

Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

|  |  |  |  |
| --- | --- | --- | --- |
| **Направление подготовки** | | 09.03.00 Информатика и вычислительная техника | |
| **Факультет** | | Информационных технологий и управления | |
| **Кафедра** | | Систем автоматизированного проектирования и управления | |
| **Курс**  **Группа** | | I  4303 | |
|  |  |  |  |

**КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ**

**«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ»**

|  |  |  |
| --- | --- | --- |
| **Тема:** | Программный комплекс для изучения основ полиморфизма на примере сортировки матриц (принцип сортировки: упорядочить каждый столбец матрицы по возрастанию абсолютных величин). | |
| Выполнил обучающийся | | Г. И. Иванов |
| Руководитель, доц. | | И. Г. Корниенко |
| Консультант, ст. преп. | | А. К. Федин |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc166948265)

[1 Аналитический обзор 4](#_Toc166948266)

[1.1 Матричные вычисления и компьютерные системы для работы с ними. Сравнительная характеристика существующих систем-аналогов. Обоснование актуальности разработки программного комплекса для изучения основ полиморфизма на примере сортировки матриц. 4](#_Toc166948267)

[1.2 Общая характеристика и особенности процесса сортировки матриц. 8](#_Toc166948268)

[1.3 Обзор и обоснование выбора инструментальных средств для разработки консольного приложения 8](#_Toc166948269)

[2 Цель и задачи курсового проекта 12](#_Toc166948270)

[3 Технологическая часть 13](#_Toc166948271)

[3.1 Формализованное описание процесса сортировки матриц как объекта обработки информации 13](#_Toc166948272)

[3.2 Постановка задачи обработки информации 13](#_Toc166948273)

[3.3 Разработка функциональной структуры программного комплекса для изучения основ полиморфизма на примере сортировки матриц 13](#_Toc166948274)

[3.4 Создание алгоритма сортировки матриц (принцип сортировки: упорядочить каждую нечетную строку матрицы по возрастанию суммы значений цифр элементов матрицы) 14](#_Toc166948275)

[3.5 Разработка структуры интерфейсов для пользователя программного комплекса 20](#_Toc166948276)

[3.6 Описание структур данных и алгоритмов (формат представления данных) 20](#_Toc166948277)

[3.7 Описание структуры программы (модули, основные функции) 21](#_Toc166948278)

[3.8 Тестирование программного комплекса (на примере ввода случайных значений) 24](#_Toc166948279)

[ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ 27](#_Toc166948280)

[ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ 29](#_Toc166948281)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 30](#_Toc166948282)

# ВВЕДЕНИЕ

Полиморфизм является одним из основополагающих принципов объектно-ориентированного программирования, который позволяет объектам различного типа вести себя схожим образом. Это повышает гибкость и повторное использование кода, делая его более поддерживаемым и масштабируемым. В реальных прикладных задачах, связанных с обработкой данных, полиморфизм играет важную роль, обеспечивая возможность обрабатывать разнообразные типы данных и динамически изменять поведение кода в зависимости от конкретных обстоятельств.

Данный курсовой проект посвящен исследованию полиморфизма в контексте сортировки матриц, а также проектирование и создание прикладного программного обеспечения, позволяющего решать задачу сортировки матриц, а также проводить сравнение методов сортировки. Кроме того, программа подразумевает собой возможность заполнения матриц вручную, случайными числами и с файла, а также сохранение данных в файл.

Матричные вычисления используются для решения различных задач в науке, технике, экономике и других областях. Ниже приведены некоторые основные применения матричных вычислений:

- матрицы применяются для операций с изображениями, анимацией и компьютерной графикой, так как они позволяют эффективно хранить и обрабатывать большие объемы данных.

- матричные вычисления используются в эконометрике для анализа временных рядов, оценки стохастических моделей и прогнозирования экономических показателей.

- матричные операции широко используются в алгоритмах машинного обучения, таких как метод главных компонент, линейная регрессия, метод опорных векторов и нейронные сети.

Для реализации проекта будет использоваться язык программирования C++.

В результате успешной реализации данного проекта пользователи смогут иметь доступ к консольному приложению, который даст им возможность быстро и удобно выполнять различные операции сортировки.

# 1 Аналитический обзор

**1.1** Матричные вычисления и компьютерные системы для работы с ними. Сравнительная характеристика существующих систем-аналогов. Обоснование актуальности разработки программного комплекса для изучения основ полиморфизма на примере сортировки матриц.

Консольное приложение предназначено для использования пользователями, чтобы получить мгновенные результаты. Оно облегчает выполнение процессов сортировки без непосредственного вмешательства человека.

При использовании консольного приложения происходит обработка введённых пользователем или программой данных. Система анализирует и обрабатывает полученные данные, затем выполняет соответствующие вычисления. Интерфейс консольного приложения представляет собой пункты меню, с которыми необходимо взаимодействовать при помощи ввода соответствующих чисел с клавиатуры.

Существует несколько систем для работы с матрицами, которые обладают различными характеристиками и возможностями. Ниже приведена сравнительная характеристика трёх найденных из них:

Таблица 1 – Сравнение существующих приложений и веб-сайтов

|  |  |  |  |
| --- | --- | --- | --- |
| Функции приложений | Matrix calculator  <https://matrixcalc.org> | Mathcad 15 (M050) | Maple 7 |
| Возможность ввода данных с клавиатуры | + | + | + |
| Перемножение матриц | + | + | + |
| Возможность ввода данных при помощи файла | - | + | + |
| Транспонирование матрицы | + | + | + |
| Лицензия | - | + | + |

Существующие системы для работы с матрицами представлены на рисунках 1, 2, 3.



Рисунок 1 – Веб-сайт Matrix calculator

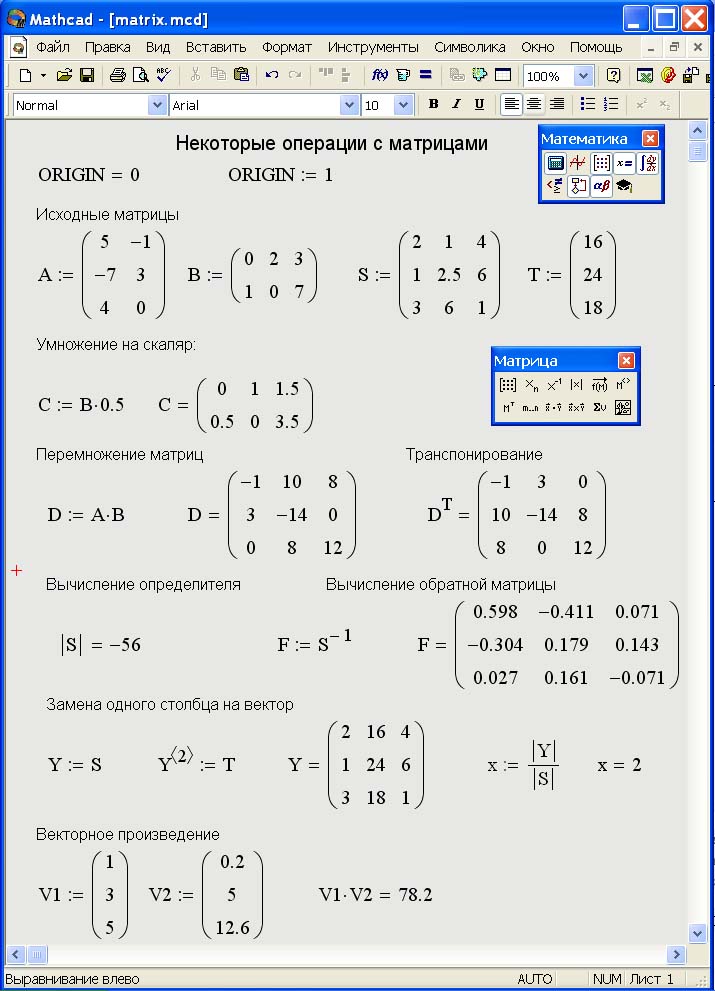


Рисунок 2 – Приложение Mathcad

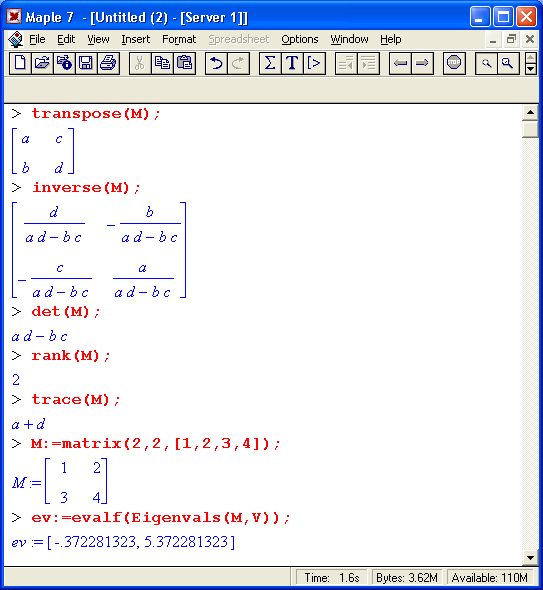


Рисунок 3 – Приложение Maple

Разработка консольного приложения является актуальной задачей по нескольким причинам:

1. образовательная значимость: программный комплекс будет служить эффективным инструментом для обучения студентов концепциям полиморфизма, что является ключевым аспектом объектно-ориентированного программирования. Понимание полиморфизма необходимо для разработки гибких и масштабируемых программных решений.
2. научная и практическая ценность: проект может быть использован в качестве основы для дальнейших научных исследований в области оптимизации вычислений и алгоритмов. Кроме того, разработанный комплекс может быть интегрирован в учебный процесс в вузах, что повысит качество образования в области информационных технологий.
3. сортировка матриц по заданному принципу может найти применение в различных областях, включая компьютерное зрение и обработку данных, где требуется предварительная обработка и упорядочивание информации.

Эти пункты подчеркивают важность проекта как с точки зрения образования, так и с точки зрения практического применения и научных исследований.

1.2 Общая характеристика и особенности процесса сортировки матриц.

Ознакомившись с функционалом существующих систем-аналогов, было принято решение реализовать в проекте следующий функционал:

* 1. интуитивно-понятный интерфейс,
  2. возможность ввода данных с клавиатуры,
  3. возможность заполнением случайными числами,
  4. возможность ввода данных при помощи файла,
  5. возможность сохранения результата в файл.

Особенности процесса сортировки матриц:

1. цикл сортировки должен работать только по нечетным строкам матрицы.
2. внутри каждой нечетной строки происходит сравнение суммы цифр элементов.

1.3 Обзор и обоснование выбора инструментальных средств для разработки консольного приложения

Для реализации поставленной задачи был выбрал язык программирования C++, располагающий всеми необходимыми инструментами для реализации данного проекта. В качестве интегрированной среды разработки программного обеспечения (Integrated Development Environment, IDE) используется Microsoft Visual Studio 2022.

Также необходимо выбрать оптимальную среду разработки. Сравним среду разработки Visual Studio 2022 с Eclipse и Xcode, используя несколько ключевых критериев, и рассмотрим, почему Visual Studio 2022 может быть предпочтительным выбором для поставленной задачи:

1. интеграция с C++: Visual Studio 2022 и Xcode предлагают встроенную поддержку C++, в то время как для работы с C++ в Eclipse требуется установка дополнительных плагинов. Это делает Visual Studio и Xcode более удобными для проектов на C++.
2. отладка: Visual Studio имеет один из лучших в индустрии отладчиков, который предоставляет мощные возможности для диагностики и устранения ошибок в коде. Eclipse и Xcode также предлагают хорошие инструменты отладки, но они могут быть менее интуитивными по сравнению с Visual Studio.
3. пользовательский интерфейс: Visual Studio и Xcode имеют современный и интуитивно понятный интерфейс, который может быть более привлекательным для разработчиков, в то время как пользовательский интерфейс Eclipse может показаться устаревшим и перегруженным.
4. производительность: Visual Studio обычно обеспечивает более высокую производительность и меньше потребляет системных ресурсов по сравнению с конкурентами, что может быть критично при работе с большими проектами.
5. поддержка Windows: Xcode – среда разработки только для пользователей MacOS и iOS, в то время как для работы в Visual Studio и Eclipse можно использовать обе операционные системы.

Таким образом, при выборе среды разработки для курсового проекта на языке C++ рекомендуется обратить внимание на Visual Studio 2022, как на инструмент с обширным функционалом и хорошей поддержкой языка C++.

Среды разработки представлены ниже на рисунках 4, 5 и 6.

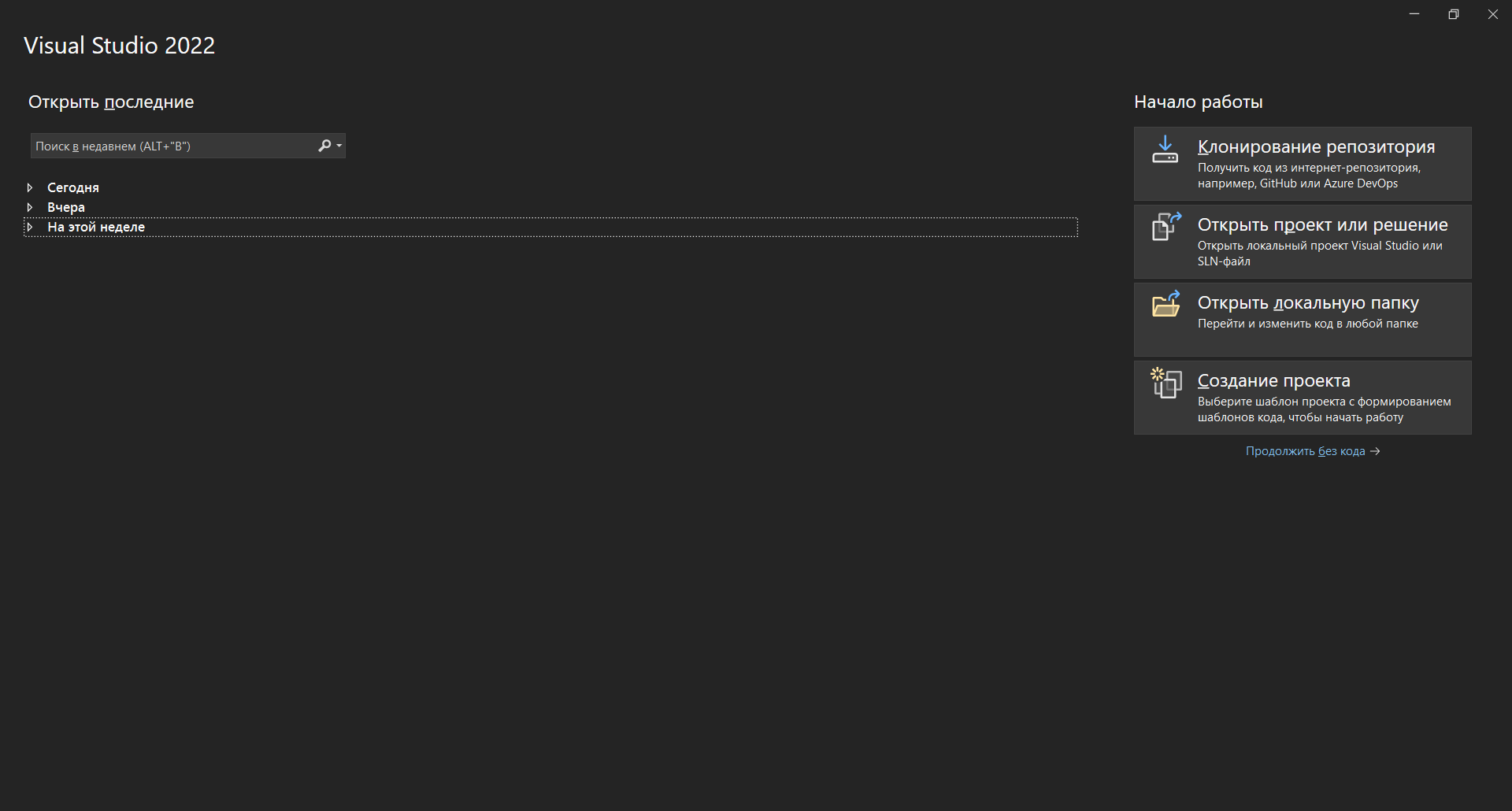


Рисунок 4 – Среда разработки Visual Studio 2022

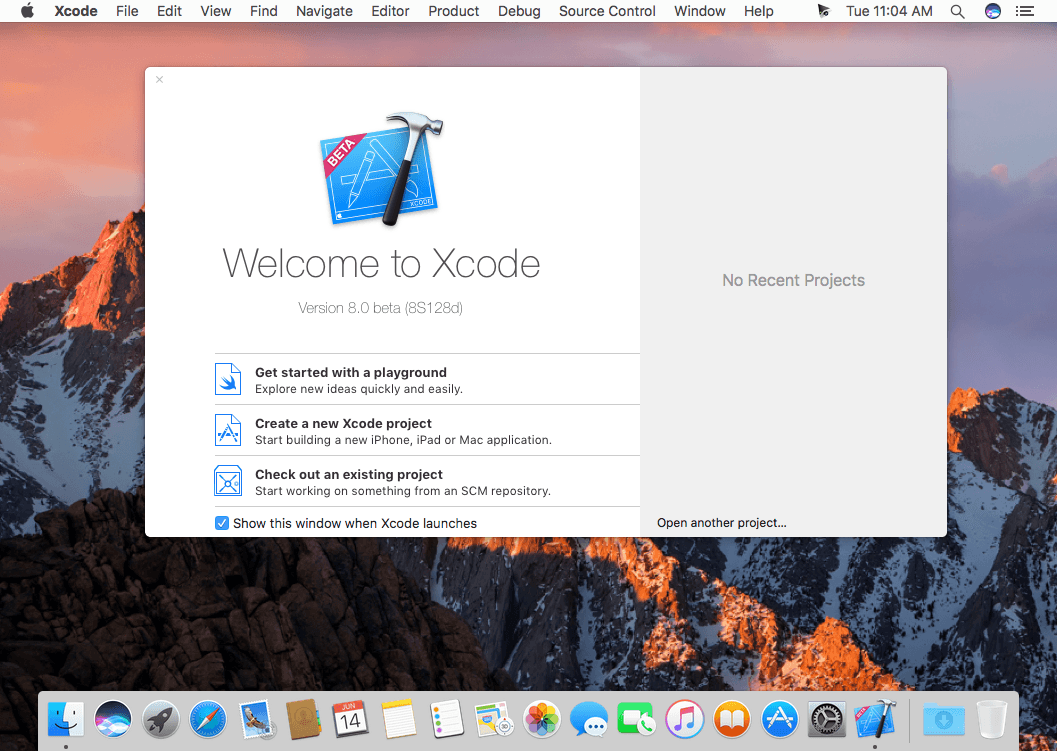


Рисунок 5 – Среда разработки Xcode

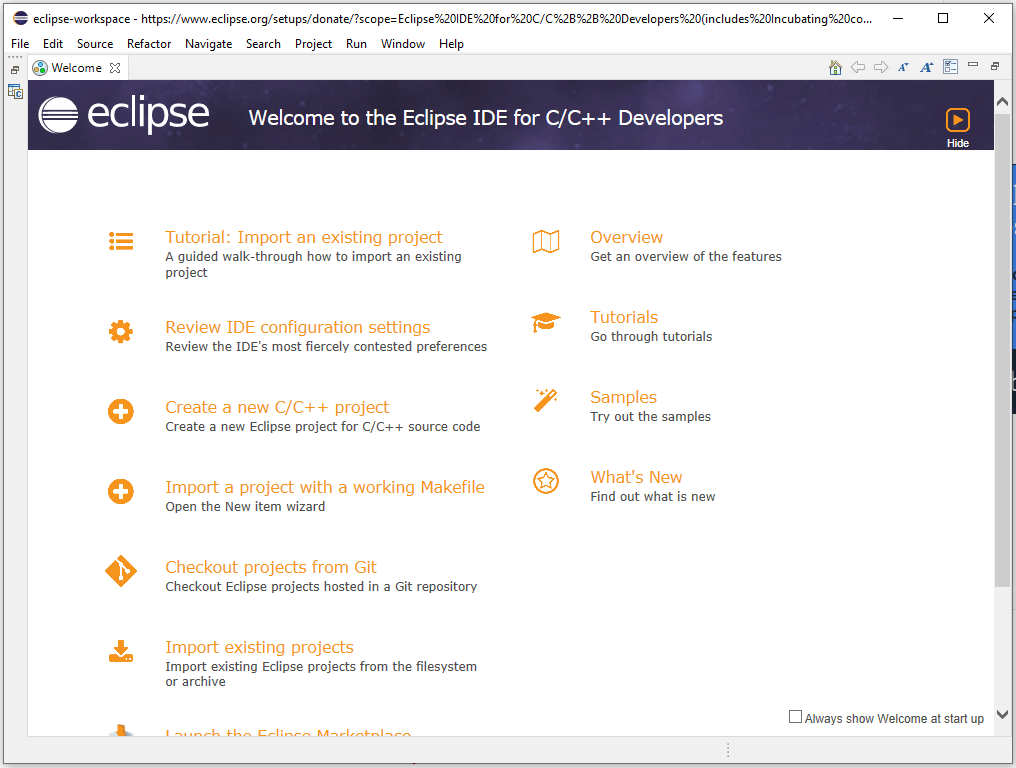


Рисунок 6 – Среда разработки Eclipse

# 2 Цель и задачи курсового проекта

Целью работы является проектирование и создание прикладного программного обеспечения, позволяющего решать задачу сортировки динамических матриц, а также проводить сравнение методов сортировки.

Для выполнения задачи были использованы несколько методов сортировок, а также реализация сравнительной таблицы для каждой из них. Используемые методы сортировки: пузырьковая, выбора, вставки, Шелла, быстрая.

Для достижения поставленной цели были определены следующие задачи:

- формализованное описание процесса сортировки матриц как объекта обработки информации.

- постановка задачи обработки информации.

- разработка функциональной структуры программного комплекса для изучения основ полиморфизма на примере сортировки матриц.

- создание алгоритма сортировки матриц (упорядочить каждую нечетную строку матрицы по возрастанию суммы значений цифр элементов матрицы).

- разработка структуры интерфейсов для пользователя программного комплекса.

- описание структур данных и алгоритмов (формат представления данных).

- описание структуры программы (формат представления данных).

- тестирование программного комплекса (на заданном примере).

- оформление документации (пояснительной записки) по проекту.

# 3 Технологическая часть

3.1 Формализованное описание процесса сортировки матриц как объекта обработки информации

Для реализации представленного проекта одной из задач являлась необходимость иметь представление о моделируемом объекте, а именно о том, как представить процесс сортировки матриц, способы ввода и вывода полученных данных, в виде взаимодействия программного интерфейса с пользователем.

При моделировании процесса сортировки, как объекта обработки информации, было разработано формализованное описание процесса (рисунок 7).



Рисунок 7 – Формализованное описание процесса сортировки матриц как объекта обработки информации

3.2 Постановка задачи обработки информации

Постановка задачи обработки информации состоит в создании алгоритма обработки полученных от пользователя данных, их проверке и обработке, выполнении необходимых операций сортировки и отображении результата в консольном приложении в виде отсортированной матрицы различными методами, а также сравнительной таблицы методов сортировки.

3.3 Разработка функциональной структуры программного комплекса для изучения основ полиморфизма на примере сортировки матриц

В процессе разработки программного комплекса была разработана функциональная структура (рисунок 8).



Рисунок 8 – Функциональная структура программного комплекса

3.4 Создание алгоритма сортировки матриц (принцип сортировки: упорядочить каждую нечетную строку матрицы по возрастанию суммы значений цифр элементов матрицы)

Для поставленной задачи необходимо было использовать 5 алгоритмов сортировки: пузырьковой, выбора, вставки, Шелла и быстрой (рисунки 9-13).



Рисунок 9 – Блок-схема алгоритма пузырьковой сортировки



Рисунок 10 – Блок-схема алгоритма сортировки выбором



Рисунок 11 – Блок-схема алгоритма сортировки вставками



Рисунок 12 – Блок-схема алгоритма сортировки Шелла



Рисунок 13 – Блок-схема алгоритма быстрой сортировки

3.5 Разработка структуры интерфейсов для пользователя программного комплекса

На рисунке 15 представлена структура интерфейса программного комплекса для пользователя.



Рисунок 14 – Структура интерфейса программного комплекса

3.6 Описание структур данных и алгоритмов (формат представления данных)

Программа использует следующие основные переменные:

Таблица 2 – Основные пользовательские переменные программы:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| command | int | Пользовательский тип данных, предназначенный для выбора пункта главного меню |
| n | int | Переменная содержит значение количества строк матрицы |
| m | int | Переменная содержит значение количества столбцов матрицы |
| filename | string | Переменная хранит в себе путь или название файла |

Таблица 3 – Основные переменные, используемые в базовом классе ISort

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| comparisons | int | Переменная содержит в себе количество сравнений во время процесса сортировки |
| swaps | int | Переменная содержит в себе количество перестановок во время процесса сортировки |

3.7 Описание структуры программы (модули, основные функции)

В силу своей сложности программа помещена в несколько исполняемых модулей.

Каждый модуль имеет свой уникальный набор функций. Ниже представлены таблицы с названием и описанием функций.

Таблица 4 – Функции составляющие модуль Interface

|  |  |
| --- | --- |
| Имя | Описание |
| Function | Функция, представляющая пункты главного меню для пользовательского интерфейса |
| ShowGreeting | Функция, выводящая на экран приветствие, информацию о студенте, группе, варианте и задании курсовой работы |

Таблица 5 – Функции, составляющие модуль Functions

|  |  |
| --- | --- |
| Имя | Описание |
| Transponding | Транспонирует матрицу. |
| Display | Отображает матрицу на экране. |
| HandFilling | Заполняет матрицу вручную. |
| RandomFilling | Заполняет матрицу случайными числами. |
| Filling | Выбирает способ заполнения матрицы. |

Таблица 6 – Функции, составляющие модуль Sort

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | | Описание | |
| Sort (ISort) | | Виртуальная функция для сортировки матрицы. | |
| Get\_swap\_counter | | Возвращает количество перестановок. | |
| Get\_comparison\_counter | Возвращает количество сравнений. | |
| Show\_counters | Показывает количество перестановок и сравнений. | |
| Reset\_counters | Сбрасывает счетчики перестановок и сравнений. | |
| ~ISort | Виртуальный деструктор класса ISort. | |
| Sort (BubbleSort) | Реализует сортировку матрицы методом пузырьковой сортировки. | |
| Bubble\_sort | Сортирует одномерный массив методом пузырьковой сортировки. | |
| Sort (QSort) | Реализует сортировку матрицы методом быстрой сортировки. | |
| Partition | Разделяет массив для быстрой сортировки. | |
| QuickSort | Сортирует одномерный массив методом быстрой сортировки. | |
| Sort (SelectionSort) | Реализует сортировку матрицы методом сортировки выбором. | |
| Selection\_sort | Сортирует одномерный массив методом сортировки выбором. | |
| Sort (ShellSort) | Реализует сортировку матрицы методом сортировки Шелла. | |
| Shell\_sort | Сортирует одномерный массив методом сортировки Шелла. | |

Таблица 7 – Функции, составляющие модуль filework

|  |  |
| --- | --- |
| Имя | Описание |
| FileIsEmpty | Проверяет, пуст ли файл. |
| OpenFile | Открывает файл для чтения. |
| OpenFile | Открывает файл для записи. |
| FileFilling | Заполняет матрицу из файла. |
| TestFileFilling | Заполняет матрицу из указанного файла для тестирования. |
| FileInput | Записывает матрицу в файл. |

3.8 Тестирование программного комплекса (на примере ввода случайных значений)

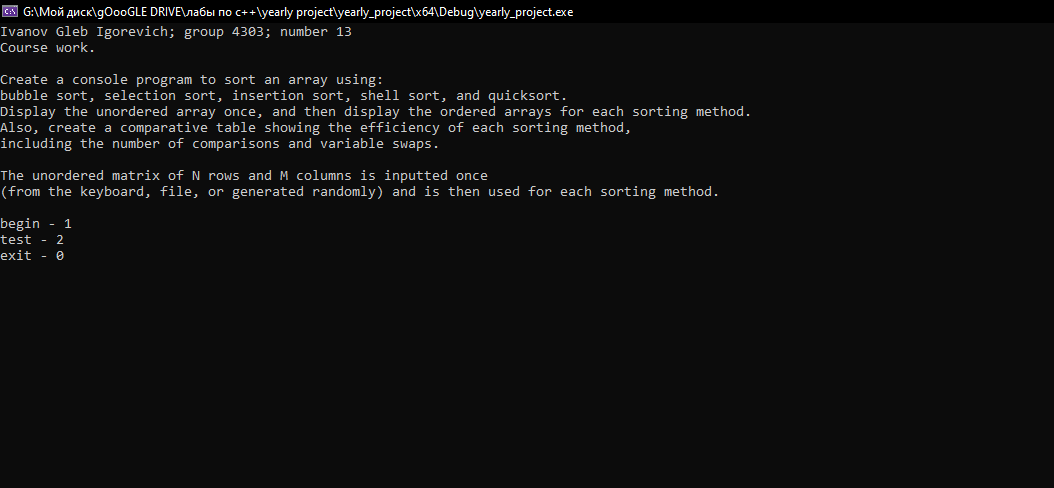


Рисунок 15 - Запуск программы

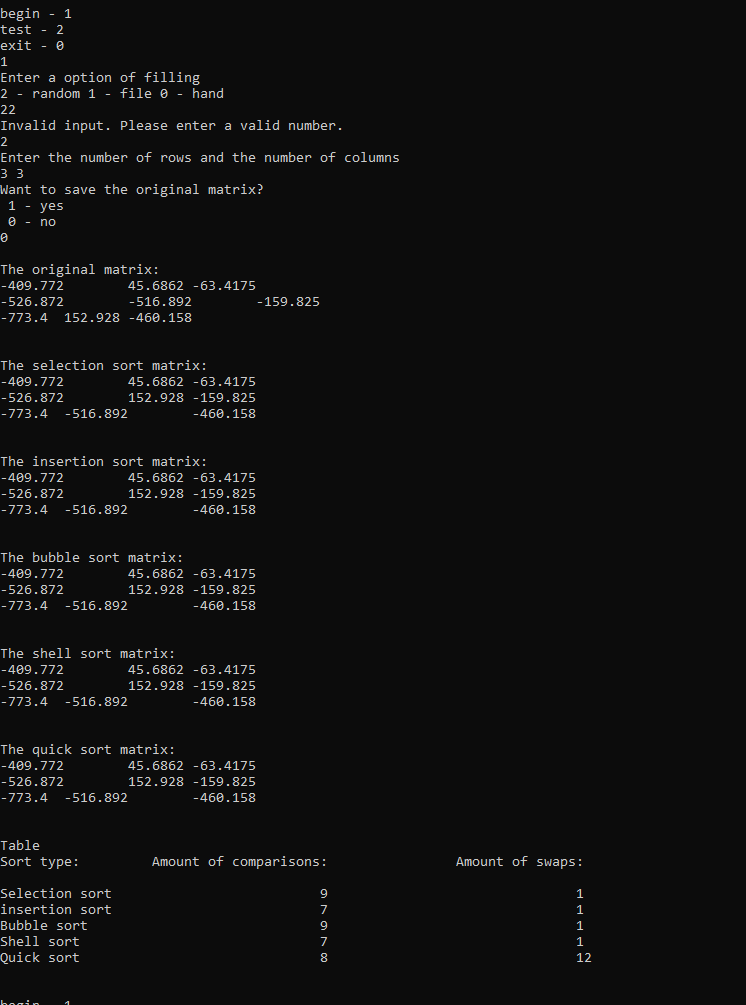


Рисунок 16 - Заполнение матрицы случайными числами и вывод результата

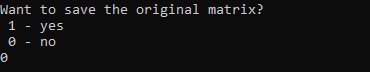


Рисунок 17 - Сохранение данных в файл

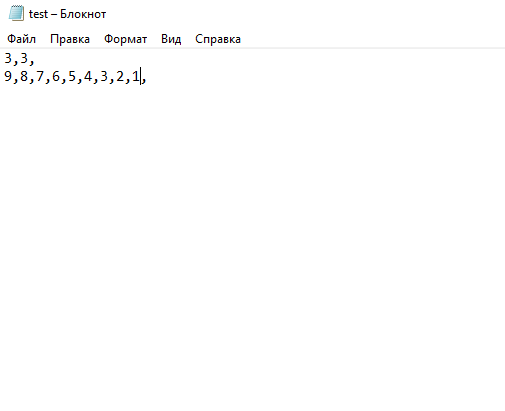
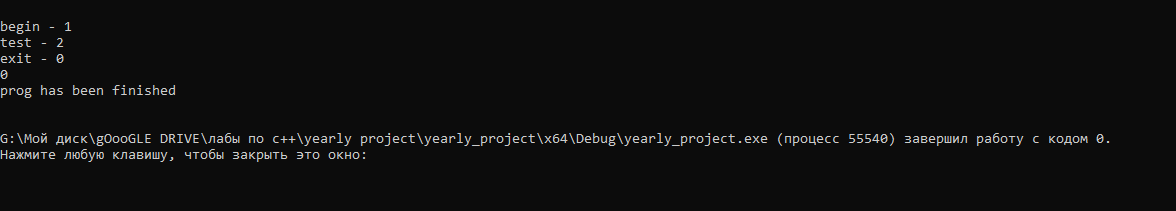


Рисунок 18 - Сохранённые данные

Рисунок 19 - Выход из программы

# ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ

Консольное приложение разработано под управлением операционной системы Windows 10. Средой разработки является Microsoft Visual Studio 2022. Структура программного обеспечения представлена на рисунке 21.



Рисунок 20 – Трёхуровневая структура программного обеспечения

Таблица 8 – Характеристика программного обеспечения

| Показатель | Значение |
| --- | --- |
| Среда разработки | Microsoft Visual Studio 2022 |
| Технология программирования | Объектно-ориентированное программирование |
| Язык программирования | C++ |
| Количество входных переменных | 6 |
| Количество внутренних переменных | 39 |
| Количество выходных переменных | 10 |
| Количество классов, структур | 6 |
| Количество функций | 37 |

Требования к ЭВМ, необходимой для нормального функционирования программы представлены в таблице 9.

Таблица 9 – Минимальные системные требования

| Показатель | Значение |
| --- | --- |
| Тип ЭВМ | Персональный компьютер |
| Тактовая частота процессора, ГГц | 2 |
| Объем оперативной памяти, ГБ | 8 |
| Объем внешней памяти, ГБ | 176 |
| Состав и характеристика периферийных устройств ЭВМ | Клавиатура, мышь, монитор с разрешением 1920 × 1080 пикселей |
| Операционная система | Windows 10 |

# ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ

В рамках курсового проекта был разработан программный комплекс, демонстрирующий принципы полиморфизма на примере сортировки матриц. Для достижения цели были реализованы и сравнены пять методов сортировки: пузырьковая, выбором, вставки, Шелла и быстрая.

Сравнительный анализ показал, что быстрая сортировка в большинстве случаев является наиболее эффективным методом для обработки матриц большого размера. Сортировка Шелла также показала хорошие результаты, особенно при сортировке матриц среднего размера. Традиционные методы, такие как пузырьковая, выбором и вставками, оказались менее эффективными, но их простота и наглядность были полезны для демонстрации основных концепций полиморфизма.

Разработанное консольное приложение успешно демонстрирует использование различных алгоритмов сортировки и принципов полиморфизма.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 928 с. – ISBN 978-5-4461-0927-2.

2 Липпман, С. Б. Язык программирования С++. Базовый курс: книга / С. Б. Липпман, Ж. Лажойе, Б. Э. Му. – 5-е изд. – Москва : Вильямс, 2019. – 1120 с. – ISBN 987-5-8459-1839-0.

3 Хомоненко, А. Д. Visual C++ на примерах: книга / А. Д. Хомоненко, Г. Ф. Довбуш. – Санкт-Петербург : БХВ-Петербург, 2007. – 528 с. – ISBN 978-5-94157-918-1.

4 Павловская, Т. А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. — Санкт-Петербург: Питер, 2021. — 461 с. – ISBN 978-5-4461-3916-3.

5 Прата, С. Язык программирования С++. Лекции и упражнения / С. Прата. – 6-е издание - Москва : Диалектика (Вильямс), 2020. – 1248 с. – ISBN 978-5-907114-00-5.