Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчёт по лабораторной работе

**Структуры хранения матриц специального вида**

Выполнил:

студент ф-та ПРИН гр. 381908-4

Трофимов В.А.

Проверил:

ассистент каф. МОСТ, ИИТММ

Лебедев И.Г

Нижний Новгород

2020 г.

Содержание

[Введение 3](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962758)

[Постановка задачи 4](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962759)

[Руководство программиста 6](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962761)

[Описание структур данных 6](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962762)

[Описание алгоритмов 6](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962763)

[Описание структуры программы 6](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962764)

[Заключение 7](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962765)

[Литература 8](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962766)

[Приложения 9](file:///C:\Users\Dragonsnom\Desktop\github\1-laba\mp2-lab1-set\report.doc#_Toc270962767)

# Введение

Понятие Матрица в европейской науке было введено в работах У. Гамильтона3 и А. Кэли4 в середине XIX века.

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.

Следствием разнообразия областей применения матричного аппарата в современной науке является наличие в любом из больших математических программных комплексов (Mathcad, Mathematica, Derive, Mapple) подсистем, выполняющих операции над матрицами, а также существование специальных программных библиотек (ScalaPack, PlaPack), рассчитанных на обработку огромных (десятки и сотни тысяч строк) матриц, в том числе с использованием распределенных (параллельных) вычислений.

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;

• умножение;

• копирование;

• сравнение.

Программные средства должны содержать:

• класс Вектор (на шаблонах);

• класс Матрица (на шаблонах);

• тестовое приложение, позволяющее задавать матрицы и осуществлять основные операции над ними.

Сделаем следующие основные допущения:

• Условимся рассматривать в дальнейшем верхнетреугольные квадратные матрицы, состоящие из элементов произвольного типа.

• Будем считать размер матрицы конечным числом, не превышающим 231

# Руководство программиста

## **Описание структуры программы**

Реализует операции:

bool operato r== (const TMatrix& mt) const; //сравнение Матриц

TMatrix operator = (const TMatrix& mt); //присваивание Матриц

TMatrix operator+ (const TMatrix& mt); //сложение Матриц

TMatrix operator- (const TMatrix& mt); //вычитание Матриц

TMatrix operator\* (const TMatrix& mt); //умножение Матриц

Для структуры данных Вектор были реализованы следующие операции:

• вычисление длины;

• сравнение;

• прибавление/вычитание скаляра;

• умножение на скаляр;

• сложение/вычитание векторов;

• скалярное произведение векторов;

• создание копии.

Для структуры данных Матрица были реализованы следующие операции:

• сравнение:

• сложение/вычитание матриц:

• умножение матриц.

С учетом структуры данных целесообразной представляется следующая модульная структура программы:

• Vector.h, Vector.cpp – модуль, реализующий структуру данных Вектор;

• Matrix.h, Matrix.cpp – модуль, реализующий структуру данных Матрица;

•test\_matrix.cpp – модуль программы тестирования матриц.

• test\_vector.cpp – модуль программы тестирования векторов.

## **Описание алгоритмов**

Создание вектора:

* Инициализируем размер вектора;
* Заполняем вектор;

Создание матрицы:

* Инициализируем строки матрицы
* Заполняем строки элементами от 0 до кол-во векторов, которые понадобятся для создание матрицы;

# Эксперименты

В данной лабораторной работе были трудности с онлайн компилятором “Travis”, его компилятор очень чувствителен к написанию файлов. Практически все операции были выполнены с помощью методов if и for

# Заключение

В ходе выполнения данной лабораторной работы были реализованы операции над матрицами и их создание, был изучен новый вид проверки работоспособность программ “Travis”. А также полная самостоятельность исполнения поставленных задач.

# Литература

1. Столлингс, В. Структурная организация и архитектура компьютерных систем, 5-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 896 с.: ил. — Парал. тит. англ.
2. Johnson M. Superscalar Microprocessor Design. — Englewood Cliff, New Jersey: Prentice Hall, 1991.
3. Касперски К. Техника оптимизации программ. Эффективное использование памяти. — СПб.: БХВ-Петербург, 2003. — 464 с.: ил.
4. Stone H. High performance Computer Architecture. — Reading, MA: Addison-Wesley, 1993.
5. Tullsen D.M., Eggers S.J. Effective Cache Prefetching on a Bus-Based Multiprocessor. — ACM Transactions on Computer Systems, pp. 57-88, Feb 1995.
6. Chandra D., Guo F., Kim S., Solihin Y. Predicting inter-thread cache contention on a chip multi-processor architecture. — Proceedings of the 11th International Symposium on High Performance Computer Architecture (HPCA), pp. 340–351, Feb 2005.
7. Press W., Teukolsky S., Vetterling W., Flannery B. Numerical Recipes in C. The Art of Scientific Computing. Second Edition. — Cambridge University Press, 1992.
8. Камаев А.М., Сиднев А.А., Сысоев А.В. Об одном подходе к анализу эффективности приложений // Труды 50-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук»: Часть I. Радиотехника и кибернетика. - М.: МФТИ, 2007.
9. Debugging and performance monitoring. Intel® 64 and IA-32 Architectures Software Developer’s Manual. Volume 3B: System Programming Guide, Part 2. May 2007. — [http://www.intel.com/products/processor/manuals/]
10. Юнаковский А.Д. Начала вычислительных методов для физиков. – Н. Новгород: ИПФ РАН, 2007.

# Приложения

