**Лабораторная работа №4**

**Сапицкий Алексей К4110с**

**Цель работы**

* изучить основы разработки объектных моделей с использованием шаблонов GRASP для распределения обязанностей между классами.

**Концепции**

Представленные далее диаграммы, дополняют описание функциональной модели описывающей платформу умного дома. В данной работе было рассмотрено серверно-клиентской взаимодействие и клиентская реализация процесса проверки принятия пользовательских соглашений. Который необходим в рамках требований ТЗ касающихся юридической и патентной составляющих.

**Диаграмма последовательностей**

На данной диаграмме взаимодействие клиента и сервера представлено в последовательности шагов которые в итоге приводят к выполнению определенного пользовательского сценария. В данном случае можно подробно рассмотреть, как поэтапно должен реализовываться процесс. На данном уровне детализации уже возможно начинать разработку как на стороне клиента, так и на стороне сервера. В то же время данная диаграмма так же может служить ориентиром для тестировщиков, проверяющих соответсвие работы системы тех. заданию.

A screenshot of a cell phone

Description automatically generated

**Диаграмма классов**

На данной диаграмме предствалены отношения между классами в модуле соглашений, который отвечает за отправку/принятие соглашений, а также их сохранение. Данная диаграмма обычно созадется разработчиком в процессе декомпозиции задачи и является схемтаическим отображением реальных взаимодействий между классами в коде.

A close up of a piece of paper

Description automatically generated

Вывод

В данной лабораторной работе были изучены методы для построения диаграмм последовательностей и диаграмм классов. Оба типа этих диаграмм являют собой довольно низкоуровневое представление рассматриваемого процесса и поэтому часто полезны для разработчиков и тестировщиков в реальных проектах. А диаграммы классов являются неотъемлемой частью разработки любой качественной архитектуры крупного приложения, так как позволяют заметно ускорить и упростить процесс написания кода разработчиком. И позволяют исключить значительные изъяны в архитектуре ПО.