Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Информационных технологий, математики и механики

Отчёт по лабораторной работе

**Структура хранения верхнетреугольных матриц**

Выполнил:

студент ИТММ гр. 381808-2

Некравцев А. А.

Проверил:

Нижний Новгород

2019 г.

**Содержание**

Введение ….............................................................................................................3

Постановка задачи ….............................................................................................4

Руководство пользователя …................................................................................5

Руководство программиста …..............................................................................6

Описание структуры программы ….........................................................6

Описание структур данных …..................................................................7

Описание алгоритмов …...........................................................................8

Заключение …......................................................................................................10

Приложение ….....................................................................................................11

**Введение**

Матричные обозначения широко распространены в современной математике и её приложениях. Матрица – полезный аппарат для исследования многих задач теоретической и прикладной математики. Так, одной из важнейших является задача нахождения решения систем линейных алгебраических уравнений.

Помимо матриц общего вида, для которых наиболее естественной и наиболее часто используемой представляется программная реализация в виде двумерного массива, в математических приложениях выделяются различные матрицы специальных видов (треугольные, диагональные, …). Для таких матриц предпочтительно создание собственных способов хранения и обработки, учитывающих специфику их структуры, и потому более эффективных. Изучению некоторых из них посвящена данная работа.

**Постановка задачи**

В рамках лабораторной работы ставится задача создания программных средств, поддерживающих эффективное хранение матриц специального вида (верхнетреугольных) и выполнение основных операций над ними:

* сложение/вычитание;
* копирование;
* сравнение.

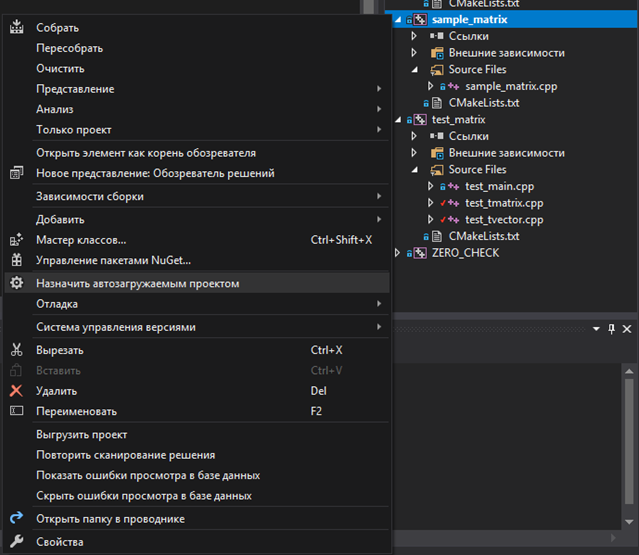
В процессе выполнения лабораторной работы требуется использовать фрэймворк для написания автоматических тестов Google Test.

1. Выполнение работы предполагает решение следующих задач:
2. Реализация методов шаблонного класса TVector согласно заданному интерфейсу.
3. Реализация методов шаблонного класса TMatrix согласно заданному интерфейсу.
4. Обеспечение работоспособности тестов и примера использования.
5. Реализация заготовок тестов, покрывающих все методы классов TVector и TMatrix.
6. Модификация примера использования в тестовое приложение, позволяющее задавать матрицы и осуществлять основные операции над ними.

**Руководство пользователя**

Пользователю предоставляется возможность воспользоваться двумя раннее написанными классам TVector и TMatrix, оформленные в файле utmatrix.h.

Необходимо лишь запустить проект sample\_matrix (ctrl+F5), назначив его автозагружаемым проектом.



**Руководство программиста**

**Описание структуры программы**

* gtest — библиотека Google Test.
* include — директория для размещения заголовочных файлов.
* samples — директория для размещения тестового приложения.
* src — директория для размещения исходных кодов (cpp-файлы).
* test — директория с модульными тестами и основным приложением, инициализирующим запуск тестов.

В решении содержатся следующие модули:

* Модуль utmatirx, содержащий реализацию классов Вектор и Матрица (файл ./include/utmatrix.h). Поскольку оба класса шаблонные, реализацию методов необходимо выполнять непосредственно в заголовочном файле. При этом интерфейсы классов должны оставаться неизменными.
* Тесты для классов Вектор и Матрица (файлы ./test/test\_tvector.cpp, ./test/test\_tmatrix.cpp).
* Пример использования класса Матрица (файл ./samples/sample\_matrix.cpp).

**Описание структур данных**

В данной лабораторной работе, мы храним матрицу в виде набора векторов разной длины.

Очевидно сходство в задании структуры данных Вектор, как набора элементов, связанных отношением следования, и структуры данных Матрица, как набора элементоввекторов, связанных отношением следования. Этот факт позволяет единообразно организовать алгоритмы обработки векторов и матриц, а, следовательно, использовать при разработке требуемых классов механизм наследования.

То есть для реализации алгоритмов нами использовано 2 класса:

* Класс «Вектор» (TVector).
* Класс «Матрицы» (TMatrix), который будет использовать класс векторов.

**Описание алгоритма**

**Класс TVector**

TVector<ValType>::TVector(**int** s, **int** si) // конструктор создает вектор с размером s и с индексом si

TVector<ValType>::TVector(**const** TVector<ValType> &v) // конструктор копирования

TVector<ValType>::~TVector() // освобождает выделенную память под вектор

ValType& TVector<ValType>::**operator**[](**int** pos) // дает доступ к элементу вектора

**bool** TVector<ValType>::**operator**==(**const** TVector& v) **const** //сравнение двух

**bool** TVector<ValType>::**operator**!=(**const** TVector &v) **const** // векторов

TVector<ValType>& TVector<ValType>::**operator**=(**const** TVector &v) // присваивание

TVector<ValType> TVector<ValType>::**operator**+(**const** ValType &val) // прибавить скаляр (также перегружен и с -)

TVector<ValType> TVector<ValType>::**operator**\*(**const** ValType &val) // скалярное произведение векторов

TVector<ValType> TVector<ValType>::**operator**+(**const** TVector<ValType> &v) // сложение векторов (также перегружен и с -)

**Класс TMatrix**

Основан на методах TVector.

|  |
| --- |
|  |

**Заключение**

Целью работы была реализация структуры данных для хранения матриц специального вида (верхнетреугольных матриц).

В результате была написана структура данных для хранения матриц специального вида (верхнетреугольных матриц), в которой уменьшены затраты памяти для хранения векторов, путем создания векторов разной длины. Были переопределены арифметические операции для работы с матрицами, реализованными в данной структуре.

Программные средства созданы при помощи среды программирования С++ и содержат:

* класс Вектор (на шаблонах);
* класс Матрица (на шаблонах);

В программе также реализована система обработки ошибок.

**Приложение**

Фрагменты исходного кода представлены в личном репозитории на Github:

[https://github.com/Alexey42/mp2-lab2-matrix/tree/develop](https://github.com/Alexey42/mp2-lab2-matrix/tree/ed)