

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«Национальный исследовательский

**Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

**Кафедра математического обеспечения и суперкомпьютерных
технологий**

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

**«Разработка структуры хранения верхнетреугольной
матрицы»**

Выполнил: студент группы
381706-2

Крюков Дмитрий Алексеевич

_____ Подпись

Научный руководитель:

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

_____ Подпись

Нижний Новгород

2018

Содержание

Содержание.....	2
1. Введение.....	3
2. Постановка задачи.....	4
3. Руководство пользователя.....	5
4. Руководство программиста.....	6
4.1 Описание структуры программы.....	6
4.2 Описание структур данных.....	6
4.3 Описание алгоритмов.....	7
6. Заключение.....	8
7. Литература.....	9

1. Введение

Матрица — математический объект, записываемый в виде прямоугольной таблицы элементов кольца \ или поля (например, целых, действительных или комплексных чисел), которая представляет собой совокупность строк и столбцов, на пересечении которых находятся её элементы. Количество строк и столбцов задает размер матрицы.

Верхняя треугольная матрица (или верхнетреугольная матрица) — квадратная матрица, у которой все элементы ниже главной диагонали равны нулю

Для таких матриц эффективнее хранить в памяти нужно только возможно ненулевые элементы, заведомые нули ниже главной диагонали хранить незачем.

2. Постановка задачи

Цель данной лабораторной работы — разработать на языке программирования C++ статическую библиотеку, реализующую структуру данных для хранения верхнетреугольных матриц и структуру данных для хранения вектора.

Для вектора определены операции:

- сравнение
- присваивание
- умножить на скаляр
- векторные операции
- сложение векторов
- вычитание векторов
- скалярное произведение

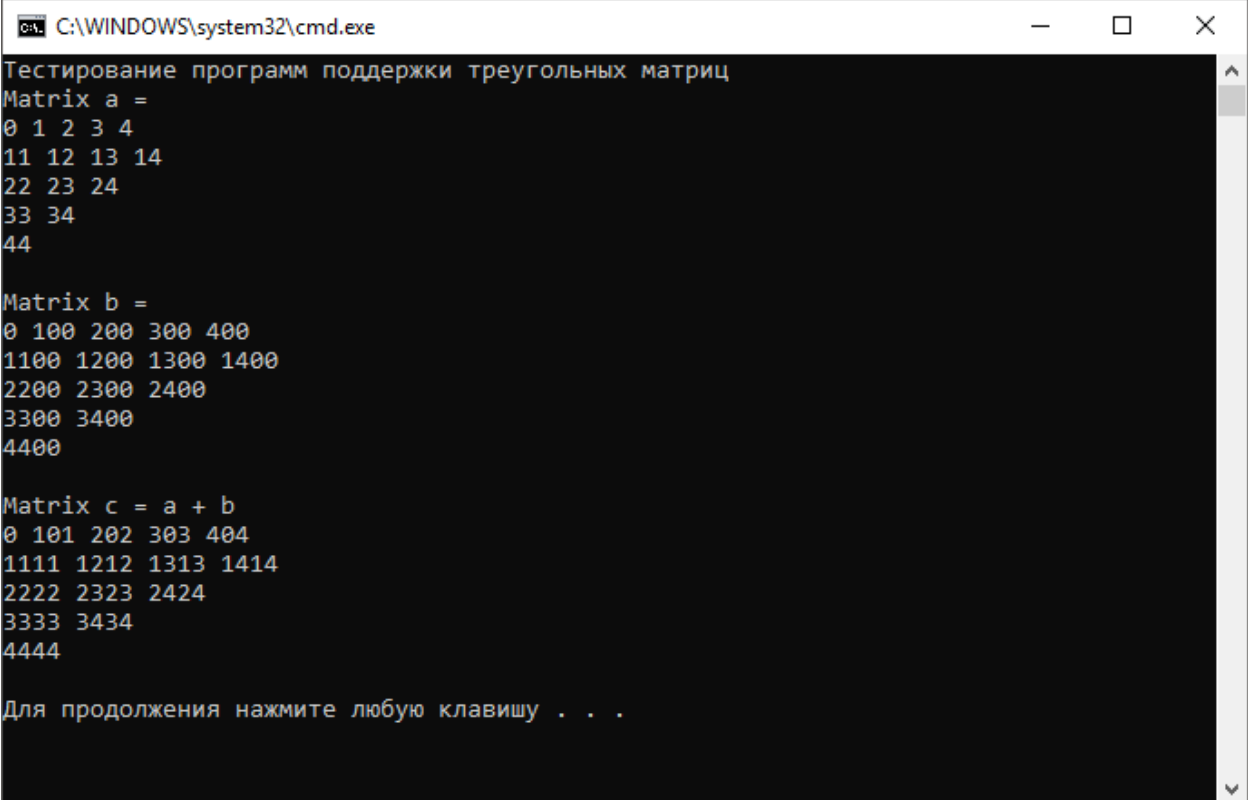
Для матрицы определены операции:

- сложение
- вычитание
- умножение
- деление

3. Руководство пользователя

Данная программа предназначена для тестирования реализации класса матрица.

На экран выводятся 2 матрицы, затем производится сложение этих матриц и на экран выводится результат.



```
C:\WINDOWS\system32\cmd.exe
Тестирование программ поддержки треугольных матриц
Matrix a =
0 1 2 3 4
11 12 13 14
22 23 24
33 34
44

Matrix b =
0 100 200 300 400
1100 1200 1300 1400
2200 2300 2400
3300 3400
4400

Matrix c = a + b
0 101 202 303 404
1111 1212 1313 1414
2222 2323 2424
3333 3434
4444

Для продолжения нажмите любую клавишу . . .
```

4. Руководство программиста

4.1 Описание структуры программы

1. Модуль `vectorlib` (`vectorlib.h`) – реализация класса вектор
2. Модуль `matrixlib` (`matrixlib.h`) – реализация класса вектор
3. Модуль `vector(main.cpp)` – реализация программы для тестирования программ поддержки треугольных матриц
4. Модуль `vectortest(vectortest.cpp)` - тестирование класса вектор при помощи Google C++ Testing Framework.
5. Модуль `matrixtest(matrixtest.cpp)` - тестирование класса матрица при помощи Google C++ Testing Framework.

4.2 Описание структур данных

Структура `vector`:

Поля:

`pVector` - память для вектора

`size` - размер вектора

`startIndex` - индекс первого элемента вектора

Методы:

`IsEmpty()` - контроль пустоты

`GetCount()` - число элементов в стеке

`IsFull()` - контроль переполнения

`Put(const int Val)` - добавить значение

`Get()` - извлечь значение

Структура matrix:

Поля унаследованы из класса вектор

Методы:

`operator==(const TMatrix &mt)` - сравнение

`det()` - определитель

`invert()` - обратная матрица

`operator= (const TMatrix &mt)` - присваивание

`operator+ (const TMatrix &mt)` - сложение

`operator- (const TMatrix &mt)` - вычитание

`operator* (const TMatrix &mt)` - умножение

`operator/ (TMatrix &mt)` - деление

4.3 Описание алгоритмов

Нахождение обратной матрицы:

Создать матрицу res размера size

Для $i = 0$ до size

`res[i][i] = 1/pVector[i][i];`

Для $i = 0$ до size

 Для $j = i + 1$ до size

`sum = 0;`

 Для $k = i$ до j

`sum += res[i][k] * pVector[k][j]`

`res[i][j] = -sum / pVector[j][j]`

6. Заключение

В ходе работы реализован класс матрицы, в нем реализованы функции сложение вычитание умножение деление

Реализован класс вектор, в нем реализованы функции сравнение, присваивания, умножить на скаляр, сложение векторов, вычитание векторов, скалярное произведение

7. Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Обращение верхней треугольной матрицы - Режим доступа: <https://lektsii.org/10-95480.html>