МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

Высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**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

Банденков Даниил Викторович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2019.

# Содержание

[Содержание 2](#_Toc534333789)

[1 Введение 3](#_Toc534333790)

[2 Постановка задачи 4](#_Toc534333791)

[3 Руководство пользователя 5](#_Toc534333792)

[4 Руководство программиста 6](#_Toc534333793)

[4.1 Описание структуры программы 6](#_Toc534333794)

[4.2 Описание структур данных 6](#_Toc534333795)

[4.3 Описание алгоритмов 7](#_Toc534333796)

[5 Заключение 8](#_Toc534333797)

[6 Литература 9](#_Toc534333798)

# Введение

Матрица — это двумерный массив, каждый элемент которого имеет два индекса: номер строки и номер столбца, поэтому для работы с элементами матрицы необходимо использовать два цикла. Если значениями параметра первого цикла будут номера строк матрицы, то значениями параметрами второго — столбцы (или наоборот). Обработка матрицы заключается в том, что вначале поочередно рассматриваются элементы первой строки (столбца), затем второй и т.д. до последней.

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.

# Постановка задачи

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

Для матрицы определены операции:

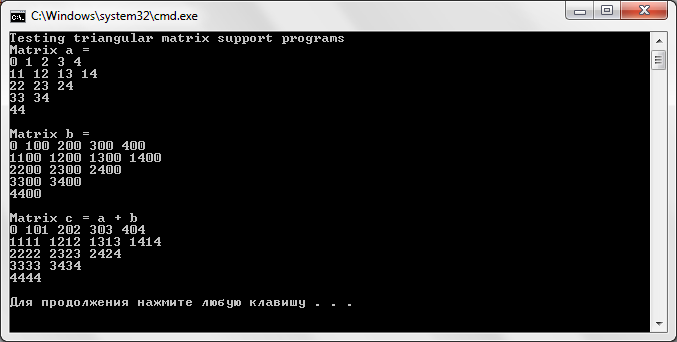
1. Сложение
2. Вычитание
3. Умножение
4. Деление

Для вектора определены операции:

1. Сравнение
2. Присваивание
3. Умножение на скаляр
4. Векторные операции
5. Сложение векторов
6. Вычитание векторов
7. Скалярное произведение

# Руководство пользователя

Данная программа тестирует реализацию класса матриц. На экран выводится 2 матрицы, затем производится сложение этих матриц и на экран выводится результат (см Рис 1).

Рис 1.Пример использования программы.

# Руководство программиста

## Описание структуры программы

*В решении содержатся следующие модули:*

1. vectorlib (vectorlib.h)– модуль реализующий класс вектор.
2. matrixlib (matrixlib.h)– модуль реализующий класс матриц.
3. vector(main.cpp)- модуль тестирования поддержки программы треугольных матриц.
4. vectortest(vectortest.cpp)- модуль тестирования класса вектор при помощи Google C++ Testing Framework.
5. matrixtest(matrixtest.cpp)- модуль тестирования класса матрица при помощи Google C++ Testing Framework.

## Описание структур данных

**Структура vector:**

*Поля:*

pVector - память для вектора.

size - размер вектора.

startIndex — индекс первого элемента вектора.

*Методы:*

int IsEmpty() - контроль пустоты.

int GetCount() - число элементов в стеке.

int IsFull() - контроль переполнения.

void Put(const int Val) - добавить значение.

ValType Get() - извлечь значение.

**Структура matrix:**

*Поля унаследованы из класса вектор*

*Методы:*

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

invert() - обратная матрица.

operator= (const TMatrix &mt) - присваивание.

operator+ (const TMatrix &mt) - сложение.

operator- (const TMatrix &mt) - вычитание.

operator\* (const TMatrix &mt) - умножение.

operator/ (const TMatrix &mt) - деление.

## Описание алгоритмов

*Нахождение обратной матрицы (псевдокод)*

Создать матрицу obrat размера size

Для i = 0 до size шаг 1

obrat[i][i] = 1 /pVector[i][i]

Для i = 0 до size шаг 1

Для j = i + 1 до size шаг 1

sum = 0

Для k = i до j шаг 1

sum += obrat[i][k] \* pVector[k][j]

obrat[i][j] = -sum /pVector[j][j]

Вернуть obrat;

# Заключение

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

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

# Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. <http://kvodo.ru/urok-9-matritsyi-obshhie-svedeniya.html>

(Дата обращения 02.01. 2019)

1. https://ru.wikipedia.org/wiki/Matrix (Дата обращения 02.01. 2019)