МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

Кафедра информационных технологий и экономической информатики

ОТЧЕТ

по лабораторной работе №1

Выполнил:

Студент группы ПрИ-202 Приходько Даниил Александрович

Студент группы ПрИ-202 Саламатин Алексей

Студент группы ПрИ-202 Скоробогатов Максим

Принял:

Преподаватель ИИТ Николаев Иван Евгеньевич

Отчет защищен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата оценка

Челябинск 2024 г.

Содержание

Цель работы

Цель данной лабораторной работы — провести эмпирический анализ временной сложности алгоритмов. В рамках работы вводится понятие временной сложности алгоритма, рассматривается математический аппарат для оценки временной сложности и правила применения этого аппарата.

Задание

Для каждого n от 1 до 2000 произведите для пяти запусков замер среднего машинного времени исполнения программ, реализующих нижеуказанные алгоритмы и функции. Изобразите на графике полученные данные, отражающие зависимость среднего времени исполнения от n. Проведите теоретический анализ временной сложности рассматриваемых алгоритмов и сравните эмпирическую и теоретическую временные сложности.

Теоретическая часть

На деле был произведен уход от начально задания, так как объем входных данных был недостаточен для наглядной оценки временной сложности работы предоставленных алгоритмов. Длина входного вектора была изменена и теперь пользователь выбирает количество подаваемых на вход программе чисел в пределе от 1 до 1.000.000.000, для того чтобы увидеть прогрессию функции на большом размере входного вектора. Помимо этого также и количество запусков выбирается пользователем и варьируется от 1 до 10. Также пользоателю была предоставлена возможность самостоятельно выбирать максимальный размер случайного числа, в размере до 1.000.000. Для алгоритмов возведения в степень, есть возможность выбирать степени в пределах от 2 до 200. В алгоритме перемножения матриц можно выбрать размер стороны квадратной матрицы от 1 до 1.000.000.000. А в алгоритме Дейкстры предоставляется возможность указать количество вершин графа также от 1 до 1.000.000.000.

Блок схемы и оценка врменной сложности:

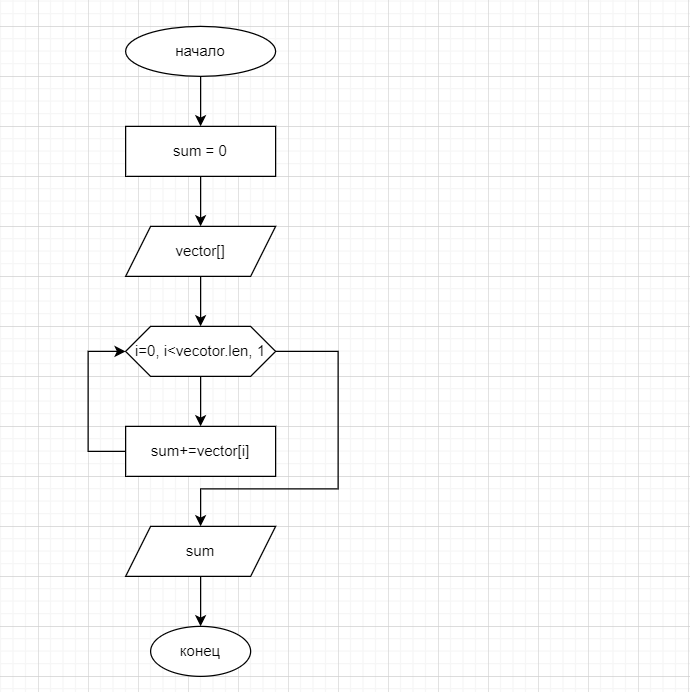
Блок 1

1) Постоянная функция – функция которая независимо от количества и размера входных данных возвращает 1.

Теоретическая сложность O(1) , т.к не зависит от n

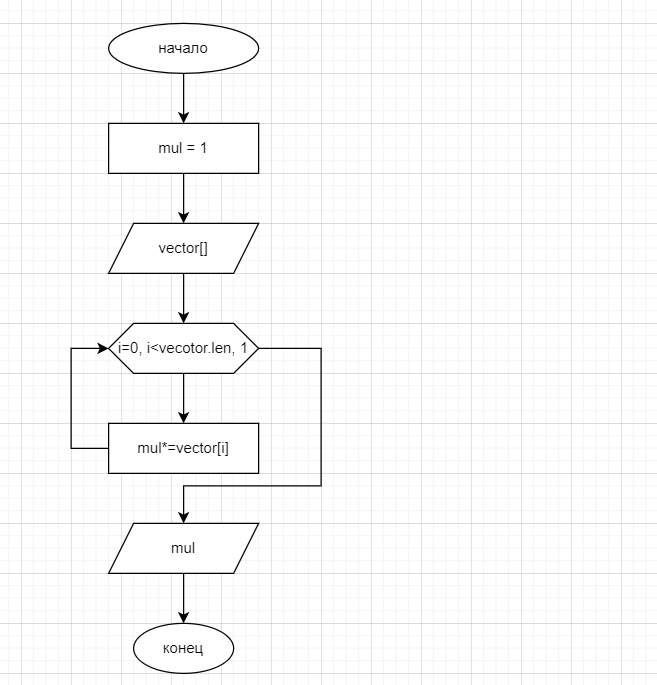
2) Сумма элементов – алгоритм, который перебором последовательно складывает все элементы вектора и возвращает их сумму

Теоретическая сложность O(n)



3) Произведение элементом - алгоритм, который перебором последовательно пееремножает все элементы вектора и возвращает их произведение

Теоретическая сложность O(n)



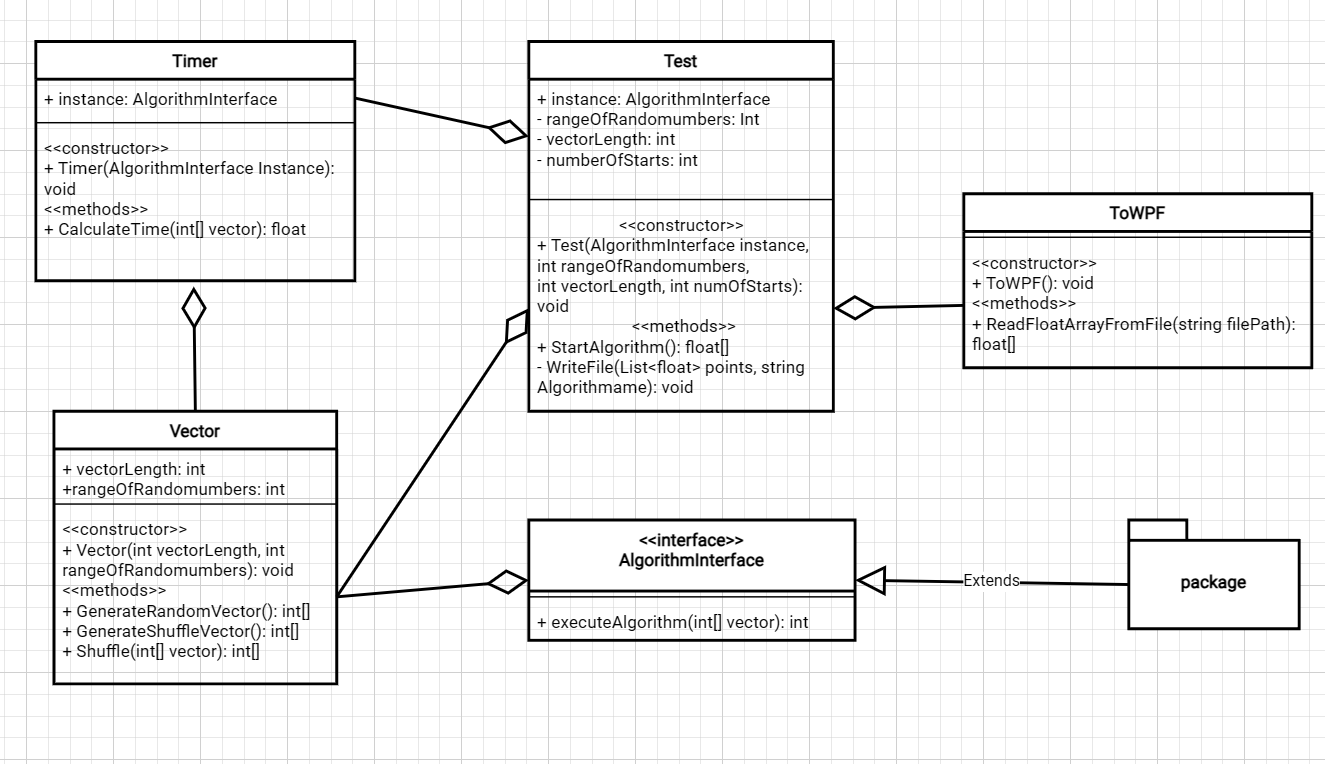
4) Метод Горнера -

Модель проекта

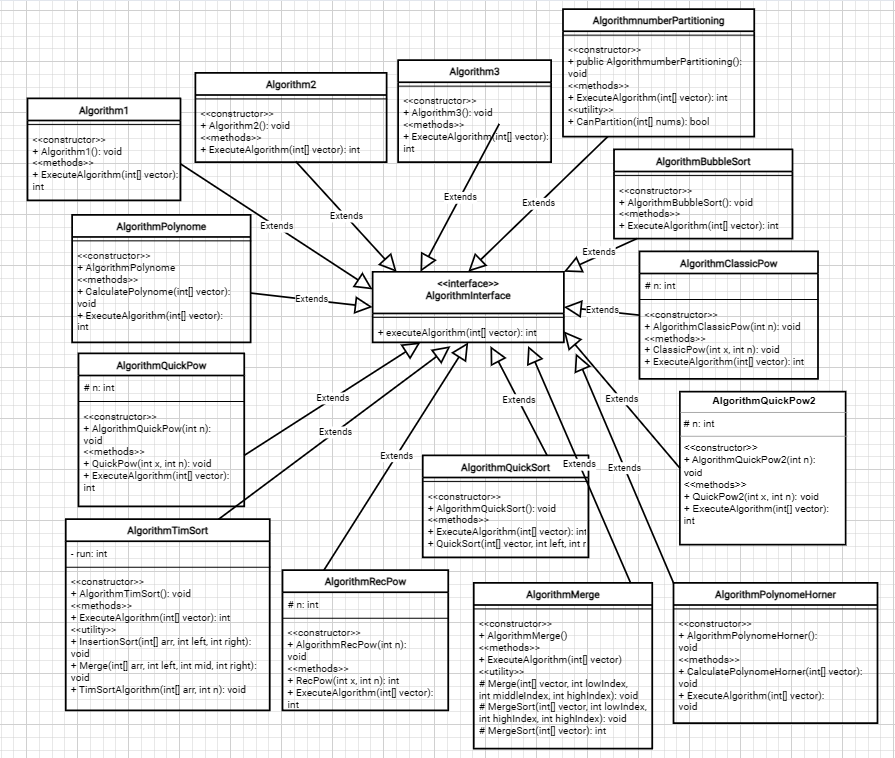
Все приложение работает на языке C#, для содания внешней оболочки использован WPF, для отрисовки графиков – ScottPlot.

Приложение поделено на 2 части – фронтенд и бэкенд. В основе бэкенда лежит три созданных библиотеки:

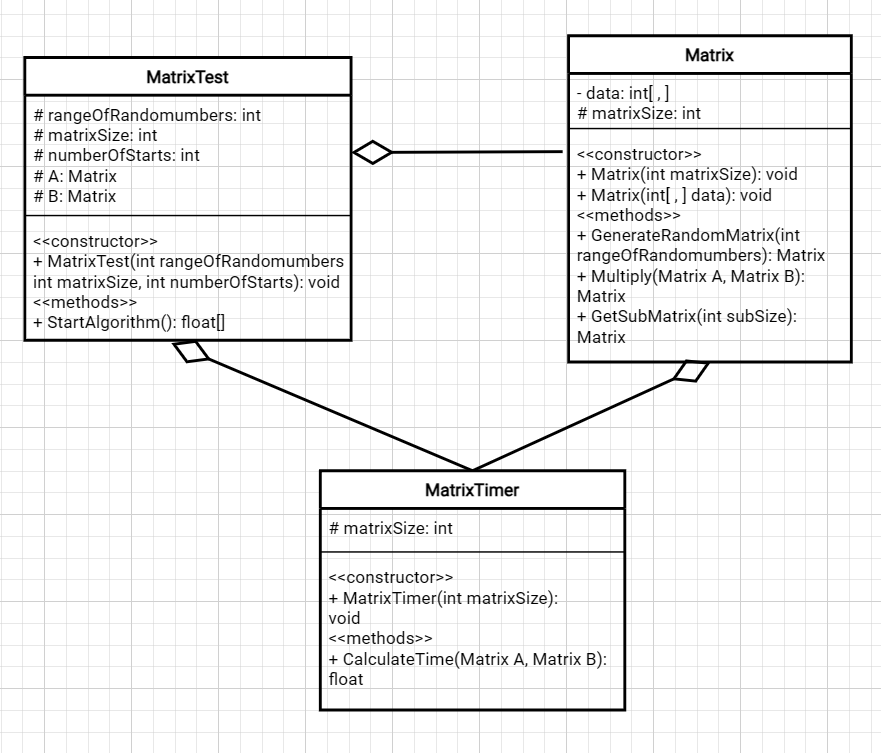
1) AlgLogic – необходима для работы с алгоритмами которым на вход подается вектор



Здесь под пакетом имеются все классы-наследники интерфейса



2) MatrixEntities – необходима для выполнения операций над матрицами



3) DijkstraAlgorithm – необходима для работы с графами и выполнения алгоритма Дейкстры.