Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского»

Факультет информационных технологий математики и механики

Отчёт по лабораторной работе

Вектора и матрицы

Выполнил:

студент ф-та ИТММ гр. 38-03

Ульянов Д.М.

Проверил:

Заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений

Лебедев И.Г.

Нижний Новгород

2022 г.

**Содержание**

[Введение 3](#_gjdgxs)

[Постановка задачи 4](#_30j0zll)

[Руководство пользователя 5](#_1fob9te)

[Руководство программиста 6](#_3znysh7)

[Описание структур данных 6](#_3znysh7)

[Описание алгоритмов 6](#_2et92p0)

[Описание структуры программы 6](#_17dp8vu)

[Заключение 7](#_3dy6vkm)

[Литература 8](#_3rdcrjn)

[Приложения 9](#_26in1rg)

[Приложение 1 9](#_lnxbz9)

[Приложение 2 9](#_2s8eyo1)

**Введение**

**C++** (читается *си-плюс-плюс*) — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространенные контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C — наибольшее внимание уделено поддержке объектно-ориентированного и обобщенного программирования.

**C++** широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java и C#.

Синтаксис C++ унаследован от языка [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Изначально одним из принципов разработки было сохранение совместимости с C. Тем не менее C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

1. **Постановка задачи**

Написать классы для работы с векторами и матрицами, использовать шаблоны.

Вектора в математическом понимании: имеется набор значений из N мерного пространства, размерность задается как параметр.

Матрица должна быть наследником вектора.

Классы вектора и матрицы должны быть вынесены в статическую библиотеку.

Продемонстрировать их работу на примере (написать в main пример).

Должны быть:

конструкторы (по умолчанию, инициализатор, копирования), деструктор, доступ к защищенным полям;

перегруженные операции: +,-,\*,/,=,==, [] потоковый ввод и вывод;

перегруженные операции +,-,\*,/ должны быть реализованы для векторов (вектор +-\*/ вектор), матриц (матрица +-\* матрица), матрично-векторные (матрица \* вектор и наоборот);

Дополнительное задание:

1 - написать метод, ищущий количество вхождений указанного значения;

2 - написать метод, ищущий все вхождения заданного значения;

3 - написать метод деления матриц.

4 - в классе вектор должна быть возможность отсортировать его тремя способами (пузырек, вставка, быстрая сортировки).

**2. Руководство пользователя**

Для созданий вектора есть три конструктора: по умолчанию, инициализатор, копирования.



Доступ к полю len - количество элементов в векторе;.



Можно складывать, вычитать, умножать, делить, присваивать, сравнивать вектора и обращаться отдельно к элементам вектора.











Также можно умножить вектор на матрицу (вектор векторов).







Векторы можно выводить и задавать с клавиатуры. Если вы хотите ввести вектор с клавиатуры то сначала надо указать его длину.





Есть дополнительные методы для работы с векторами: изменение размера вектора, подсчет количества элементов определенного значения в векторе, нахождение всех индексов элементов, имеющих конкретное значение, 3 вида сортировки (пузырьком, вставками, быстрая).

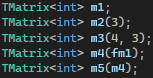








Для созданий матрицы есть пять конструкторов: по умолчанию, инициализатор (создает квадратную / прямоугольную матрицу), копирования (копирует матрицу или вектор векторов).



Доступ к полю width - количество векторов в матрице.



Можно складывать, вычитать, умножать, присваивать (матрицы к матрицам или вектор векторов к матрицам), сравнивать матрицы и обращаться отдельно к векторам матрицы.















Также можно умножить матрицу на вектор.



Матрицы можно выводить и задавать с клавиатуры. Если вы хотите ввести матрицу с клавиатуры то сначала надо указать ее ширину, а затем длину.





Есть дополнительные методы для работы с матрицами: изменение размера матрицы, подсчет количества элементов определенного значения в матрице, нахождение всех индексов элементов, имеющих конкретное значение.

**3. Руководство программиста**

***3.1 Описание структуры программ***

Заголовочный файл Vector.h состоит из шаблонного класса TVector и реализации методов класса.

Заголовочный файл Matrix.h является наследником класса TVector содержит методы для работы с матрицами.

main - программа запрашивает выбор типа чисел: целые или дробные. При выборе соответствующего типа демонстрируются примеры методов.

***3.2 Описание структур данных***

Класса TVector имеет 2 поля len и data, где находятся длинна массива и массив шаблонных элементов соответственно.

Класса TMatrix имеет 3 поля width, len, data. data содержит вектор векторов, width количество векторов в основном векторе, len длина внутреннего вектора.

***3.3 Описание алгоритмов***

ВЕКТОР

Конструктор по умолчанию зануляет 2 поля вектора. Конструктор инициализатор принимает длину вектора, затем создает неинициализированный массив полученной длинны. Конструктор копирования копирует поля принимаемого вектора. Метод GetLen возвращает значение поля len. При выполнения операторов +, -, \*, / у 2 векторов сравнивается длина, если их длины равны то результатом будет вектор той же длины, полученный применением соответствующих операций поэлементно. Оператор \* дополнительно перегружен для умножения вектора на матрицу (вектор векторов). Если длина вектора равна ширине матрицы, то результатом является вектор длины матрицы, полученный как поэлементная сумма вектора на столбцы матрицы. Оператор присваивания вектора - копирует поля вектора. Сравнение векторов происходит поэлементно. Оператор [] принимает целочисленный индекс, если он не выходит за границы массива, то возвращает элемент находящийся на указанной позиции. Метод ReSize изменяет размер массива на принимаемый целочисленный параметр. Метод CountElem принимает параметр который является элементом подсчета в векторе и возвращает количество вхождений этого элемента в векторе. Метод MasIndex принимает 2 параметра: элемент подсчета и целочисленный параметр которому будет присвоено количество вхождений этого элемента. Данный метод возвращает массив индексов на местах которых находится элемент подсчета. Оставшиеся 3 метода сортируют массив в векторе (SortPuzirec, SortVstsvka, SortBistraya). Для быстрой сортировки нужно задать 2 параметра 0 и len - 1.

МАТРИЦА

Конструктор по умолчанию зануляет 3 поля матрицы. Конструктор инициализатор создает квадратную матрицу (массив массивов), если принимаемый целочисленный параметр единственный или прямоугольную матрицу, если принимаемых параметров 2, тогда прямоугольная матрица будет длины первого параметра и ширины второго параметра. Конструктор копирования копирует поля полученной матрицы или вектора векторов. Метод GetWidth возвращает значение поля width. При выполнения операторов +, -, \*, / у 2 матриц сравниваются длина и ширина, если их длина и ширина равны то результатом будет матрица той же длины и ширины, полученная применением соответствующих операций поэлементно. Оператор \* дополнительно перегружен для умножения матрицы на вектор. Если длина матрицы равна длине вектора, то результатом является вектор длина которого равна ширине матрицы, полученный как поэлементная сумма строк матрицы на вектор. Оператор присваивания матрицы - копирует поля матрицы или вектора векторов. Сравнение матриц происходит поэлементно. Оператор [] принимает целочисленный индекс, если он не выходит за границы, то возвращает вектор находящийся на указанной позиции. Метод ReSize изменяет размер матрицы на 2 принимаемых целочисленных параметра (длина и ширина). Метод CountElem принимает параметр который является элементом подсчета в матрице и возвращает количество вхождений этого элемента в матрице. Метод MasIndex принимает 2 параметра: элемент подсчета и целочисленный параметр которому будет присвоено количество вхождений этого элемента. Данный метод возвращает двумерный массив индексов на местах которых находится элемент подсчета.

**Эксперименты**

Для получения времени работы программы я вводил разное количество чисел и заносил результаты в таблицу.

| Время в секундах | При 10000 элементах в векторе, или матрица 100 на 100 элементов | При 1000000 элементах в векторе, или матрица 1000 на 1000 элементов | При 100000000 элементах в векторе, или матрица 10000 на 10000 элементов |
| --- | --- | --- | --- |
| вектор + вектор | 0.001 | 0.011 | 0.991 |
| матрица + матрица | 0.001 | 0.02 | 2.682 |
| матрица \* матрицу | 0.01 | 16.437 | больше 27000 |

**4.Заключение**

Я сделал статическую библиотеку для работы с векторами и матрицами. Реализовал все необходимые методы и убедился, что время выполнения операции + для 2 векторов O(n), у операции + для 2 матриц имеет сложность O(), а \* для 2 матриц O() (если матрицы квадратичные).

**5.Литература**

1. <https://ru.wikipedia.org/wiki/C%2B%2B>.
2. <https://github.com/LebedevIlyaG/3821>

**Приложения**

***Приложение 1***

***Приложение 2***