**Пояснительная записка**

Роженко Варвара бпи205

Практическое задание 1, вариант 187 (условие задач 5, обработка данных в контейнере 14)

**Описание полученного задания**

**Задание:** Разработка контейнера, содержащего квадратные матрицы с действительными числами.  
**Виды матриц:** 1) обычный двумерный массив, 2) диагональная матрица, 3) нижняя треугольная матрица.   
**Диагональная матрица** реализована на основе одномерного массива, содержащего элементы, стоящие на диагонали.  
**Нижняя треугольная матрица** реализована на основе одномерного массива с формулой пересчета.   
**Общая для всех альтернатив переменная** – размерность.  
**Общая функция для всех альтернатив функция** – вычисление среднего арифметического.   
**Функция обработки данных в контейнере** – упорядочивание по убыванию, используя сортировку Шелла.

**Формат ввода команды:** случайная генерация матриц (*./abc -n number inputData outputData*, где inputData (файл со входными данными) outputData (файл для записи результата) файлы формата txt) или считывание из файла *(./abc -f inputData outputData*, где inputData (файл со входными данными) outputData (файл для записи результата) файлы формата txt).

**Формат описания матриц:** (вид матрицы) (размерность матрицы) (элементы матрицы в строку через пробел)

Виды матрицы соответствуют номерам: 1 – обычный двумерный массив,  
2 – диагональная матрица, 3 – нижняя треугольная матрица.

Пример описания матрицы: 1 2 1 1 2 2 – двумерный массив размерности 2:

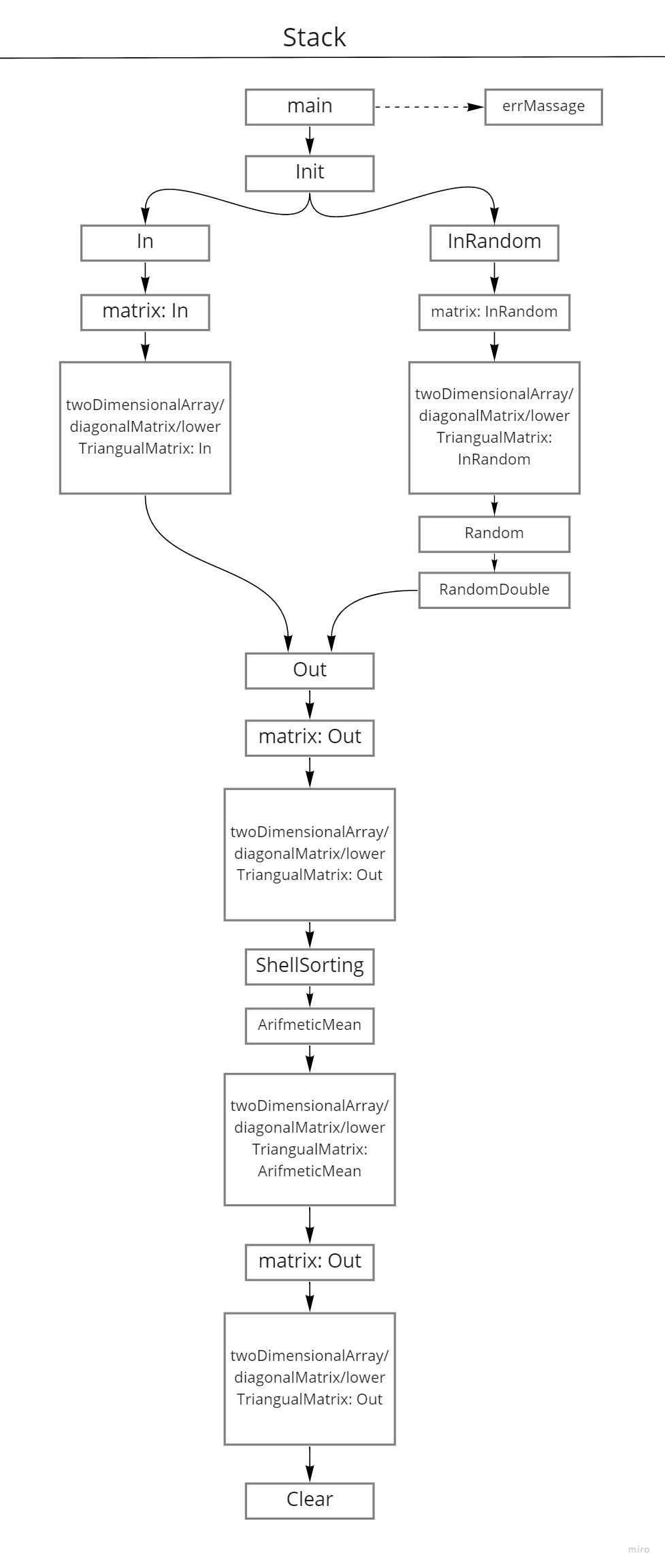
**Структурная схема ВС с размещенной на ней разработанной программой**

Таблица типов:

|  |  |
| --- | --- |
| int | 4 |
| double | 8 |
| enum | 4 |
| struct twoDimensionalArray  dimension: int tdaElements: double\*\* | 12  4[0]  8[4] |
| struct diagonalMatrix  dimension: int tdaElements: double\* | 12  4[0]  8[4] |
| struct lowerTriangularMatrix  dimension: int tdaElements: double\* | 12  4[0]  8[4] |
| struct matrix  k: enum  union:  tda twoDimensionalArray  dm diagonalMatrix  ltm lowerTriangularMatrix | 16  4[0]  12[4]  12[4]  12[4] |
| struct container  max\_len: enum  len: int  \*cont[max\_len]: matrix | 40008  4[0]  4[4]  40\*10000[8]=40000 |

Память программы:

|  |  |
| --- | --- |
| int main(int argc, char\* argv[])  argc: int  argv: char\*\*  c: container  fileInput: FILE\*  size: int  fileOutput: FILE\* | 40040  4[0]  8[4]  40008[12]  8[40020]  4[40028]  8[40032] |
| void Init(container &c)  c: container | 40008  40008[0] |
| matrix\* In(FILE \*file)  file: FILE\*  k: int  m: matrix\* | 20  8[0]  4[8]  8[12] |
| matrix\* InRandom()  k: int  m: matrix\* | 12  4[0]  8[4] |



**Характеристики программы**

Число интерфейсных модулей: 6

Число модулей реализации: 5+main.cpp

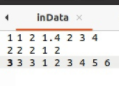
Общее число строк кода: 558

**Время выполнения программы для различных тестовых наборов данных**

1. **­­­­­**время: **Изображение выглядит как текст

   Автоматически созданное описание**

входные данные:



выходные данные:

Изображение выглядит как текст

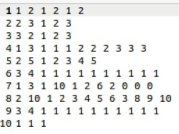
Автоматически созданное описание

1. время:

Изображение выглядит как текст

Автоматически созданное описание

входные данные:

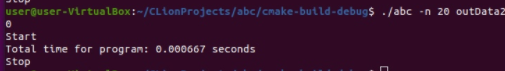


выходные данные:

Изображение выглядит как текст, газета

Автоматически созданное описание

1. время для 20 элементов:



выходные данные:

Изображение выглядит как текст

Автоматически созданное описание

1. время для 8000 элементов:

Изображение выглядит как текст

Автоматически созданное описание

1. время для 10000 элементов:

Изображение выглядит как текст

Автоматически созданное описание

1. переполнение:

