Институт информационных технологий и управления

в технических системах

Кафедра информационных технологий и компьютерных систем

ОТЧЕТ

по лабораторной работе № 9

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ**

**ОПЕРАЦИИ НАД МАТРИЦАМИ»**

по дисциплине «Программирование. Базовые процедуры обработки информации»

Выполнил студент группы ИВТ/б-11д

 Орлов И.В.

Проверил доцент Петров И.И.

Севастополь

2020

**1. ЦЕЛЬ РАБОТЫ**

Целью данной работы является исследование циклических алгоритмов и программ, осуществляющих типичные операции над двумерными массивами (матрицами), получение навыков разработки и использования подпрограмм (методов-процедур и методов-функций).

2. **ПОСТАНОВКА ЗАДАЧИ**

1) Ознакомиться с теоретическими сведениями, приведенными в пункте 3.4 методических указаний: многомерные массивы в Java и операции над матрицами.

2) Разработать программу, реализующую операции над матрицами, по заданию в соответствии с номером варианта. Программа должна удовлетворять нижеперечисленным требованиям.

В основной программе (методе main) нужно инициализировать две квадратные матрицы (число строк равно числу столбцов), содержащих элементы заданного числового типа.

Основная программа (метод main) должна вызывать три вспомогательные подпрограммы (три метода).

Первый **метод-функция** предназначен для вычисления скалярного значения в соответствии с заданием, указанным в столбце 3 таблицы вариантов. Под скалярным значением понимается значение простого типа.

Второй **метод-функция** предназначен для вычисления векторного значения в соответствии с заданием, указанным в столбце 4 таблицы вариантов. Под векторным значением понимается значение типа «одномерный массив».

Третий **метод-функция** должен возвращать матрицу, являющуюся результатом выполнения операции над матрицами, заданной в столбце 5 таблицы вариантов.

Кроме того, необходимо включить в проект **методы-процедуры** для вывода на экран матрицы и одномерного массива.

Основная программа (метод main) должна выполнять следующие действия:

1) инициализировать две квадратные матрицы А и В;

2) вывести:

исходные матрицы А и В;

результат выполнения метода-функции 1 для матрицы А и матрицы В;

результат выполнения метода-функции 2 для матрицы А и матрицы В;

результат выполнения метода-функции 3 для матрицы A и матрицы B.

Вариант задания V вычислен по формуле

 V = (N%14 != 0) ? N%14 : 14; ,

где N – номер студента в списке группы:

1%14=1;

Данные варианта задания приведены в таблице 2.1

Таблица 2.1 – Вариант задания

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер варианта | Базовый тип матриц (тип элементов ) | Функция 1 (метод 1) c параметром A | Функция 2(метод 2) с параметром A | Функция 3(метод 3) с параметрами A, B, m  |
| 1 | byte | Cумма элементов, расположенных под главной диагональю | Вектор, содержащий минимальные значения каждой строки матрицы A | Матрица:m(A+B) |

**3. АНАЛИЗ ЗАДАЧИ**

**3.1. Входные данные, выходные данные, функции**

**и структура программы**

Согласно заданию программа должна инициализировать две квадратные матрицы А и В с элементами типа byte и применить к каждой из них три разработанных метода-функции. Результаты обработки должны выводиться в окно терминала. Вызывающая программа будет иметь линейную структуру. Все подробности обработки массивов будут скрыты в соответствующих методах. Алгоритм вычислительного процесса:

1. инициализировать матрицу А;
2. инициализировать матрицу B;
3. вывести матрицу A (вызов процедуры вывода матрицы);
4. вывести матрицу В (вызов процедуры вывода матрицы);
5. вычислить и вывести сумму элементов, расположенных под главной диагональю для матрицы А (вывод значения, возвращаемого соответствующей функцией);
6. сформировать и вывести вектор, содержащий минимальные значения каждой строки матрицы А (вызов процедуры вывода вектора с передачей в качестве аргумента значения, которое вернет соответствующая функция);
7. сформировать и вывести матрицу, являющуюся результатом умножения суммы матриц А и В на константу m (вызов процедуры вывода матрицы с передачей в качестве аргумента значения, которое вернет соответствующая функция);
8. вычислить и вывести сумму элементов, расположенных под главной диагональю для матрицы В (вывод значения, возвращаемого соответствующей функцией);
9. сформировать и вывести вектор, содержащий минимальные значения каждой строки матрицы В (вызов процедуры вывода вектора с передачей в качестве аргумента значения, которое вернет соответствующая функция);
10. сформировать и вывести матрицу, являющуюся результатом умножения суммы матриц В и А на константу m (вызов процедуры вывода матрицы с передачей в качестве аргумента значения, которое вернет соответствующая функция).

***Вставьте названия методов, которые будут использоваться в вашей программе***

Структура программы изображена на рисунке 3.1.



 Рисунок 3.1 – Структура программы

**3.2. Входные данные, выходные данные, выполняемые действия**

**для вызываемых методов**

Первый метод – method1 – предназначен для вычисления значения суммы элементов матрицы matr, расположенных под главной диагональю. По заданию он должен быть реализован как функция. Формальный параметр метода: matr – локальная ссылочная переменная (ссылка на двумерный массив (матрицу) элементов типа byte). При вызове метода переменной matr будет присвоено значение соответствующего фактического параметра, т.е. адрес матрицы, созданной в вызывающем методе (main). Таким образом, передача параметра методу осуществляется по ссылке. Представляется целесообразным, чтобы метод возвращал значение типа int (для одиночного целочисленного значения целесообразно применять универсальный тип int).

Второй метод – method2 – предназначен для создания одномерного массива (вектора), содержащего минимальные значения каждой строки матрицы matr. По заданию метод должен быть реализован как функция. Формальный параметр метода: matr – локальная ссылочная переменная (ссылка на двумерный массив (матрицу) элементов типа byte). При вызове метода переменной matr будет присвоено значение соответствующего фактического параметра, т.е. адрес матрицы, созданной в вызывающем методе (main). Таким образом, передача параметра методу осуществляется по ссылке. Тип возвращаемого значения – ссылка на одномерный массив элементов типа byte, созданный в методе. Схема алгоритма метода2 изображена на рисунке 3.2.

***Схема алгоритма должна быть изображена аккуратно и достаточно крупно***

Рисунок 3.2 – Схема алгоритма метода 2

Третий метод – method3 – предназначен для создания матрицы, являющейся результатом умножения суммы двух матриц matr1 и matr2 на целочисленное значение m. По заданию метод должен быть реализован как функция. Метод имеет три формальных параметра. Первый и второй формальные параметры метода: matr1 и matr2 – локальные ссылочные переменные (ссылки на матрицы элементов типа byte). Передача первых двух параметров осуществляется по ссылке. Третий формальный параметр – локальная переменная m типа int. При вызове метода переменной m будет присвоено значение соответствующего фактического параметра, т.е. переменной или int-литерала, указанного в вызове метода. Таким образом, второй параметр передается по значению. Тип возвращаемого значения –ссылка на матрицу элементов типа int. Тип int выбран, исходя из того, что элемент результирующей матрицы, являющийся суммой соответствующих элементов исходных матриц, умноженной на константу, может не поместиться в разрядную сетку типа byte. Чтобы выводить и исходные матрицы и результирующую матрицу нужно в программе иметь два перегруженных метода вывода матриц: первый – с параметром типа byte[][], второй – с параметром типа int[][].

Схема алгоритма метода 3 изображена на рисунке 3.3.

***Схема алгоритма должна быть изображена аккуратно и достаточно крупно***

Рисунок 3.3 – Схема алгоритма метода method3

Четвертый метод – putVect – предназначен для вывода в окно терминала элементов одномерного массива (вектора) в одну строку. Метод является процедурой. Формальный параметр метода: vector – локальная ссылочная переменная (ссылка на одномерный массив элементов типа byte). Передача параметра осуществляется по ссылке. Тип возвращаемого значения – void (метод не возвращает значения).

Пятый метод – putMatr – предназначен для вывода матрицы с элементами типа byte (byte[][]) в окно терминала. Метод является процедурой. Формальный параметр метода: matr – локальная ссылочная переменная (ссылка на матрицу элементов типа byte). Передача параметра осуществляется по ссылке. Тип возвращаемого значения – void (метод не возвращает значения).

Шестой метод – putMatr (перегружен для типа int[][]) – предназначен для вывода матрицы с элементами типа int в окно терминала. Метод является процедурой. Формальный параметр метода: matr – локальная ссылочная переменная (ссылка на матрицу элементов типа int). Передача параметра осуществляется по ссылке. Тип возвращаемого значения – void (метод не возвращает значения).

**4. ТЕСТОВЫЕ ПРИМЕРЫ И РЕЗУЛЬТАТЫ ИХ ОБРАБОТКИ ВРУЧНУЮ**

*В данном пункте нужно привести пример двух матриц одинаковой размерности и ожидаемые результаты его обработки. Если есть частные случаи, в которых вид исходных матриц существенно влияет на результат обработки (например, требуется сложить или перемножить квадратные матрицы разной размерности), нужно придумать тестовые примеры для таких частных случаев и привести ожидаемые результаты обработки для каждого тестового примера.*

**6.ТЕКСТ ПРОГРАММЫ**

Согласно пункту 2.8.1. методических указаний к лабораторным работам, текст программы должен быть отформатирован следующим образом:

1. шрифт: arial, 12 пт, полужирный;
2. абзац:
* выравнивание по левому краю;
* уровень – основной текст;
* отступ слева – 0;
* отступ справа – 0;
* интервал перед – 0;
* интервал после – 0;
* первая строка – нет (отступа или выступа),
* междустрочный интервал – множитель, значение – 1,2;
1. комментарии (обязательно должны присутствовать в тексте программы) выделить курсивом и синим цветом.

**7. СВЕДЕНИЯ ОБ ОТЛАДКЕ ПРОГРАММЫ И ПРОВЕРКЕ**

**ЕЕ РАБОТОСПОСОБНОСТИ**

*В данном пункте сначала следует описать ошибки, которые возникли при отладке программы и способ их устранения.*

Результаты выполнения программы представлены на рисунке 7.3.

***Скриншот окна терминала (не мельчить!)***

 Рисунок 7.3 – Результаты выполнения программы

На тестовых примерах программа выдала ожидаемый результат, что позволяет сделать вывод о ее работоспособности.

**ВЫВОД**

В ходе выполнения лабораторной работы были изучены принципы структурирования данных в виде многомерных (двумерных) массивов в Java, принципы обработки двумерных массивов с помощью использования вложенных операторов цикла, принципы структурирования программ, правила передачи параметров в метод и возвращения методом значения, а также разработана программа, вызывающая различные методы обработки матриц.

***Уважаемые студенты! Обращаю ваше внимание на то, что шаблоны отчетов даются вам для того, чтобы вы научились писать технические отчеты и правильно составлять документацию к программам. Поэтому настоятельно прошу сначала внимательно прочитать текст шаблона, при необходимости творчески переработать его в соответствии с данными своего варианта и только затем вставлять результаты работы своей программы.***