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**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Матрица»**

**Выполнил:**студент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2018

Оглавление

[1.Введение 3](#_Toc536816936)

[2.Постановка задачи 4](#_Toc536816937)

[3.Руководство пользователя 5](#_Toc536816938)

[4.Руководство программиста 6](#_Toc536816939)

[4.1.Описание структуры программы 6](#_Toc536816940)

[4.2.Описание структур данных 6](#_Toc536816941)

[4.3.Описание алгоритмов 7](#_Toc536816942)

[5.Эксперименты 8](#_Toc536816943)

[6.Заключение 9](#_Toc536816944)

[7.Литература 10](#_Toc536816945)

1.Введение

Матрицы широко применяются в математике с древних времен. Они позволяют легче представлять информацию и соответственно решать практические задачи. В данной работе представлены верхнетреугольные матрицы. Главное их преимущество: экономное использование памяти. Впервые матрицы упоминались ещё в древнем Китае. Основным применением матриц было решение линейных уравнений. После развития теории определителей в конце 17-го века Габриэль Крамер начал разрабатывать свою теорию в 18-м столетии и опубликовал «правило Крамера» в 1751 году. Примерно в этом же промежутке времени появился «метод Гаусса». Теория матриц начала своё существование в середине XIX века в работах Уильяма Гамильтона и Артура Кэли. Фундаментальные результаты в теории матриц принадлежат Вейерштрассу, Жордану, Фробениусу. Термин «матрица» ввел Джеймс Сильвестр в 1850 г.

2.Постановка задачи

Необходимо написать реализацию вектора и треугольной матрицы со всеми сопутствующими конструкторами, методами и перегрузками. Произвести замеры времени. Матрица представляет собой вектор от вектора. В данной лабораторной работе представлены такие операции для матрицы как, умножение, деление, сложение, взятие элемента по индексу и так далее.

3.Руководство пользователя

Программа демонстрирует перегрузку ввода и вывода для векторов, матриц. Также показывается умножение и деление матриц.

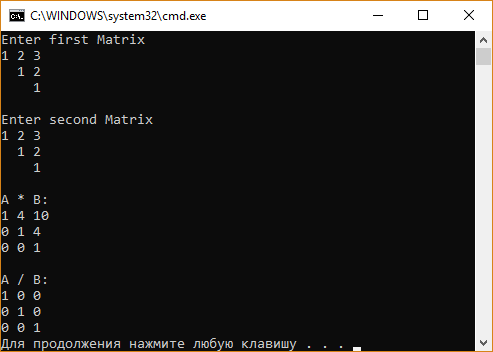


рис. 1

4.Руководство программиста

4.1.Описание структуры программы

Программа состоит из таких модулей как:

* Matrix.h - матрица
* TVector.h - вектор

4.2.Описание структур данных

int l - длина вектора

T\* p – указатель на начало вектора

* TVector() – конструктор по умолчанию
* TVector(int n, int \_pos) – конструктор с параметрами
* TVector(const TVector<T>& A) – конструктор копирования
* ~TVector() - деструктор
* int GetSize() const – метод возвращает размер вектора
* bool operator != (const TVector<T>& A) – перегрузка != возвращает 1 если векторы не равны
* bool operator == (const TVector<T>& A) - перегрузка == возвращает 1 если векторы равны
* TVector<T>& operator = (const TVector<T> &A) - перегрузка =
* T& operator [](int i) const – возвращает ссылку на определенный элемент из вектора с учетом позиции
* T& GetValue(int i) const – возвращает ссылку на определенный элемент из вектора
* TVector<T> operator + (const TVector<T> &A) – перегрузка + складывает 2 вектора
* TVector<T> operator - (const TVector<T>& A) – перегрузка -вычитает один вектор из другого
* TVector<T> operator \* (const TVector<T> &A) – перегрузка \* умножает один вектор на другой
* TVector<T> operator + (const T& k) – перегрузка + прибавляет к каждому элементу из вектора значение k
* TVector<T> operator - (const T& k) - перегрузка - вычитает из каждого элемента вектора значение k
* TVector<T> operator \* (const T& k) - перегрузка \* умножает каждый элемент из вектора на значение k
* TMatrix(int n) – конструктор с параметром
* TMatrix(const TMatrix<T>& A) – конструктор копирования
* TMatrix(const TVector<TVector<T> >& A) – конструктор преобразования из двойного вектора в матрицу
* bool operator==(const TMatrix<T>& A) – перегрузка == возвращает 1 если матрицы равны
* bool operator!=(const TMatrix<T>& A) - перегрузка != возвращает 1 если матрицы отличаются
* TMatrix& operator=(const TMatrix<T>& A) – перегрузка =
* TMatrix operator+(const TMatrix<T>& A) – перегрузка + складывает 2 матрицы
* TMatrix operator-(const TMatrix<T>& A) – перегрузка – вычитает одну матрицу из одной
* TMatrix operator\*(const TMatrix<T>& A) – перегрузка \* умножает одну матрицу на другую
* TMatrix operator/(const TMatrix<T>& B) – перегрузка / умножает матрицу на обратную от другой матрицы

4.3.Описание алгоритмов

Описание алгоритма умножения для матриц

Проверка на одинаковую длину главного вектора

Создание временной матрицы temp

От i = 0 до длинны главного вектора матрицы

От j = i до длинны главного вектора матрицы

От k = i до j + 1

temp[i][j] += эл. из осн. матрицы с индексом [i][k] \* эл. из втор. [k][j]

Вернуть temp

5.Эксперименты

Для эксперимента был взята логическая операция \* O(n^3) из TMatrix.



рис. 2 график

Характеристики компьютера: Amd fx-8350, 8gb оперативной памяти

6.Заключение

Благодаря этой работе у меня появился опыт взаимодействия с матрицами и векторами. Матрицы значительно уменьшают затраты на хранение информации и помогают решать многие практические задания. Также я увидел множество возможностей для применения матриц, укрепил свои знания по математике, а также узнал множество алгоритмов.

7.Литература

1. Википедия https://ru.wikipedia.org/wiki/Массив\_(программирование) (Дата обращения 16.12.2018)
2. Википедия https://ru.wikipedia.org/wiki/Матрица\_(математика) (Дата обращения 16.12.2018)