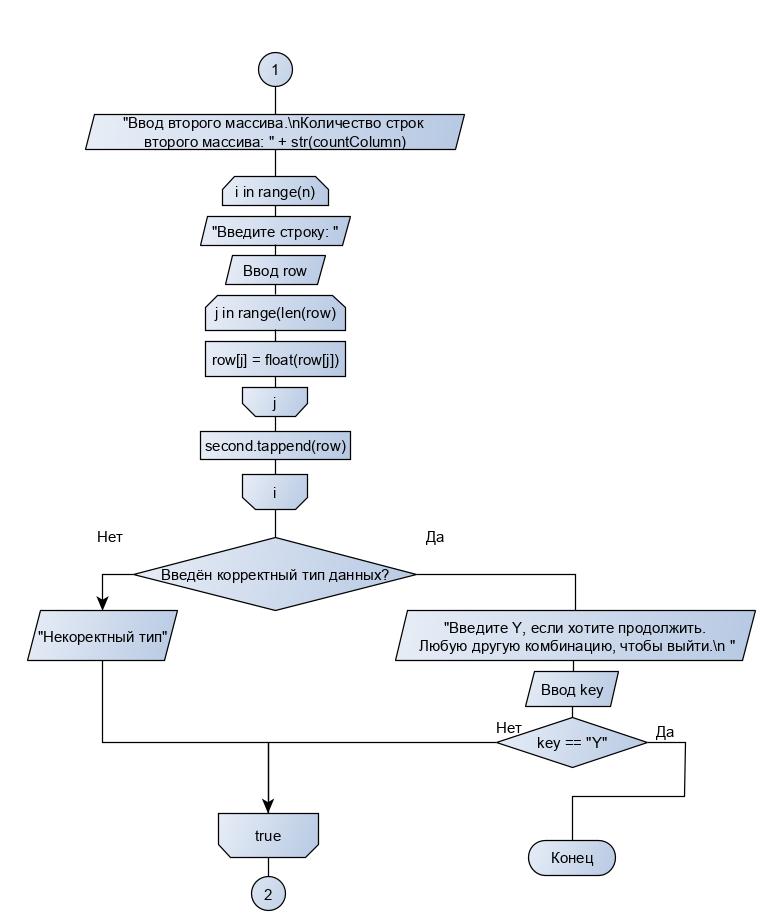
 

Рисунок 1 – функция main

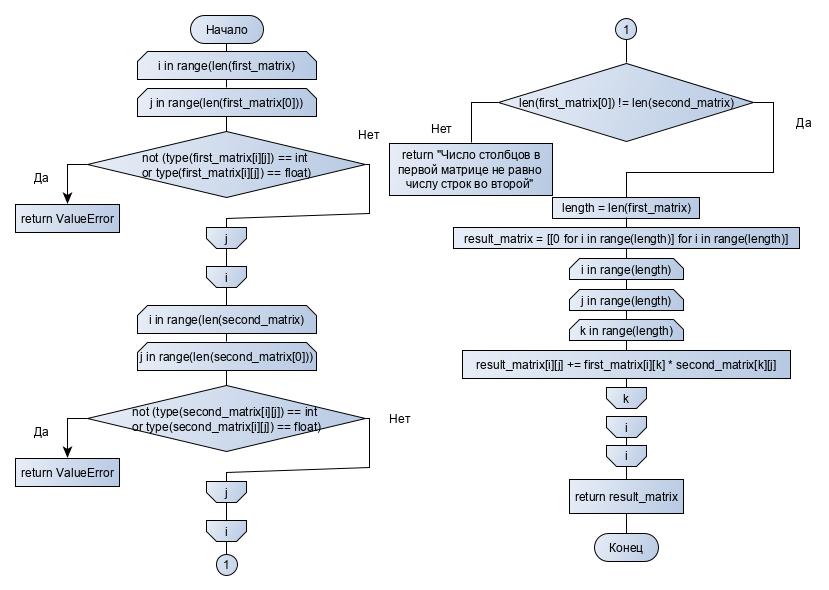


Рисунок 2 - функция LAB6\_VAR\_1

ОПИСАНИЕ ТЕСТОВОГО НАБОРА

Таблица 2 - Тестовый набор

|  |  |  |  |
| --- | --- | --- | --- |
| Описание тестового случая | Входные данные | Ожидаемый результат | Результат теста |
| Классический случай с двумя квадратными матрицами | [[1, 2], [3, 4]], [[5, 6], [7, 8]] | [[19.0, 22.0], [43.0, 50.0]] | OK |
| Передан список с строковыми данными | [["one", "two"], [3, 4]], [[5, 6], [7, 8]] | ValueError | OK |
| Передан список с логическими значениями | [[False, True], [3, 4]], [[5, 6], [7, 8]] | ValueError | OK |
| Преданы список с тремя столбцами в первом и двумя строками и список с двумя столбцами и двумя строками | [[1, 2, 3], [3, 4, 5]], [[5, 6], [7, 8]] | "Число столбцов в первой матрице не равно числу строк во второй" | OK |
| Преданы список с двумя столбцами и строками в первом и двумя строками и список с двумя столбцами и тремя строками | [[5, 6], [7, 8]], [[1, 2, 3], [3, 4, 5]] | [[23.0, 34.0], [31.0, 46.0]] | OK |

ПРИМЕР РАБОТЫ ПРОГРАММЫ

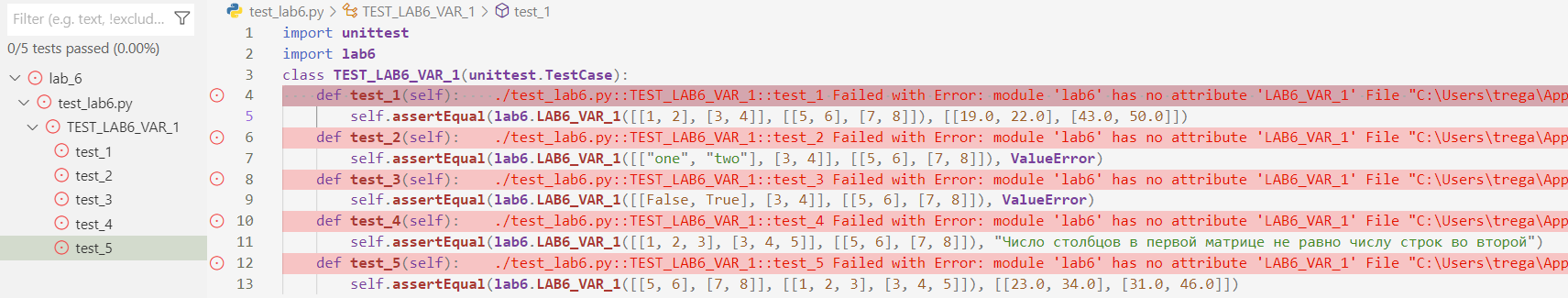


Рисунок 3 - Результат работы тестов до написания программы

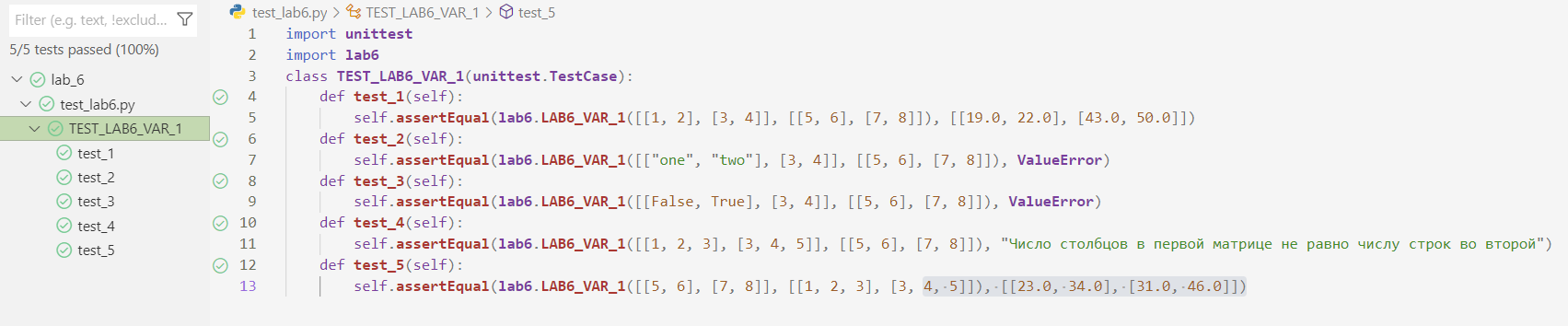


Рисунок 4 - Результат работы тестов после написания программы

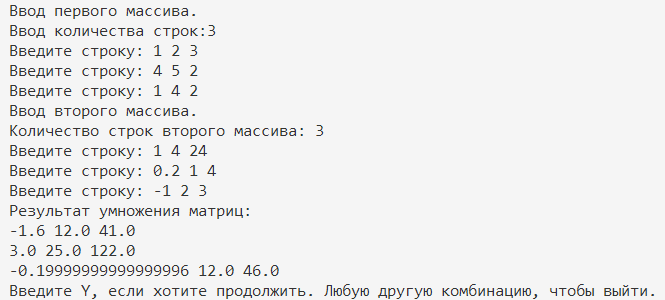


Рисунок 5 - Результат работы программы

ВЫВОД

Были улучшены навыки работы с python, изучена навыки работы с массивами, была выполнена поставленная задача. Составленная программа была успешно протестирована.

ПРИЛОЖЕНИЯ

Приложение 1

Код программы

Код тестов:

import **unittest**

import **lab6**

class **TEST\_LAB6\_VAR\_1**(**unittest**.**TestCase**):

    def **test\_1**(self):

        self.**assertEqual**(**lab6**.**LAB6\_VAR\_1**([[1, 2], [3, 4]], [[5, 6], [7, 8]]), [[19.0, 22.0], [43.0, 50.0]])

    def **test\_2**(self):

        self.**assertEqual**(**lab6**.**LAB6\_VAR\_1**([["one", "two"], [3, 4]], [[5, 6], [7, 8]]), **ValueError**)

    def **test\_3**(self):

        self.**assertEqual**(**lab6**.**LAB6\_VAR\_1**([[False, True], [3, 4]], [[5, 6], [7, 8]]), **ValueError**)

    def **test\_4**(self):

        self.**assertEqual**(**lab6**.**LAB6\_VAR\_1**([[1, 2, 3], [3, 4, 5]], [[5, 6], [7, 8]]), "Число столбцов в первой матрице не равно числу строк во второй")

    def **test\_5**(self):

        self.**assertEqual**(**lab6**.**LAB6\_VAR\_1**([[5, 6], [7, 8]], [[1, 2, 3], [3, 4, 5]]), [[23.0, 34.0], [31.0, 46.0]])

Код функции LAB6\_VAR\_1:

def **LAB6\_VAR\_1**(first\_matrix, second\_matrix):

*#Проверка на соответствие элементов массива разрешённым типам*

    for i in **range**(**len**(first\_matrix)):

        for j in **range**(**len**(first\_matrix[0])):

            if not (**type**(first\_matrix[i][j]) == **int** or **type**(first\_matrix[i][j]) == **float**):

                return **ValueError**

    for i in **range**(**len**(second\_matrix)):

        for j in **range**(**len**(second\_matrix[0])):

            if not (**type**(second\_matrix[i][j]) == **int** or **type**(second\_matrix[i][j]) == **float**):

                return **ValueError**

*#Конец проверки на соответствие типам*

    if **len**(first\_matrix[0]) != **len**(second\_matrix): *#если число столбцов не равно числу строк во второй,*

        return "Число столбцов в первой матрице не равно числу строк во второй" *#Вернуть соответсвующее сообщение*

    length = **len**(first\_matrix) *#запись длины первой матрицы*

    result\_matrix = [[0 for i in **range**(length)] for i in **range**(length)] *#Создание результирующей матрицы, определяемую длиной первой матрицы*

*#Процесс вычисления матрицы*

    for i in **range**(length):

        for j in **range**(length):

            for k in **range**(length):

                result\_matrix[i][j] += first\_matrix[i][k] \* second\_matrix[k][j]

*#Конец вычисления*

    return result\_matrix

Код функции main:

import **lab6**

*#Иницилизация переменных*

key = ""

first = []

second = []

result = []

countСolumn = 0 *#количество столбцов*

while True:

*#Начало блока обработки исключений*

    try:

        n = **int**(**input**("Ввод первого массива. \nВвод количества строк:")) *#Ввод количества строк в первом массиве*

        for i in **range**(n): *#цикл по строкам*

            row = **input**("Введите строку: ").**split**() *#Ввод строки как массива, разделительный символ "пробел"*

            for i in **range**(**len**(row)): *#цикл по элементам в строке*

                row[i] = **float**(row[i]) *#приведение элемента строки к типу float*

            if (countСolumn <= **len**(row)): *#Если количество столбцов предыдущей строки меньше, чем данной,*

                countСolumn = **len**(row) *#приравнять количество столбцов к данному*

            first.**append**(row) *#добавить строку как новый элемент списка*

**print**("Ввод второго массива.\nКоличество строк второго массива: " + **str**(countСolumn))

        for i in **range**(countСolumn): *#цикл по строкам*

            row = **input**("Введите строку: ").**split**() *#Ввод строки как массива, разделительный символ "пробел"*

            for i in **range**(**len**(row)): *#цикл по элементам в строке*

                row[i] = **float**(row[i]) *#приведение элемента строки к типу float*

            second.**append**(row) *#добавить строку как новый элемент списка*

        result = **lab6**.**LAB6\_VAR\_1**(first, second) *#Вызов функции и запись результата её работы*

**print**("Результат умножения матриц:")

        for i in **range**(**len**(result)): *#Цикл по строкам*

            for j in **range**(**len**(result[i])): *#цикл по элементам в строке*

**print**(result[i][j], end=' ') *#вывод элемента*

**print**() *#переход на новую строку*

    except **ValueError**: *#При вызове исключения "ValueError"*

**print**("Некорректный тип")

    key = **str**(**input**("Введите Y, если хотите продолжить. Любую другую комбинацию, чтобы выйти.\n "))*#Ввод комбинации для продолжения или завершения программы*

    if(key == "Y"):

        countСolumn = 0 *#Обнуление счётчика столбцов*

        continue *#Продолжение программы*

    else:

            break *#Завершение программы*

*#Конец бесконечного цикла*