

**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 № 3

по дисциплине «Программная инженерия»

Выполнила: ст. гр. МО 32/2

Переузник В. С.

Проверил: доцент каф. ИТ

Полетайкин А. Н.

Краснодар

2025

Тема: техническое задание на создание программного продукта.

Индивидуальная тема: «Программа для аппроксимации функции».

Цель: освоение методик предварительного анализа разрабатываемой программы, формулирования функциональных и нефункциональных требований к программной реализации отдельных задач и к программе в целом, выработка навыков разработки технического задания.

Задание: 1. Установить назначение и общую цель создания программы. Определить структуру программы и состав функциональных задач; 2. Разработать функциональные требования к программе и построить модель требований в нотации UML; 3. Разработать специальные требования к информационному и математическому обеспечению программной реализации задач; 4. Разработать требования к инструментальному ПО, необходимому для разработки и эксплуатации ПП; 5. Установить необходимые нефункциональные требования к ПП.

Ход выполнения работы:

1. Назначение программы и цели ее создания

Назначение: программный продукт предназначен для автоматизации процесса математической аппроксимации экспериментальных данных. Вид автоматизируемой деятельности – расчет и анализ. ПП может быть использован на объектах информатизации, связанных с научными исследованиями, инженерными расчетами, обработкой данных экспериментов и учебным процессом (на кафедрах вузов, в исследовательских лабораториях, проектных отделах).

Цели: ПП нужен для устранения недостатков ручного метода расчета и ограничений универсальных программных продуктов (выявленных в лабораторной работе №2), а именно: низкой скорости, высокой вероятности ошибок, ограниченности анализа альтернативных моделей.

За счет использования ПП на объекте информатизации улучшается:

- 1) Производительность труда исследователя (время на проведение полного цикла аппроксимации - от ввода данных до получения отчета – значительно сокращается до секунд);
- 2) Точность и достоверность результатов (исключаются арифметические ошибки, присущие ручным расчетам);
- 3) Качество анализа (появляется возможность быстрого перебора и сравнения множества математических моделей для выбора оптимальной);
- 4) Стандартизация и воспроизводимость результатов (автоматическое формирование отчета обеспечивает единообразие документирования).

Критерии оценки достижения целей: сокращение времени на выполнение аппроксимации более чем на 90% по сравнению с ручным расчетом; снижение количества ошибок в расчетах до 0%; возможность тестирования большего количества различных типов функций за один сеанс работы.

На основе декомпозиции, выполненной в лабораторной работе №1, установлен следующий состав функциональных задач, подлежащих программной реализации:

1. Управление данными: ввод, импорт и валидация исходных экспериментальных данных.
2. Выбор модели аппроксимации: задание типа аппроксимирующей функции и ее параметров.
3. Выполнение расчетов: вычисление коэффициентов модели, расчет значений аппроксимирующей функции и оценка точности аппроксимации.
4. Визуализация результатов: построение графика, отображающего исходные данные и подобранную функцию.
5. Экспорт результатов: сохранение и вывод итогов работы (коэффициентов, формулы, графика, метрик точности) в виде отчета.

2. Функциональные требования и модель требований UML

Для формализации функциональных требований к программному продукту «Аппроксимация функции» была разработана статическая объектная модель в виде диаграммы вариантов использования. Диаграмма построена дедуктивным методом: в качестве базовых вариантов использования взяты задачи, указанные в подразделе 1, которые затем были детализированы с использованием отношений включения (<<include>>) и расширения (<<extend>>), принятых в нотации UML. Диаграмма представлена на Рис. 2.

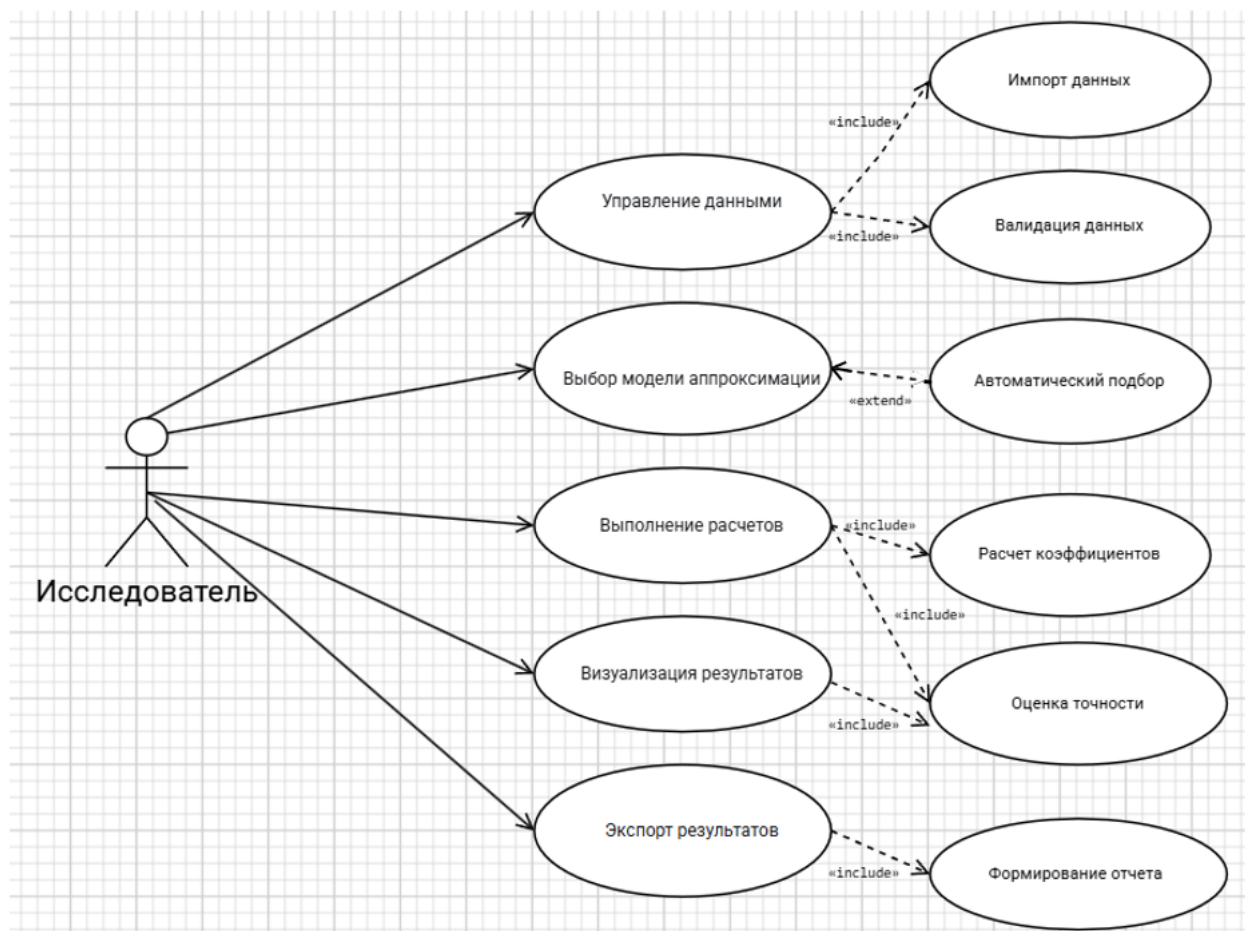


Рис. 2 — Диаграмма вариантов использования для ПП «Аппроксимация функции»

2.1. Требования к задаче «Управление данными»

Действия программы:

- Будет отображаться таблица для ввода пар координат (x, y).
- Будут обрабатываться CSV-файлы с разделителем-запятой.
- Будет проверяться корректность данных: числовой формат, отсутствие пустых значений

Требования к входным данным:

- *Наименование:* экспериментальные данные
- *Форма представления:* целые числа и числа с плавающей запятой
- *Источник:* прямой ввод в браузере или текстовый файл

Требования к выходным данным:

- Проверенные данные будут представлены в виде списка пар чисел.
- Сообщения об ошибках будут выводиться в виде текста на странице

Требования к программной реализации:

- Хранение данных: будет использоваться простой список в памяти браузера.
- Структура интерфейса: будет представлять обычную HTML таблицу и кнопку загрузки файла.
- Метод программирования: будут применяться простые функции на JavaScript.
- Именованное: data, points, checkData()

2.2. Требования к задаче «Выбор модели аппроксимации»

Действия программы:

- Будет отображаться выпадающий список с выбором: "Линейная", "Квадратичная", "Кубическая", "Экспоненциальная".
- По умолчанию будет выбрана "Линейная" (автоматический подбор).
- Степень полинома будет определяться автоматически по типу функции.

Требования к входным данным:

- Наименование: выбор пользователя.
- Форма представления: текст из выпадающего списка.
- Источник: Пользователь будет выбирать мышкой.

Требования к выходным данным:

- Выбранный тип функции.
- Степень полинома будет вычисляться автоматически.

Требования к программной реализации:

- Хранение данных: будут использоваться простые переменные.
- Структура интерфейса: будет представлять выпадающий список.
- Метод программирования: будет применяться обработка выбора из списка.
- Именованное: functionType

2.3. Требования к задаче «Выполнение расчетов»

Действия программы:

- По кнопке "Посчитать" будут находиться коэффициенты для формулы методом наименьших квадратов.
- Будет вычисляться точность по формуле и выводиться это значение.

- Будет показываться формула вида, например: $y = 2x + 1$.

Требования к входным данным:

- Наименование: данные и выбранная модель.
- Форма представления: список чисел и настройки.
- Источник: результаты предыдущих шагов.

Требования к выходным данным:

- Будут получены коэффициенты формулы.
- Будет вычислено значение точности.
- Будет сформирована готовая формула в текстовом виде.
- Время расчета: не будет превышать 1 секунду.

Требования к программной реализации:

- Хранение данных: будут использоваться переменные с результатами.
- Структура интерфейса: будет представлена кнопка "Посчитать" и блок для вывода формулы и точности.
- Метод программирования: будут применяться математические формулы на JavaScript.
- Именованное: coefficients, rSquared, calculate(), showFormula()

2.4. Требования к задаче «Визуализация результатов»

Действия программы:

- Будут рисоваться точки которые ввел пользователь.
- Будет рисоваться плавная линия полученной функции.
- Будут подписываться оси X и Y.

Требования к входным данным:

- Наименование: исходные точки и коэффициенты.
- Форма представления: списки чисел.
- Источник: результаты расчетов.

Требования к выходным данным:

- Будет создаваться картинка с графиком на странице.
- Качество: будет обеспечена четкая видимость точек и линии.

Требования к программной реализации:

- Хранение данных: будут использоваться данные для графика.
- Структура интерфейса: будет выделено место для графика на странице.

- Метод программирования: будет использоваться библиотека Chart.js.
- Именованное: chart, drawGraph().

2.5. Требования к задаче «Экспорт результатов»

Действия программы:

- Будут показываться все результаты на странице в красивом виде.
- Будет предоставлена возможность скопировать формулу.

Требования к входным данным:

- Наименование: все полученные результаты.
- Форма представления: текст и числа.
- Источник: результаты работы программы.

Требования к выходным данным:

- Будет формироваться красивый единый текст с формулой и точностью на странице.
- Достоверность: формула будет соответствовать введенным данным.

Требования к программной реализации:

- Хранение данных: все будет храниться в памяти браузера.
- Структура интерфейса: будет создан блок с результатами на странице.
- Метод программирования: будет применяться вывод текста в HTML.
- Именованное: showResults(), displayFormula().

Общие требования:

- Простота: все должно работать быстро и понятно.
- Время: расчеты не должны превышать 1-2 секунд.
- Надежность: проверка, что все данные правильные.

3. Специальные требования

Требования к информационному обеспечению:

- Требования к составу и структуре информации:
 - 1) Входные данные: будут представлять массив пар чисел (x, y) в формате JSON.
 - 2) Выходные данные: будет формироваться объект результатов в формате JSON, содержащий коэффициенты, формулу, значение точности.

3) Обязательные атрибуты: каждая точка данных должна содержать координаты x и y числового типа

- Требования к структурированию данных:

- 1) Будут использоваться стандартные структуры JavaScript (массивы, объекты).

- 2) Данные будут храниться только в оперативной памяти браузера.

- Требования к форматам данных:

- 1) Формат обмена данными: JSON.

- 2) Формат файлов импорта: CSV с разделителем-запятой.

- 3) Графические данные: обычное изображение графика на веб-странице.

- Требования к информационной безопасности:

- 1) Не будет осуществляться передача данных на сервер (все расчеты будут выполняться локально, в браузере, данные никуда не будут отправляться).

- 2) Будет проводиться валидация входных данных на корректность, в том числе для предотвращения инъекций кода.

Требования к математическому обеспечению

- Требования к математическим методам:

- Основной метод: будет применяться Метод Наименьших Квадратов (МНК) для решения систем линейных уравнений.

- Типы аппроксимирующих функций:

- Линейная: $y = ax + b$

- Квадратичная: $y = ax^2 + bx + c$

- Кубическая: $y = ax^3 + bx^2 + cx + d$

- Экспоненциальная: $y = ae^{bx}$ (расчет будет проводиться через логарифмы $\ln(y) = \ln(a) + bx$)

- Требования к алгоритмам:

- 1) Решение систем линейных уравнений методом матричных операций

- 2) Расчет коэффициента детерминации

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

8

где y_i – исходное значение, \hat{y}_i – расчетные значения, \bar{y}_i – среднее значение y .

3) Будет автоматически определяться степень полинома по выбранному типу функции.

- Ограничения на применение:

1) Минимальное количество точек: будет составлять на 1 больше количества коэффициентов.

2) Максимальная степень полинома: будет составлять 3 (кубическая функция).

3) Для экспоненциальной функции: все значения y должны быть > 0 .

- Требования к подготовке данных:

1) Будет проводиться проверка некорректных значений (NaN, Infinity, текст).

2) Будет проверяться достаточность данных для выбранного типа функции

- Требования к инструментальному ПО:

1) Будет использоваться библиотека Math.js для матричных операций.

2) Будут применяться встроенные математические функции JavaScript.

4. Требования к инструментальному ПО

- Требования к операционной среде:

1) Операционные системы: будут поддерживаться Windows 10, 11, macOS 10.14+, современные дистрибутивы Linux.

2) Браузеры: будут поддерживаться Google Chrome 90+, Mozilla Firefox 85+, Safari 14+, Microsoft Edge 90+.

3) Дополнительно: будет требоваться включенная поддержка JavaScript.

- Требования к средствам программной инженерии:

1) CASE-средства: Aspose Diagram (для построения UML-диаграмм).

2) Средства моделирования: использование нотации UML 2.x для диаграмм вариантов использования.

- Требования к инструментальным средствам разработки:

1) Язык программирования: JavaScript.

2) Веб-технологии: HTML5, CSS3.

3) Среда разработки: Visual Studio Code.

4) Система контроля версий: Git.

- Требования к использованию готовых программных пакетов:

- 1) Библиотека для графиков: Chart.js.

- 2) Библиотека для математических вычислений: Math.js.

- Требования к вспомогательным программным средствам:

- 1) Для запуска проекта: будет использоваться встроенный веб-сервер в VS Code (расширение Live Server).

- 2) Для отладки: инструменты разработчика в браузере (F12)

5. Нефункциональные требования

Требования к качеству ПП:

1. Работоспособность - система будет корректно выполнять аппроксимацию при наличии достаточного количества валидных данных.
2. Производительность - время расчета для 1000 точек данных не будет превышать 2 секунд.
3. Надежность - при некорректных входных данных система будет выводить сообщение об ошибке, а не завершать работу аварийно.
4. Удобство использования - интерфейс будет интуитивно понятен пользователю с базовой математической подготовкой.
5. Сопровождаемость - код будет читаемым и содержать комментарии для основных алгоритмов.
6. Безопасность - все вычисления будут выполняться локально, передача данных на сервер будет отсутствовать.

Требования к аппаратному обеспечению:

- Процессор: будет требоваться современный процессор с поддержкой JavaScript.
- Память: будет требоваться не менее 512 МБ оперативной памяти.
- Дисковое пространство: будет требоваться не менее 10 МБ для хранения веб-страницы и ресурсов.
- Монитор: будет требоваться разрешение не менее 1024×768 пикселей

Требования к пользовательскому интерфейсу:

- Наглядность - график и результаты расчетов будут четко видимы и читаемы.
- Простота освоения - время обучения основным функциям не будет превышать 10 минут.

- Отзывчивость - интерфейс будет реагировать на действия пользователя без заметных задержек

Вывод: в ходе выполнения лабораторной работы №3 было разработано техническое задание на создание программного продукта «Аппроксимация функции». В результате работы:

Были определены цели и задачи программы - автоматизация процесса математической аппроксимации экспериментальных данных для повышения точности и скорости расчетов.

Разработана диаграмма вариантов использования, которая наглядно показывает взаимодействие пользователя с системой через пять основных функций: управление данными, выбор типа функции, выполнение расчетов, визуализация результатов и экспорт.

Сформулированы технические требования к реализации программы веб-приложения на JavaScript с использованием библиотек Chart.js для графиков и Math.js для математических вычислений.

Определены специальные требования к математическому обеспечению, включая метод наименьших квадратов и основные типы аппроксимирующих функций.

Установлены критерии качества работы программы - производительность, надежность, простота использования и безопасность данных.

Полученное техническое задание полностью готово для перехода к следующему этапу и содержит все необходимые требования для успешной реализации проекта.