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 № 22

Выполнил:

студент группы ИВТАПбд-21

Старостин К. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков Ильнур Ирекович

Ульяновск, 2023

# Задание по варианту

Необходимо реализовать предложенный вариант задания. Выполнить обработку элементов прямоугольной матрицы A, имеющей N строк и M столбцов. Исходная матрица состоит из нулей и единиц. Добавить к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.

# Требования к программе

1. Ввод элементов списка должен быть доступен путем автоматической генерации. Необходимо использовать библиотеку numpy. Результаты выполнения должны сохраняться в файл (исходные данные и результат обработки).
2. Исходный код должен быть откомментирован.
3. Необходимо реализовать правильную декомпозицию программы на методы.

# Описание реализации

Здесь мы определяем три метода: generate\_matrix, add\_even\_column и save\_matrix. Метод generate\_matrix генерирует случайную матрицу заданного размера с помощью функции np.random.randint. Размер задаётся пользователем перед вызовом этого метода.

Метод add\_even\_column добавляет столбец, используя функцию np.sum для подсчета количества единиц в каждой строке (axis=1 считает среднее значение вдоль столбцов) и сравнивает остаток от деления на два с единицей.

Функция np.column\_stack объединяет исходную матрицу и столбец с четным количеством единиц в каждой строке. Метод save\_matrix сохраняет матрицу в файл, используя функцию np.savetxt. fmp = ‘%d’ – параметр задаёт формат, в котором будут сохраняться значения.

Пользователь вводит высоту и ширину матрицы, значения передаются в метод generate\_matrix, выводится матрица. После выполняется метод add\_even\_column, в которых поступает сгенерированная матрица, матрица выводится.

После этого в файл сохраняется начальная и конечная матрица.

# Описание возникших затруднений

Возникли небольшие трудности с определением количества единиц в каждой строке. О том, как это можно было сделать представления было не много. Поэтому на просторах интернета было найдено решение данного вопроса.

Также возникли проблемы с записью двух матриц в один файл, поэтому я записал их в два разных файла.

# Результат работы программы

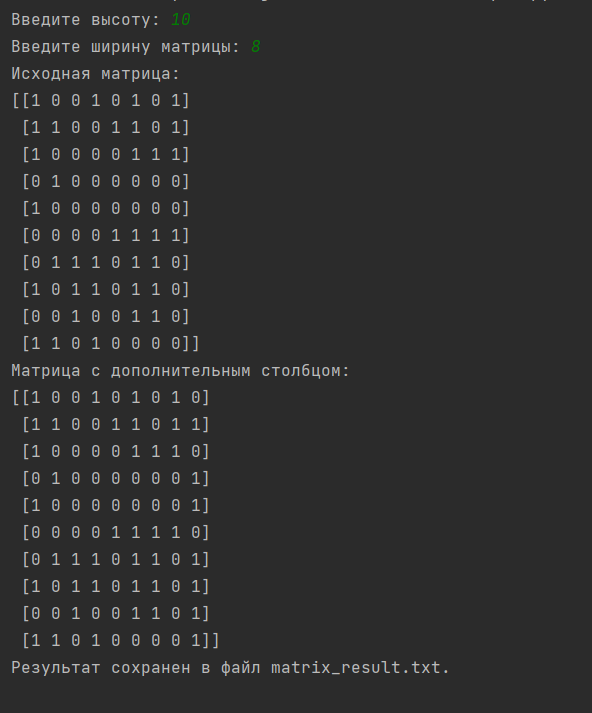


Рисунок 1 Результат работы

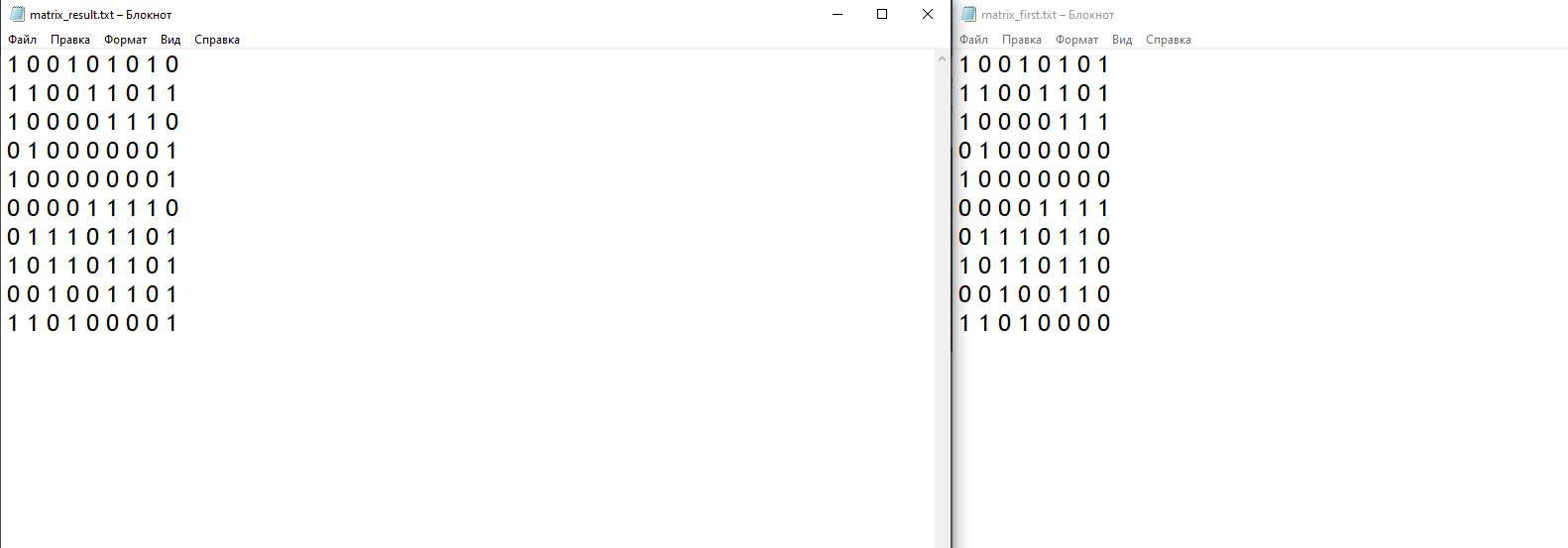


Рисунок 2 Вывод результата и начальной матрицы в файлы

# Код

import numpy as np  
  
  
# Генерирует матрицу случайных нулей и единиц размера N на M  
  
  
def generate\_matrix(n, m):  
 return np.random.randint(2, size=(N, M))  
  
# Добавляет к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.  
  
  
def add\_even\_column(matrix):  
 even\_count = np.sum(matrix, axis=1) % 2 == 1  
 return np.column\_stack((matrix, even\_count))  
  
# Сохраняет матрицу в файл.  
  
  
def save\_matrix(matrix, file\_name):  
 with open(file\_name, "w") as f:  
 np.savetxt(f, matrix, fmt="%d")  
  
# Генерируем матрицу  
  
  
N = int(input("Введите высоту: "))  
M = int(input("Введите ширину матрицы: "))  
matrix = generate\_matrix(N, M)  
print("Исходная матрица:")  
print(matrix)  
  
# Добавляем столбец с четным количеством единиц в каждой строке  
matrix\_with\_even\_column = add\_even\_column(matrix)  
print("Матрица с дополнительным столбцом:")  
print(matrix\_with\_even\_column)  
  
# Сохраняем матрицу в файл  
file\_name\_two = "matrix\_first.txt"  
file\_name = "matrix\_result.txt"  
save\_matrix(matrix, file\_name\_two)  
save\_matrix(matrix\_with\_even\_column, file\_name)  
print(f"Результат сохранен в файл {file\_name}.")