

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 № 5**

**по дисциплине «Программная инженерия»**

Выполнила: ст. гр. МО 32/2

Переузник В. С.

Проверил: доцент каф. ИТ

Полетайкин А. Н.

Краснодар

2025

Тема: Разработка концептуальной структуры программного продукта:  
объектно-ориентированный подход.

Индивидуальная тема: «Программа для аппроксимации функции».

Цель: освоение методики объектно-ориентированного анализа, приобретение навыков формирования сценариев работы ПП и построения диаграммы классов ПП на языке UML.

Задание: 1) Используя материалы отчета о выполнении лабораторной работы № 4 и методику объектно-ориентированного анализа, выполнить полное перечисление потока событий каждой автоматизированной функции и составить пошаговое описание ее реализации посредством ПП; 2) Для каждой автоматизированной функции перечислить и описать все возможные альтернативные варианты реализации функции, включая некорректные ситуации и аномалии, которые могут иметь место при реализации функции и оказывать существенное влияние на работу ПП в целом; 3) Скорректировать поток событий при нормальной реализации обследуемой функции так, чтобы последствия перечисленных аномалий оказались минимальными; 4) Проанализировать описанные потоки событий на предмет выявления набора абстракций предметной области проектируемого ПП. Для этого провести синтаксический анализ текстов потоков событий. В качестве предварительных кандидатов в абстракции принять подлежащие, выделенные из потоков событий; 5) Разделить выделенные абстракции на три типа: сущности, поведения, интерфейсы. Для каждой абстракции указать ее вид согласно принятой классификации. Результат представить в виде таблицы; 6) Проанализировать поведение выделенных абстракций. Абстракции, для которых уместно выделить возможное поведение в пределах функциональности ПП, представленной диаграммой требований UML, занести данные в таблицу; 7) Опираясь на данные последней таблицы построить диаграмму классов UML (class diagram), указывая при этом лишь имена классов без указания свойств класса. 8) Выявить в потоках событий атрибуты классов и внести их в секции классов на диаграмме. В качестве атрибутов выбирать подлежащие, выявленные в потоках событий и не вошедшие в первую таблицу, а также абстракции из первой таблицы, не вошедшие во вторую; 9) Выявить в потоках событий операции классов и внести их в секции классов на диаграмме. В качестве операций выбирать сказуемые, выделенные из текстов анализируемых потоков событий. 10) Установить между классами адекватные отношения и обозначить их кратность. При необходимости пометить отношения текстовыми метками.

Ход выполнения работы:

***1. Основные потоки событий для каждой автоматизированной функции ПП.***

**1.1. Функция "Управление данными"**

- Основной поток событий:

1. Ввод пользователем экспериментальных данных через табличный интерфейс.
2. Проверка системой корректности формата введенных данных.
3. Выбор пользователем типа аппроксимирующей функции из выпадающего списка.
4. Сохранение системой проверенных данных и выбранного типа функции.

- Альтернативные потоки:

A1: Импорт данных из CSV-файла.

1. Выбор пользователем файла для импорта.
2. Загрузка системой данных из CSV-файла.
3. Проверка системой соответствия формата данных.
4. Отображение системой загруженных данных в интерфейсе.

- Потоки ошибок:

O1: Ошибка формата данных

1. Обнаружение системой нечисловых значений.
2. Выделение системой некорректных ячеек.
3. Блокировка системой возможности выполнения расчетов.

- Корректировка основного потока:

Добавление этапа предварительной проверки данных перед сохранением.

## 1.2. Функция "Выбор модели аппроксимации"

- Основной поток событий:

1. Получение системой выбранного типа функции.
2. Если тип функции пользователем не выбран – установление системой типа "Линейная" по умолчанию.
3. Определение системой параметров модели согласно выбранному типу.
4. Формирование системой параметров модели для передачи в модуль расчетов.

- Альтернативные потоки:

A1: Изменение типа функции

1. Изменение пользователем выбора типа функции в выпадающем списке.
2. Обновление системой параметров модели согласно новому выбору.

- Потоки ошибок:

O1: Несовместимость типа функции с данными.

1. Обнаружение системой противоречий между данными и выбранной моделью.
2. Вывод системой сообщения об ошибке.
3. Блокировка системой возможности выполнения расчетов.

## 1.3. Функция "Выполнение расчетов"

- Основной поток событий:

1. Получение системой проверенных данных и параметров модели.
2. Формирование системы уравнений методом наименьших квадратов.
3. Вычисление коэффициентов аппроксимирующей функции.
4. Вычисление метрики точности ( $R^2$ ).

## 5. Формирование системой аналитического представления функции.

- Альтернативные потоки:

A1: Расчет дополнительных метрик точности

1. Вычисление системой среднеквадратичной ошибки.

- Потоки ошибок:

O1: Невозможность решения системы уравнений

1. Обнаружение системой вырожденной матрицы.
2. Предложение системой изменить тип аппроксимирующей функции – вывод системой сообщения об ошибке и блокировка дальнейших действий.

### 1.4. Функция "Визуализация результатов"

- Основной поток событий:
  1. Получение системой коэффициентов функции и исходных данных.
  2. Создание графика с координатной сеткой.
  3. Отображение исходных точек на графике.
  4. Построение кривой аппроксимирующей функции.
  5. Добавление подписей осей и легенды.
- Альтернативные потоки и потоки ошибок отсутствуют.

### 1.5. Функция "Экспорт результатов"

- Основной поток событий:
  1. Сбор системой всех результатов расчетов.
  2. Формирование структурой структурированного отчета.

### 3. Просмотр пользователем итогового отчета.

- Альтернативные потоки и потоки ошибок отсутствуют.
- Модификации диаграмм:
  1. Диаграмма требований UML - уточнены названия вариантов использования.
- Диаграмма декомпозиции A0 - уточнены названия интерфейсных дуг.

## 2. Объектно-ориентированный анализ.

На основе анализа потоков событий выделены следующие абстракции:

№	Абстракция	Тип	Вид	Описание
1	Пользователь	Сущность	Роли	Исследователь, работающий с системой
2	Экспериментальные данные	Сущность	Предметы	Набор точек (X,Y) для аппроксимации
3	Аппроксимирующая функция	Сущность	Концепции	Математическая модель для приближения данных
4	JavaScript-движок	Интерфейс	Устройства	Среда выполнения клиентской части приложения
5	Math.js	Поведение	Инструменты	Библиотека для математических вычислений
6	Chart.js	Поведение	Инструменты	Библиотека для визуализации данных
7	Коэффициенты	Сущность	Показатели	Параметры аппроксимирующей функции
8	Метрики точности	Сущность	Показатели	Количественные оценки качества аппроксимации
9	Отчет	Сущность	Концепции	Структурированное представление результатов

Таблица 2.1 - Абстракции подсистемы

№	Абстракция	Требование	Описание поведения
1	Экспериментальные Данные	Управление данными	Хранение, валидация и импорт исходных данных
2	Математическая Модель	Выбор модели аппроксимации	Определение типа функции и ее параметров
3	Аппроксимация	Выполнение расчетов	Расчет коэффициентов и оценка точности модели
4	График	Визуализация результатов	Построение и отображение графиков данных
5	Отчет	Экспорт результатов	Формирование и экспорт итоговых результатов

Таблица 2.2 - Абстракции подсистемы и их поведение

На основе данных таблиц 2.1 и 2.2 построена диаграмма классов UML (Рис. 2.3). Процесс построения включал следующие этапы:

1) Построение каркаса диаграммы с указанием имен классов:

- Выделены 5 основных классов, соответствующих абстракциям поведения из таблицы 2.2
- Определена предварительная структура отношений между классами

2) Выявление атрибутов классов:

- Атрибуты выделены из потоков событий как подлежащие, не вошедшие в таблицу 2.1
- Учтены абстракции из таблицы 2.1, не вошедшие в таблицу 2.2

3) Выявление операций классов:

- Операции выделены как сказуемые (отглагольные существительные) из текстов потоков событий
- Каждая операция соответствует конкретному действию в основном потоке событий

#### 4) Установление отношений и их кратности:

- Определены адекватные отношения между классами на основе логики работы системы
- Установлена кратность отношений в соответствии с бизнес-логикой

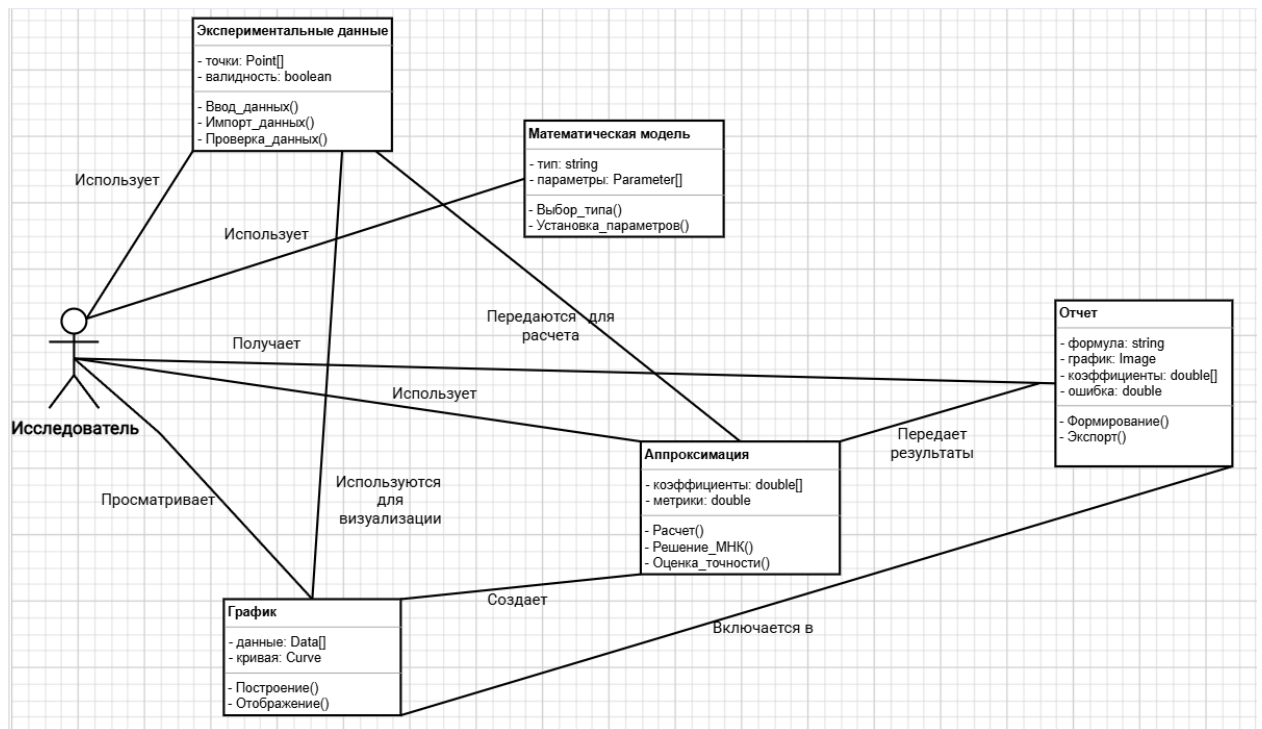


Рисунок 2.3 - Диаграмма классов UML

#### Обоснование выбора атрибутов и операций:

##### 1) Атрибуты классов выделены на основе анализа потоков событий:

Для класса "Экспериментальные данные":

- "точки" - из основного потока функции "Управление данными" (п.1.1)
- "валидность" - из проверки корректности формата данных (п.1.1)

Для класса "Математическая модель":

- "тип" - из основного потока функции "Выбор модели аппроксимации" (п.1.2)
- "параметры" - из определения параметров модели (п.1.2)



Для класса "Аппроксимация":

- "коэффициенты" - из основного потока функции "Выполнение расчетов" (п.1.3)
- "метрики" - из вычисления метрики точности (п.1.3)

Для класса "График":

- "данные" - из использования исходных данных для визуализации (п.1.4)
- "кривая" - из построения кривой аппроксимирующей функции (п.1.4)

Для класса "Отчет":

- "формула" - из формирования аналитического представления (п.1.3)
- "график" - из включения графического представления (п.1.5)
- "коэффициенты" - из передачи результатов расчетов (п.1.5)
- "ошибка" - из вычисления метрик точности (п.1.3)

2) Операции классов соответствуют существительным из потоков событий:

Для класса "Экспериментальные данные":

- "ввод данных" - из основного потока п.1.1
- "импорт данных" - из альтернативного потока A1
- "проверка данных" - из проверки формата данных

Для класса "Математическая модель":

- "выбор типа" - из выбора типа функции (п.1.2)
- "установка параметров" - из определения параметров модели

Для класса "Аппроксимация":

- "расчет" - из вычисления коэффициентов (п.1.3)
- "решение МНК" - из формирования системы уравнений
- "оценка точности" - из вычисления метрик точности

Для класса "График":

- "построение" - из создания графика (п.1.4)
- "отображение" - из отображения исходных точек

Для класса "Отчет":

- "формирование" - из сбора результатов (п.1.5)
- "экспорт" - из предоставления отчета пользователю

### *Обоснование отношений и их кратности*

Отношения между классами установлены на основе анализа взаимодействий в потоках событий:

1. Исследователь → Экспериментальные данные (1..\*). Один исследователь может работать с несколькими наборами данных
2. Экспериментальные данные → Аппроксимация (1). Один набор данных передается для расчета.
3. Математическая модель → Аппроксимация (1). Одна модель определяет параметры расчета
4. Аппроксимация → График (1). Создается один график на основе расчетов
5. Аппроксимация → Отчет (1). Результаты передаются в один отчет
6. График → Отчет (1). График включается в отчет

Диаграмма демонстрирует полное соответствие операций, выявленных в ходе объектно-ориентированного анализа, с функциональными требованиями к программному продукту.

**Вывод:** в ходе выполнения лабораторной работы была разработана концептуальная структура программного продукта "Программа для аппроксимации функции" на основе объектно-ориентированного подхода. Построены детальные потоки событий для всех автоматизированных

функций, выполнено выделение и классификация абстракций предметной области, построена полная диаграмма классов UML с обоснованием всех элементов. Все проектные решения согласованы с функциональной моделью из лабораторной работы 4 и обеспечивают надежную основу для последующей реализации системы.