**Отчет по выполнению лабораторной работы №4**

**"Создание диаграммы кооперации и диаграммы последовательности"**

Дисциплина: Технология разработки программного обеспечения

Преподаватель: Сибирев И.В.

Студент: Алиев Тимур Заурович

Группа: VДКИП 111-прог

Вариант: 1

Дата выполнения: 03 апреля 2025 г.

**Цель работы**

Ознакомиться с процессом создания диаграмм кооперации и последовательности в нотации UML для описания взаимодействия объектов в автоматизированной системе учета договоров страховой компании "АС Учет договоров". Разработать диаграммы для сценария добавления договора.

**Описание предметной области**

Предметная область — страховая компания с сетью филиалов, занимающаяся заключением договоров страхования. Основные сущности:

1. Филиал — точка оформления договоров.

2. Вид страхования — категория страхового продукта.

3. Договор — объект учета с финансовыми данными (страховая сумма, тариф, премия).

Система автоматизирует управление данными и расчеты.

**Ход выполнения работы**

**1. Выбор сценария:**

Рассмотрен сценарий "Добавление договора" как ключевая операция системы, включающая ввод данных, проверку и расчет премии.

**2. Определение объектов:**

- :Employee — сотрудник филиала.

- :Interface — пользовательский интерфейс.

- :ContractManager — управляющий объект.

- :FilialDB, :InsuranceTypeDB, :ContractDB — базы данных.

**3. Разработка диаграмм:**

- Диаграмма кооперации: показывает взаимодействие объектов с нумерацией сообщений.

- Диаграмма последовательности: демонстрирует порядок действий во времени.

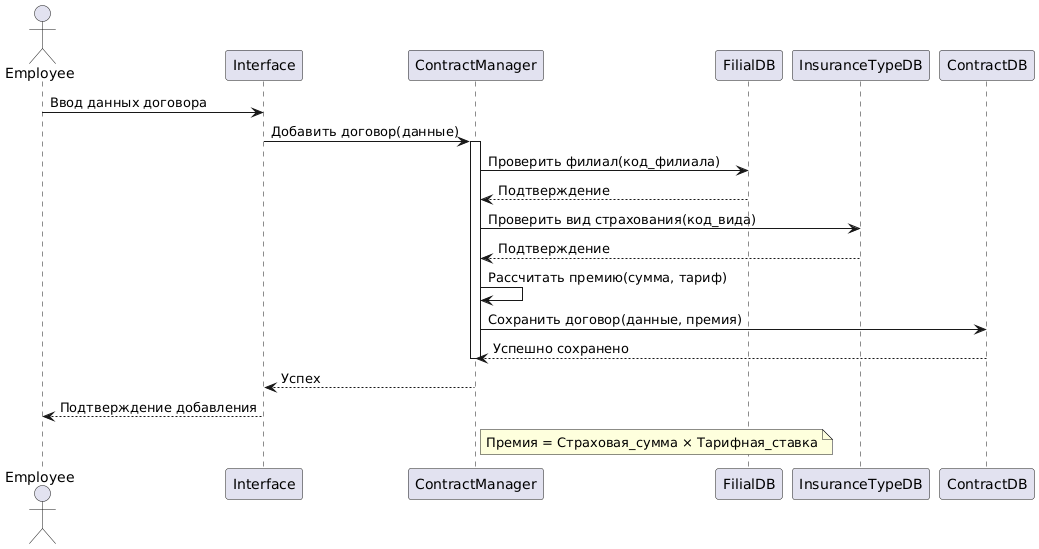
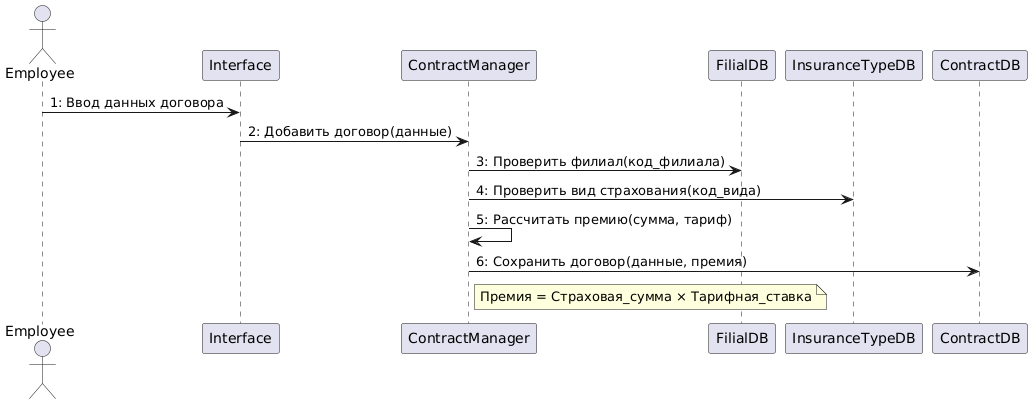
Диаграммы созданы в одном файле PlantUML для удобства.

**Результаты работы**

Разработаны две диаграммы для сценария "Добавление договора":

1. Диаграмма кооперации — отражает связи между объектами системы при добавлении договора.

2. Диаграмма последовательности — показывает последовательность вызовов от ввода данных до сохранения договора.



---

**Выводы**

В ходе работы освоены принципы построения диаграмм кооперации и последовательности. Разработанные диаграммы демонстрируют взаимодействие объектов системы, что важно для проектирования программного обеспечения. Полученные навыки применимы для дальнейшей реализации системы.