МИНИСТЕРСТВО НАУКИ И ВЫСШЕГООБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**ЛАБОРАТОРНАЯ РАБОТА**

на тему:

**«Векторы и матрицы»**

**Выполнил(а):** студент(ка) группы 3822Б1ФИ2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Холин К.И

Подпись

**Проверил:** к.т.н, доцент каф. ВВиСП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Кустикова В.Д./

Подпись

Нижний Новгород  
2023

**Содержание**

[Введение 4](#_Toc149507567)

[1 Постановка задачи 5](#_Toc149507568)

[2 Руководство пользователя 6](#_Toc149507569)

[2.1 Приложение для демонстрации работы 6](#_Toc149507570)

[2.2 Приложение для демонстрации работы множеств 7](#_Toc149507571)

[2.3 «Решето Эратосфена» 10](#_Toc149507572)

[3 Руководство программиста 11](#_Toc149507573)

[3.1 Описание алгоритмов 11](#_Toc149507574)

[3.1.1 Битовые поля 11](#_Toc149507575)

[3.1.2 Множества 11](#_Toc149507576)

[3.1.3 «Решето Эратосфена» 11](#_Toc149507577)

[3.2 Описание программной реализации 12](#_Toc149507578)

[3.2.1 Описание класса TBitField 12](#_Toc149507579)

[3.2.2 Описание класса TSet 12](#_Toc149507580)

[Заключение 13](#_Toc149507581)

[Литература 14](#_Toc149507582)

[Приложения 15](#_Toc149507583)

[Приложение А. Реализация класса TBitField 15](#_Toc149507584)

[Приложение Б. Реализация класса TSet 15](#_Toc149507585)

# Введение

Ранее мы уже сталкивались с шаблонным классом TVector, который хранил в себе компоненты вектора разного фундаментального типа данных и переменный размер. В данной лабораторной работе мы описываем реализацию класса шаблонного класса TVector с целью реализации класса матриц TMatrix,наследуемый от класса TVector,принимающий в качества параметра шаблона самого себя. Таким образом, матрица – это класс вектора векторов как интерфейс для работы с матрицами как общего,так и специального вида. В лабораторной работе рассматриваются матрица специального вида – верхне-треугольные.

# Постановка задачи

Цель – реализовать шаблонные классы: TVector и TMatrix

Задачи:

1. Класс для работы с матрицами должен поддерживать эффективное хранение данных.
2. Написать следующие операции для работы с векторами: сложение,вычитание,умножение, копирование векторов,сравнение(==,!=),ввод/вывод.
3. Добавить вспомогательные операции: [], получение стартового индекса и размера вектора, сложение,вычитание,умножение с элементом.
4. Написать следующие операции для работы с матрицами: сложение,вычитание,умножение матриц,копирование матриц, сравнение(==,!=),ввод/вывод.

# Руководство пользователя

## Приложение для демонстрации работы векторов

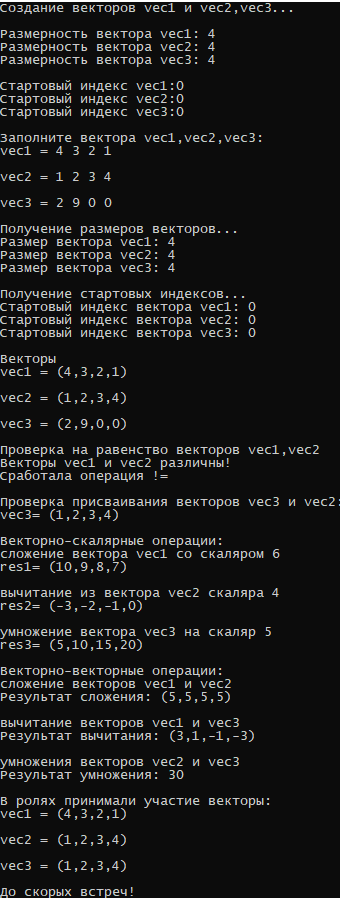
1. Запустите приложение с названием sample\_TBitField.exe.В результате появится окно, показанное на рисунке (рис.1) 
2. Основное окно программы
3. На первом шаге создаются 3 вектора(рис.2)



Рис.2 Создание векторов

1. На следующем шаге заполняются векторы vec1,vec2,vec3 и выводятся их данные на экран(рис.3)

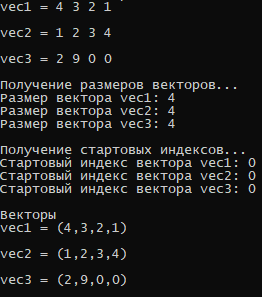


Рис.3 Заполнение векторов и вывод данных

1. Далее идёт сравнение векторов vec1 и vec2(рис.4)



Рис.4 Проверка на равенство векторов

1. На 5 шаге присваются значения вектора vec2 вектору vec3(рис.5)



Рис.5 Проверка присваивания векторов

1. На данном этапе выполняются векторно-скалярные операции с векторами(рис.6)

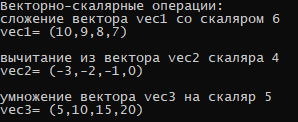


Рис.6 Векторно-скалярные операции с векторами

1. После проводятся векторно-векторные операции(рис.7)

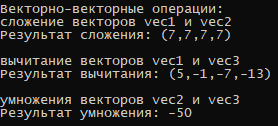


Рис.7 Векторно-векторные операции с векторами

1. На завершающем этапе выводятся на экран векторы для сравнения результатов(рис.8)

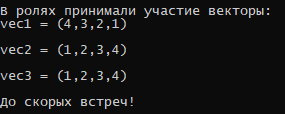


Рис.8 Текущие векторы

## Приложение для демонстрации работы матриц

1. Запустите приложение с названием sample\_tset.exe. В результате появится окно, показанное ниже (рис. 1).

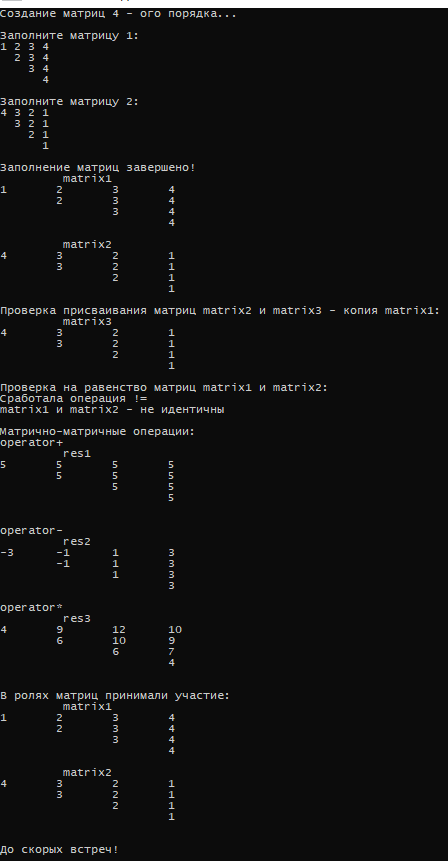


Рис.1 Окно основной программы

1. На первом шаге создаются матрицы указанного порядка (рис.2)



Рис.2. Процесс создания

1. Далее матрицы заполняются значениями того типа,который был указан до этапа создания матриц,и результаты ввода выводятся на экран (рис.3)

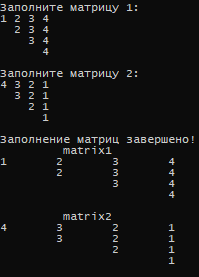


Рис.3 Заполнение матриц и вывод на экран

1. На представленном рисунке можно увидеть операции сравнения матриц(рис.4)



Рис.4 Сравнение матриц

1. На рис.5 приведены матричное-матричные операции(рис.5)

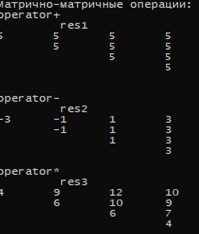


Рис.5 работа с матрицами

1. В завершение выводятся матрицы,которые принимали участие в работе(рис.6)

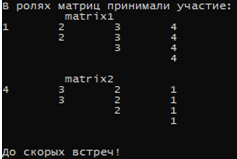


Рис.6 Вывод матриц

# Руководство программиста

## Описание алгоритмов

### Векторы

Шаблонный класс вектор представляет из себя С/Х – динамический одномерный массив с элементами типа Type(параметр шаблона). У каждого вектора есть свой стартовый индекс, с которого выделяется память для хранения элементов, и размер. Используется как интерфейс для работы с матрицами.

Методы:

Конструктор с параметрами по умолчанию.

Входные параметры: size – размер вектора, start\_index – стартовый индекс

Выходные параметры: отсутствуют

Алгоритм:

1. Выполняются проверки на отрицательность размера и максимально допустимое значение и на отрицательность стартового индекса.
2. Поля size и start\_index инициализируются значениями переданных параметров.
3. Выделяется память в количестве size под хранение элементов вектора.

Конструктор копирования

Входные данные: константная ссылка на объект типа TVector

Выходные данные: отсутствуют

Алгоритм:

1. Поля size и start\_index инициализируются значениями полей переданного объекта.
2. Выделяется память в количестве size под хранение элементов вектора
3. В цикле for творится копирование значение элементов из поля vector переданного объекта в поле vector текущего объекта типа TVector.

Деструктор

Входные данные: отсутствуют

Выходные данные: отсутствуют

Алгоритм:

1. Освобождает память из-под массива vector.

GetSize

Входные параметры: отсутствуют

Выходные параметры: size – значение поля size

Алгоритм:

Возврат значения поля size

GetStart

Входные параметры: отсутствуют

Выходные параметры: start\_index – значение поля start\_index

Алгоритм:

Возврат значения поля start\_index

Operator==

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: целое число – 0 или 1

Алгоритм:

1. Проверка на равенство размеров. В случае неравенства возвращает false.
2. Проверка на стартовый индекс. В случае неравенства возвращает false.
3. Поэлементное сравнение компонент вектора с элементами вектора переданного объекта. В случае несовпадения хотя бы с одним возвращается false.
4. Возвращается true,если все условия не выполняются

Operator !=

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: целое число – 0 или 1

Алгоритм:

Возвращается инвертированный результат операции ==

Operator=

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: ссылка на объект себя

Алгоритм:

1. Проверка стартовых индексов. В случае неравенства переприсваивание поля start\_index.
2. Проверка на равенство размеров. В случае неравенства удаляется старая память, поле size инициализируется новым значением.,память перевыделяется в количестве size.
3. Поэлементное присваивание компонент вектора текущего объекта и элементами вектора переданного объекта.
4. Возвращается \*this.

Operator[]

Входные данные: index – номер элемента

Выходные данные: элемент вектора

Алгоритм:

1. Проверка индекса на отрицательность и на обращение к мнимой части(невыделенная память). Бросается исключение в случае неуспешной проверки.
2. Возвращается номер элемента index-start\_index,так как не учитываем мнимую часть.

Operator+

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: новый объект класса типа TVector

Алгоритм:

1. Проверка на стартовые индексы. Бросается исключение в случае неравенства.
2. Проверка на равенство размеров. Бросается исключение в случае неравенства.
3. Создаётся копия текущего объекта под результат.
4. Выполняется поэлементное сложение элементов вектора текущего объекта с элементами переданного объекта типа TVector.
5. Возвращается результирующий объект типа TVector.

Operator-

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: новый объект класса типа TVector

Алгоритм:

1. Проверка на стартовые индексы. Бросается исключение в случае неравенства.
2. Проверка на равенство размеров. Бросается исключение в случае неравенства.
3. Создаётся копия текущего объекта под результат.
4. Выполняется поэлементная розность элементов вектора текущего объекта с элементами переданного объекта типа TVector.
5. Возвращается результирующий объект типа TVector.

Operator\*

Входные параметры: константная ссылка на объект типа TVector

Выходные параметры: новый объект класса типа TVector

Алгоритм:

1. Проверка на стартовые индексы. Бросается исключение в случае неравенства.
2. Проверка на равенство размеров. Бросается исключение в случае неравенства.
3. Создаётся копия текущего объекта под результат.
4. Выполняется поэлементное сложение пар произведений соответствующих элементов вектора текущего объекта с элементами переданного объекта типа TVector.
5. Возвращается результирующий объект типа TVector.

Operator>>

Входные параметры: istr-ссылка на стандартный поток ввода,obj – неконстантная ссылка на объект

Выходные параметры: istr- ссылка на стандартный поток ввода

Алгоритм:

1. Создается внешний цикл for для прохождения по элементам вектора
2. Внутренний цикл for начинает отсчёт от стартового индекса до размера вектора.
3. Выполняется заполнение на указанном промежутке.
4. Возвращается ссылка на стандартный поток ввода.

Operator<<

Входные параметры: ostr-ссылка на стандартный поток ввода,obj – константная ссылка на объект типа TVector

Выходные параметры: ostr- ссылка на стандартный поток вывода

Алгоритм:

1. Создается внешний цикл for для прохождения по элементам вектора
2. Внутренний цикл for начинает отсчёт от стартового индекса до размера вектора.
3. Выполняется вывод на указанном промежутке.
4. Возвращается ссылка на стандартный поток вывода.

### Матрицы

## Описание программной реализации

### Описание класса TVector

Назначение: представление битового поля.

Поля:

Методы:

Назначение:

Входные параметры:

Выходные параметры:

### Описание класса TMatrix

# Заключение

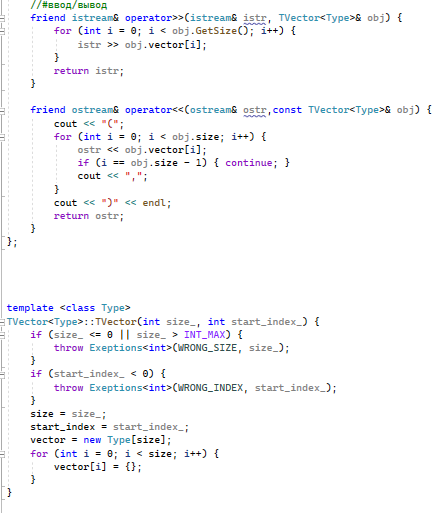
# Литература

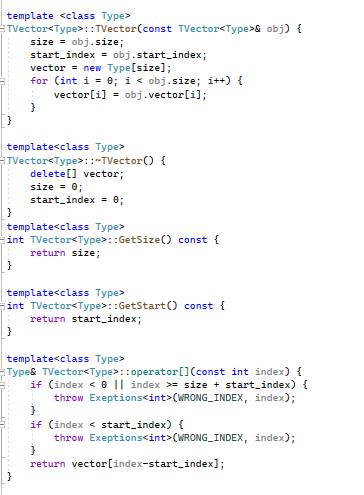
Н.В Гредасов, Линейная Алгебра – Издательство о Уральского университета. Редакционно-издательский отдел ИПЦ УрФУ 620049, 92 страницы ([с.6](https://elar.urfu.ru/bitstream/10995/78551/1/978-5-7996-2776-8_2019.pdf))

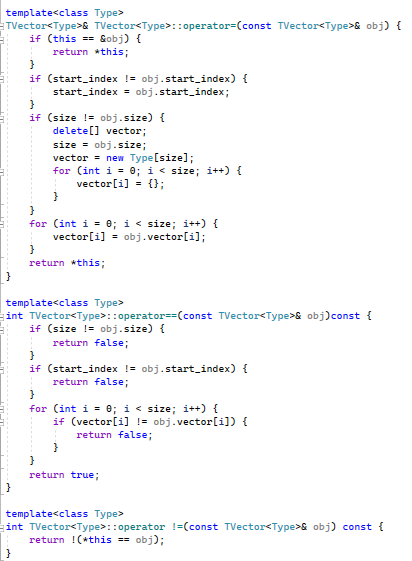
Алексей Померанцев, Векторы и матрицы – Российское Хемометрическое общество,33 раздела([р.6](https://www.chemometrics.ru/old/Tutorials/matrix.htm#ch106))

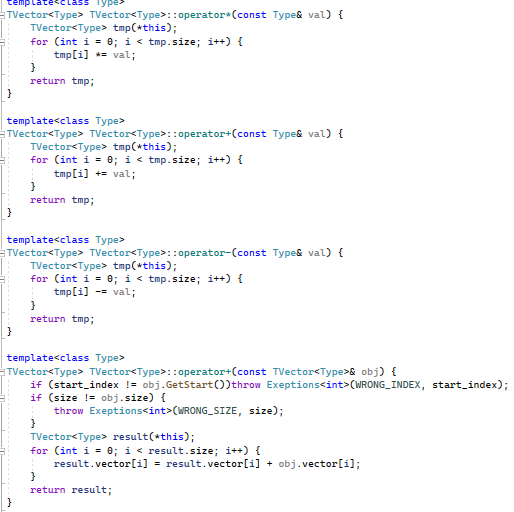
# Приложения

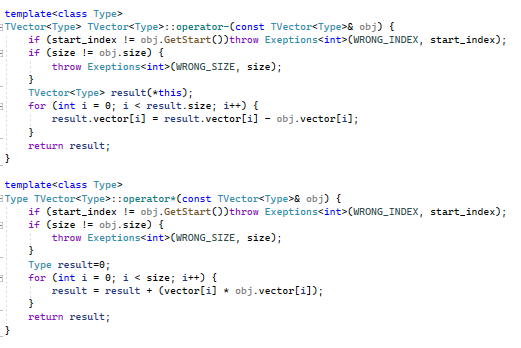
## Приложение А. Реализация класса TVector

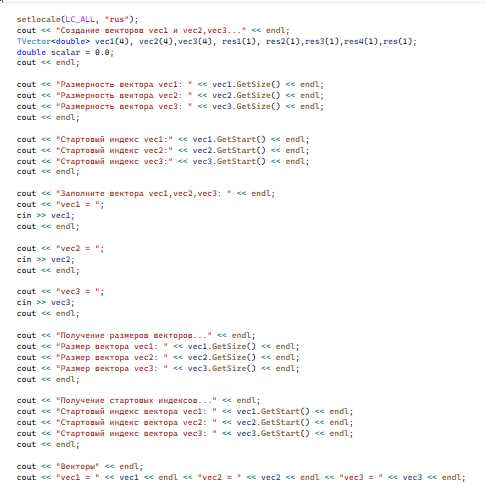


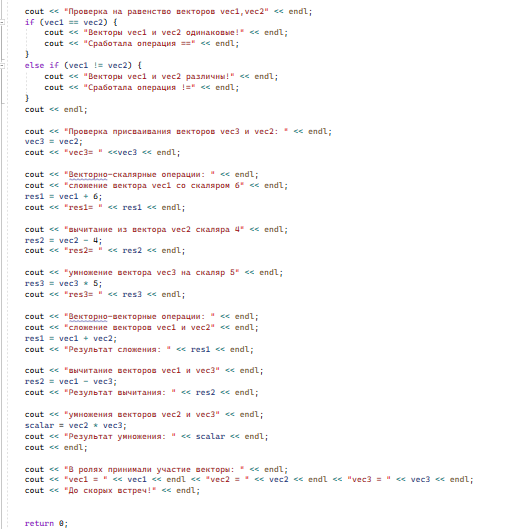












## Приложение Б. Реализация класса TMatrix

