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**

**«Структура хранения матриц специального вида»**

**Выполнил:** студент группы 381706-1

Остапович Денис Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc532232359)

[2. Постановка задачи 4](#_Toc532232360)

[3. Руководство пользователя 5](#_Toc532232361)

[4. Руководство программиста 7](#_Toc532232362)

[4.1 Описание структуры программы 7](#_Toc532232363)

[4.2 Описание структур данных 7](#_Toc532232364)

[4.3 Описание алгоритмов 9](#_Toc532232365)

[5. Заключение 12](#_Toc532232366)

[6. Литература 13](#_Toc532232367)

# 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. Хотя исторически рассматривались, например, треугольные матрицы, в настоящее время говорят исключительно о матрицах прямоугольной формы, так как они являются наиболее удобными и общими.

Существуют треугольные матрицы, которые в свою очередь делятся на верхнюю треугольную матрицу и нижнюю треугольную матрицу. Верхней треугольной матрицей называется квадратная матрица, у которой все элементы ниже главной диагонали равны нулю. А нижней треугольной матрицей называется матрица, у которой все элементы выше главной диагонали равны нулю.

В программировании матрицей называется двумерный массив. Для представления частных случаев квадратных матриц: верхние треугольные, нижние треугольные и т.д. Удобно использовать специальную структуру хранения и обработки. Например, для хранения верхних треугольных матриц необязательно хранить нули. Достаточно хранить элементы выше главной диагонали и саму главную диагональ.

**Цель лабораторной работы** – разработка структуры хранения верхнетреугольных матриц.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Описание вспомогательного класса TVector.
2. Описание класса матриц TMatrix
3. Продемонстрировать работу классов TMatrix и TVector.
4. Создать набор автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

После запуска программы создаются три матрицы одинаковой размерности. Далее первые две из них автоматически заполняются числами. После чего выводятся на экран. Далее третьей матрице поочерёдно присваивается результат допустимых матричных операций с первыми двумя матрицами и выводится на экран. После этого создаётся матрица размерности 3 и пользователю предлагается заполнить её самостоятельно. При успешном завершении операции ввода, получившаяся матрица выводится на экран. Программа завершается.

# Руководство программиста

## 4.1 Описание структуры программы

Программа состоит из следующих модулей:

* **Vector** – содержит в себе файл **main.cpp** с реализацией примера использование класса **TVector**.
* **VectorLib –** содержит в себе файл **мector.h**, в котором описан класс **TVector**.
* **VectorTest** – содержит в себе файл **VectorTest.cpp**, в котором находится набор тестов, для проверки работоспособности класса **TVector**.
* **Matrix** – содержит в себе файл **main.cpp** с реализацией примера использование класса **TMatrix**.
* **MatrixLib –** содержит в себе файл **matrix.h**, в котором описан класс **TMatrix**.
* **MatrixTest** – содержит в себе файл **MatrixTest.cpp**, в котором находится набор тестов, для проверки работоспособности класса **TMatrix**.

## Описание структур данных

**Класс TVector.**

Класс **TVector** является шаблонным и содержит два поля со спецификатором **protected**:

* **T\* vector** – указатель на область памяти для хранения вектора.
* **int size** – размер вектора.

Далее идут методы класса со спецификатором доступа **public**:

* **TVector(int \_size=5)** – конструктор инициализатор.
* **TVector(TVector<T> &A)** – конструктор копирования.
* **~TVector()** – деструктор.
* **int GetSize()** – возвращает размер вектора.
* **T& operator[] (int index)** – оператор доступа по индексу.
* **bool operator==(const TVector<T> &A)** – оператор сравнения.
* **TVector<T>& operator=(const TVector<T> &A)** – оператор присваивания.
* **TVector<T> operator+(const T &num)** – оператор прибавления к вектору числа.
* **TVector<T> operator-(const T & num)** – оператор вычитания из вектора числа.
* **TVector<T> operator\*(const T & num)** – оператор умножения вектора на число.
* **TVector<T> operator+(const TVector<T> &A)** – оператор сложения двух векторов.
* **TVector<T> operator-(const TVector<T> &A)** – оператор разности двух векторов.
* **T operator\*(const TVector<T> &A)** – оператор скалярного умножения двух векторов.
* **friend std::istream& operator >> (std::istream &istr, TVector<T1> &vector)** – дружественная функция ввода вектора в поток.
* **friend std::ostream& operator << (std::ostream &ostr, TVector<T1> &vector)** – дружественная функция вывода вектора на экран.

**Класс TMatrix.**

Класс **TMatrix** унаследован от класса **TVector** и не имеет собственных полей со спецификатором доступа **protected**.

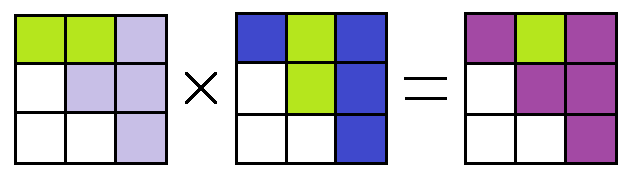
Методы со спецификатором доступа **public**:

* **TMatrix(int \_size = 5)** – конструктор инициализатор.
* **TMatrix(TMatrix<T> &A)** – конструктор копирования.
* **TMatrix(TVector<TVector<T> > &A)** – конструктор преобразования типа.
* **bool operator==(const TMatrix<T> &A)** – оператор сравнения.
* **TMatrix<T>& operator=(const TMatrix<T> &A)** – оператор присваивания.
* **TMatrix<T> operator+(const TMatrix<T> &A)** – оператор сложения матриц.
* **TMatrix<T> operator-(const TMatrix<T> &A)** – оператор разности матриц.
* **TMatrix<T> operator\*(const TMatrix<T> &A)** – оператор умножения матриц.
* **TMatrix<T> operator/(TMatrix<T> &A)** – оператор деления матриц.
* **friend std::istream& operator >> (std::istream &istr, TMatrix<T1> &matrix)** – дружественная функция ввода матрицы в поток.
* **friend std::ostream& operator << (std::ostream &ostr, TMatrix<T1> &matrix)** – дружественная функция вывода матрицы на экран.

## Описание алгоритмов

**Умножение матриц.**

Элементами матрицы C=A\*B являются равен сумме произведений элементов *i*-ой строки матрицы А на соответствующие элементы *j*-го столбца матрицы B.



**Деление матриц.**

Результатом деления матрицы A на матрицу B является матрица где – матрица, обратная к матрице B*.* Обратная матрица может быть найдена следующим образом:

* Записываем матрицу
* С помощью элементарных преобразований строк нужно получить единичную матрицу справа .

Получим матрицу - обратную матрицу к матрице B (.

# Заключение

В данной лабораторной работе мною была реализована библиотека для хранения и работы с верхними треугольными матрицами:

* Реализован и протестирован класс **TVector**.
* Реализован и протестирован класс **TMatrix**.

Таким образом, была реализована структура данных для хранения верхних треугольных матриц в памяти, как массива векторов.

# Литература

* Книги

1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
2. Bjarne Stroustrup The C++ programming language

* Ссылки в Internet

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [<http://www.itmm.unn.ru/files/2018/10/Primer-1.2.-Struktury-hraneniya-matrits-spetsialnogo-vida.pdf>], 2015.
2. Википедия [https://ru.wikipedia.org/wiki/Матрица\_(математика)