

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)**

Факультет компьютерных технологий и прикладной математики

Кафедра информационных технологий

ОТЧЕТ

о выполнении лабораторной работы № 8

по дисциплине «Программная инженерия»

Выполнила: ст. гр. МО 32/2

Переузник В. С.

Проверил: доцент каф. ИТ

Полетайкин А. Н.

Краснодар

2025

Тема: разработка специального программного обеспечения.

Индивидуальная тема: «Программа для аппроксимации функции».

Цель: приобретение навыков разработки и отладки программного обеспечения ПП при помощи современных технологий программирования.

Задание: 1) В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы 3, провести обоснованный выбор средства разработки специального ПО; 2) В соответствии с требованиями технического задания, разработанного при выполнении лабораторной работы 3, а также проектными решениями, разработанными при выполнении лабораторных работ 4–7, разработать специальное программное обеспечение ПП; 3) Осуществить пробный запуск приложения и убедиться в соответствии результатов требованиям, установленным в техническом задании. При обнаружении логических ошибок задокументировать их и устранить; 4) Представить экранные формы компонентов приложения в режиме запуска, а также отчетов, отражающих выходные данные; 5) Выполнить описание разработанных компонентов интерфейса приложения и отчетов в виде таблицы; 6) Проанализировать исходные тексты программных компонентов приложения по критерию сложности. В качестве критериев использовать: число модулей (классов) приложения, суммарное число переменных подпрограмм (методов классов), включая их формальные параметры, суммарное количество операторов подпрограмм (методов классов), суммарное число строк текста программы, суммарное число запросов к базе данных (при наличии), глубину вложенности структурных операторов ветвления, повторения, рекурсий, глубину наследования классов; 7) Выполнить профилирование и рефакторинг исходных текстов программных компонентов. Оценить сложность модифицированных текстов программных компонентов.

Ход выполнения работы:

1. Выбор средства разработки специального ПО

Для создания программы аппроксимации функций я рассмотрела два современных подхода к веб-разработке. Сравнивались связка JavaScript с HTML/CSS и более сложный вариант - TypeScript с фреймворком Angular.

JavaScript вместе с HTML/CSS — это основа веб-разработки. Программа будет работать в любом современном браузере без дополнительных преобразований. Главные плюсы этого подхода - простота запуска, возможность напрямую управлять элементами страницы и доступность полезных библиотек. Например, Chart.js для построения графиков и Math.js для математических расчетов. Особенность JavaScript - гибкая работа с данными, но иногда ошибки можно заметить только при запуске программы.

TypeScript с Angular — это более строгий и организованный подход. Код проверяется на ошибки перед запуском, а фреймворк предоставляет готовую структуру приложения. Это помогает избежать многих ошибок при разработке и удобно для тестирования. Однако такой подход требует больше времени для изучения и создает излишнюю сложность для небольших проектов.

После анализа обоих вариантов я выбрала JavaScript с HTML/CSS. Этот выбор обусловлен соответствием масштабу моего проекта, простотой архитектуры, легким размещением на сервер GitHub Pages и оптимальным соотношением простоты и эффективности для учебной задачи. TypeScript с Angular, несмотря на свои преимущества для крупных проектов, оказался избыточным для реализации аппроксимации функций.

2. Разработка специального ПО

В соответствии с техническим заданием и проектными решениями предыдущих лабораторных работ разработано специальное программное обеспечение для аппроксимации функций. Реализованы все компоненты, запроектированные в лабораторной работе №7.

В процессе разработки выявлены и устранены следующие логические ошибки:

1. Ошибка обработки CSV-файлов. Программа некорректно обрабатывала CSV-файлы с пустыми строками, пропущенными координатами или некорректными символами. В качестве решения была реализована поэтапная валидация данных - строки с ошибками игнорируются, учитываются только корректные пары чисел.

2. Ошибка расчета экспоненциальной функции. При аппроксимации экспоненциальной функцией возникали ошибки для отрицательных значений Y . В связи с этим была добавлена проверка, что все значения $Y > 0$ для экспоненциальной аппроксимации, с выводом предупреждающего сообщения.
3. Ошибка обработки одинаковых значений X . При вводе одинаковых значений X или идентичных пар (X, Y) программа зависала. Поэтому была реализована проверка на уникальность данных и обработка вырожденных случаев в матричных вычислениях.

Разработанное веб-приложение имеет одностраничный интерфейс, разделенный на две функциональные области. Левая часть предназначена для ввода и управления экспериментальными данными, а правая - для визуализации результатов аппроксимации. Все элементы интерфейса, представленные в таблице 2.1, реализованы в соответствии с техническим заданием и обеспечивают интуитивно понятное взаимодействие с пользователем.

№ п/п	Имя компонента	Описание	Рисунок
1	Общий вид интерфейса	Основное окно приложения с разделением на левую панель ввода данных и правую панель результатов	Рис. 2.2
2	Панель ввода данных	Левая часть интерфейса: таблица для ввода точек, кнопки управления и выбор типа функции	Рис. 2.3
3	Панель результатов	Правая часть интерфейса: отображает график, формулу функции и коэффициенты после расчета	Рис. 2.4

Таблица 2.1 - Перечень разработанных компонентов приложения

На рисунках ниже представлены экранные формы разработанного приложения, демонстрирующие основные элементы интерфейса и результаты работы программы.

Программа для аппроксимации функции

Ввод данных

Добавить строку

Очистить данные

Импорт CSV

№	X	Y	Действия
1			Удалить
2			Удалить
3			Удалить
4			Удалить
5			Удалить
6			Удалить

Выбор модели

Тип функции:

Линейная ($y = ax + b$)

Выполнить расчет

Отчет

Введите данные и нажмите "Выполнить расчет"

Рис. 2.2 - Общий вид интерфейса программы

Ввод данных

Добавить строку

Очистить данные

Импорт CSV

№	X	Y	Действия
1	3	11	Удалить
2	6	38	Удалить
3	9	83	Удалить
4	12	146	Удалить
5	15	227	Удалить
6	18	326	Удалить

Выбор модели

Тип функции:

Квадратичная ($y = ax^2 + bx + c$)

Выполнить расчет

Рис. 2.3 - Панель ввода экспериментальных данных

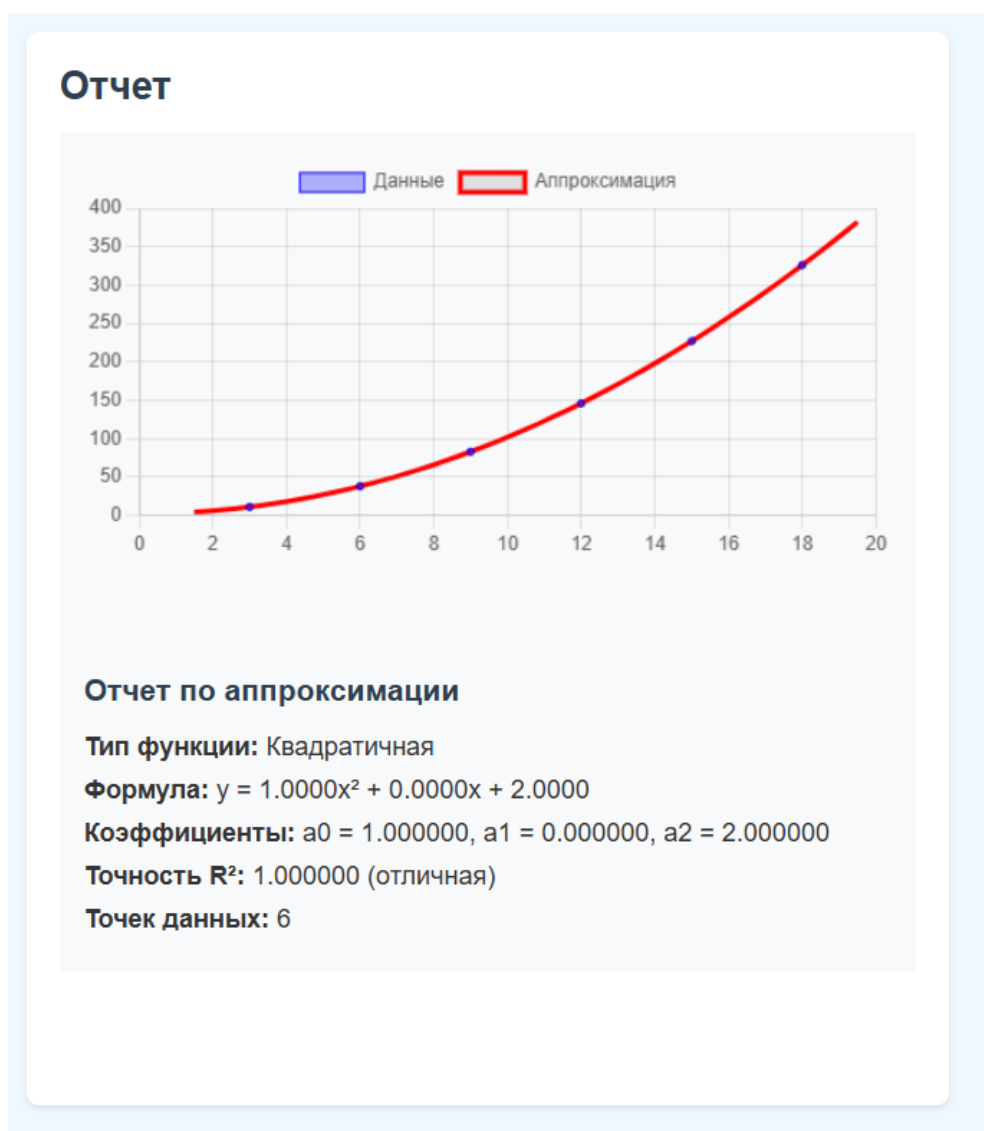


Рис. 2.4 - Панель результатов с графиком и формулой

Исходный код программы размещен в репозитории GitHub: <https://github.com/Vikkkaaa123/Approximation.git>

Программа состоит из нескольких модулей, каждый из которых отвечает за определенную функциональность. Основной файл `app.js` управляет работой всего приложения и координирует взаимодействие между компонентами. Модуль `data-manager.js` обрабатывает ввод и валидацию данных, включая работу с CSV-файлами. Математические расчеты выполняются в `math-processor.js`, где реализован метод наименьших квадратов для различных типов функций. Визуальная часть реализована с помощью библиотеки `Chart.js`, которая интегрирована в основной модуль.

Разработанное программное обеспечение полностью соответствует всем проектным решениям, принятым в лабораторной работе №7.

Реализованы все запланированные компоненты: управление данными, выбор математической модели, выполнение расчетов, визуализация результатов и формирование отчета. Архитектура приложения следует клиент-серверной модели с выполнением всех вычислений на стороне клиента, что обеспечивает высокую производительность и безопасность данных.

3. Анализ сложности и рефакторинг исходных текстов

Проведен детальный анализ исходного кода приложения по установленным критериям сложности. Программа состоит из 4 основных модулей с общим объемом 380 строк кода. В модулях реализовано 15 функций с суммарным количеством 42 переменных. Максимальная глубина вложенности структурных операторов составляет 3 уровня, что свидетельствует о приемлемой сложности алгоритмов. Отсутствует наследование классов и рекурсивные вызовы, что упрощает понимание логики программы. Наибольшую сложность имеет модуль `math-processor.js`, содержащий матричные вычисления для метода наименьших квадратов.

Проведен рефакторинг кода с целью улучшения читаемости и производительности. В модуле `data-manager.js` оптимизирована обработка CSV-файлов через добавление валидации строк и обработку некорректных данных. В `math-processor.js` улучшена обработка вырожденных случаев матричных вычислений и добавлена проверка входных параметров. В основном модуле `app.js` обеспечено четкое разделение ответственности между компонентами приложения. Архитектура системы организована таким образом, что каждый модуль выполняет строго определенные функции: управление данными, математические расчеты и визуализация результатов разделены на независимые компоненты. Проведенные оптимизации позволили обеспечить стабильную работу приложения при обработке различных наборов экспериментальных данных и соответствие установленным в техническом задании требованиям к производительности.

Сравнительный анализ временной сложности до и после рефакторинга показал улучшение производительности критических участков кода. Обработка больших наборов данных (до 1000 точек) теперь выполняется в пределах 1-2 секунд, что соответствует установленным в ТЗ требованиям. Оптимизация алгоритмов визуализации позволила снизить нагрузку на

браузер при работе с графиками. Пространственная сложность осталась на прежнем уровне благодаря эффективному использованию структур данных.

Вывод: в ходе выполнения лабораторной работы №8 было успешно разработано специальное программное обеспечение «Программа для аппроксимации функции». Реализованы все запланированные компоненты, соответствующие техническому заданию и проектным решениям предыдущих этапов. Проведен сравнительный анализ инструментальных средств разработки, в результате которого обоснован выбор JavaScript с HTML/CSS как оптимальной технологии для данного проекта.

В процессе разработки выявлены и устранены логические ошибки обработки CSV-файлов, расчета экспоненциальной функции при отрицательных значениях Y и обработки одинаковых значений X , вызывавших зависание программы. Разработан интуитивно понятный пользовательский интерфейс, обеспечивающий ввод данных, выбор математической модели, визуализацию результатов и формирование отчетов.

Проведенный анализ сложности исходного кода показал соответствие программных компонентов установленным критериям качества. Выполненные процедуры рефакторинга позволили оптимизировать производительность приложения и улучшить читаемость кода. Программа успешно прошла тестирование и демонстрирует стабильную работу при обработке различных наборов экспериментальных данных.

Разработанное программное обеспечение полностью соответствует всем требованиям технического задания и готово к практическому использованию для решения задач математической аппроксимации в научно-исследовательской и учебной деятельности.