МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**"Южно-Уральский государственный университет"**

**(национальный исследовательский университет)**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Объектно-ориентированные CASE-технологии»

ЮУрГУ – 02.04.021.2019.308/5722.КП

|  |  |
| --- | --- |
| Нормоконтролер,доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_О.Н. Иванова  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | Научный руководитель:  к.п.н.  \_\_\_\_\_\_\_\_\_\_О.Н. Иванова  Автор работы:  студент группы КЭ-120  \_\_\_\_\_\_\_\_\_\_А.В. Юферов  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

Челябинск-2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**"Южно-Уральский государственный университет"**

**(национальный исследовательский университет)**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_».\_\_\_\_\_\_\_\_\_\_\_\_.2019

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине «Объектно-ориентированные CASE-технологии»

студенту группы КЭ-120

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Юферову Александру Владимировичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество студента)

обучающемуся по направлению   
02.04.02 «Фундаментальная информатика и информационные технологии»

1. **Тема работы**Разработка UML-диаграмма для программы управления демодулятором
2. **Срок сдачи студентом законченной работы:** 28.05.2019 г.
3. **Исходные данные к работе**
4. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования – М.: Символ-Плюс, 2006. – 192 с.
5. [Арлоу Д., Нейштадт А. UML 2 и Унифицированный процесс](http://edu.susu.ru/main/mod/resource/view.php?id=25479). Практический объектно-ориентированный анализ и проектирование, 2-е изд. – Пер. с англ. – СПб.: Символ-Плюс, 2007. – 624 с., ил.
6. **Перечень подлежащих разработке вопросов**
7. Постановка задачи
8. Диаграмма классов / схема базы данных / обоснование отсутствия возможности проектирования данных диаграмм (для проектов, создаваемых на платформах, предполагающих функциональное или логическое программирование, например, при программировании микроконтроллеров)
9. Диаграмма внутренней структуры / диаграмма компонентов / диаграмма размещения
10. Диаграмма вариантов использования
11. Диаграмма взаимодействия
12. Диаграмма состояний
13. Диаграмма деятельности
14. **Дата выдачи задания:** «\_\_\_»\_\_\_\_\_\_\_\_\_2019 г.

Научный руководитель О.Н. Иванова

Задание принял к исполнению А.В. Юферов

Оглавление

[1. Постановка задачи 5](#_Toc9879821)

[2. Диаграмма классов 6](#_Toc9879822)

[3. Диаграммы компонентов и размещения 9](#_Toc9879823)

[4. Диаграмма вариантов использования 11](#_Toc9879824)

[5. Диаграмма взаимодействия 12](#_Toc9879825)

[6. Диаграмма состояний 13](#_Toc9879826)

[7. Диаграмма деятельности 14](#_Toc9879827)

[Заключение 15](#_Toc9879828)

# 1. Постановка задачи

Необходимо разработать UML-диаграммы для программы управления демодулятором.

Программа предназначена для наладчиков телекоммуникационного оборудования. Основные задачи решаемые с помощью программы: настройки частоты и контроль уровня сигналов. Связь устройства и компьютера осуществляется по технологии COM over USB. Для программы устройство будет доступно как COM-порт. Программа должна обладать графическим интерфейсом пользователя. Недопустимы ситуации, при которых интерфейс программы не отвечает на запросы пользователя. Разработка программы должна осуществляться на языке Java.

Основные параметры устройства:

* *частота несущей* (далее частота) - измеряется в МГц; представляется дробным числом; доступно на чтение и запись;
* *уровень входного сигнала* (далее уровень сигнала) – измеряется в дБмкВ; представляется целым числом; измеряемый параметр (доступен только для чтения); измерение происходит один раз в секунду.

# 2. Диаграмма классов

Для удобства чтения диаграмма классов всей системы разбита на две.

На рисунке 1 представлена основная диаграмма классов.

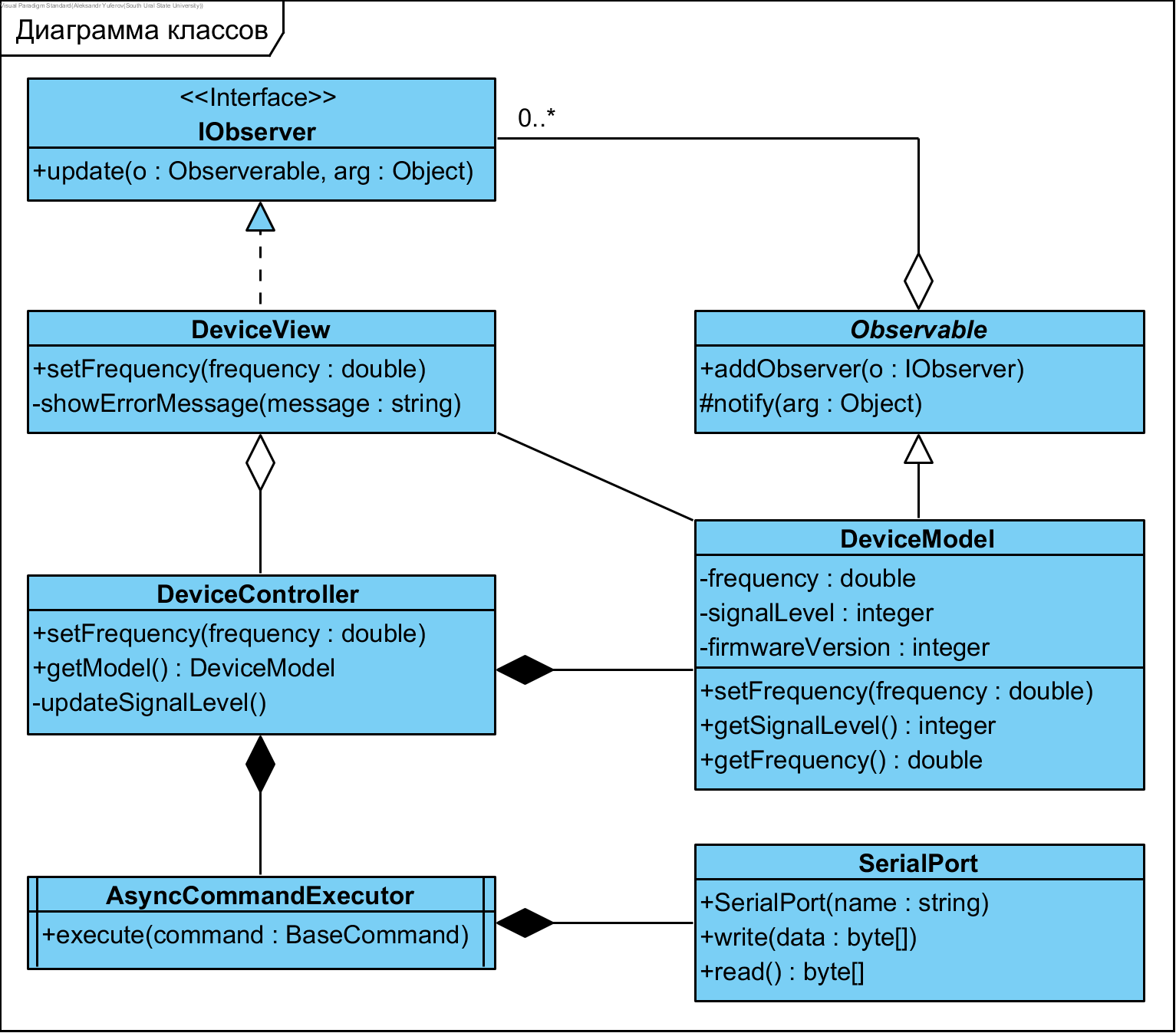


Рис. 1. Основная диаграмма классов

На диаграмме можно увидеть шаблон проектирования «модель-представление-контроллер» (англ. Model-View-Controller, MVC). В разрабатываемой системе он представлен классами DeviceView, DeviceModel и DeviceController. Класс DeviceView – представление, отвечающее за отображение данных устройства пользователю. Представление берет данные из модели – DeviceModel. При необходимости обработки пользовательских действий представление обращается к контроллеру – DeviceController, который меняет модель и осуществляет дополнительные действия. Так, например, в разрабатываемой системе контроллер отправляет новые настройки частоты на устройство с помощью класса AsyncCommandExecutor. Представление узнает об изменении модели с помощью шаблона «наблюдатель» (англ. observer).

На рисунке 2 представлена диаграмма классов, описывающая классы ответственные за общение с устройством.

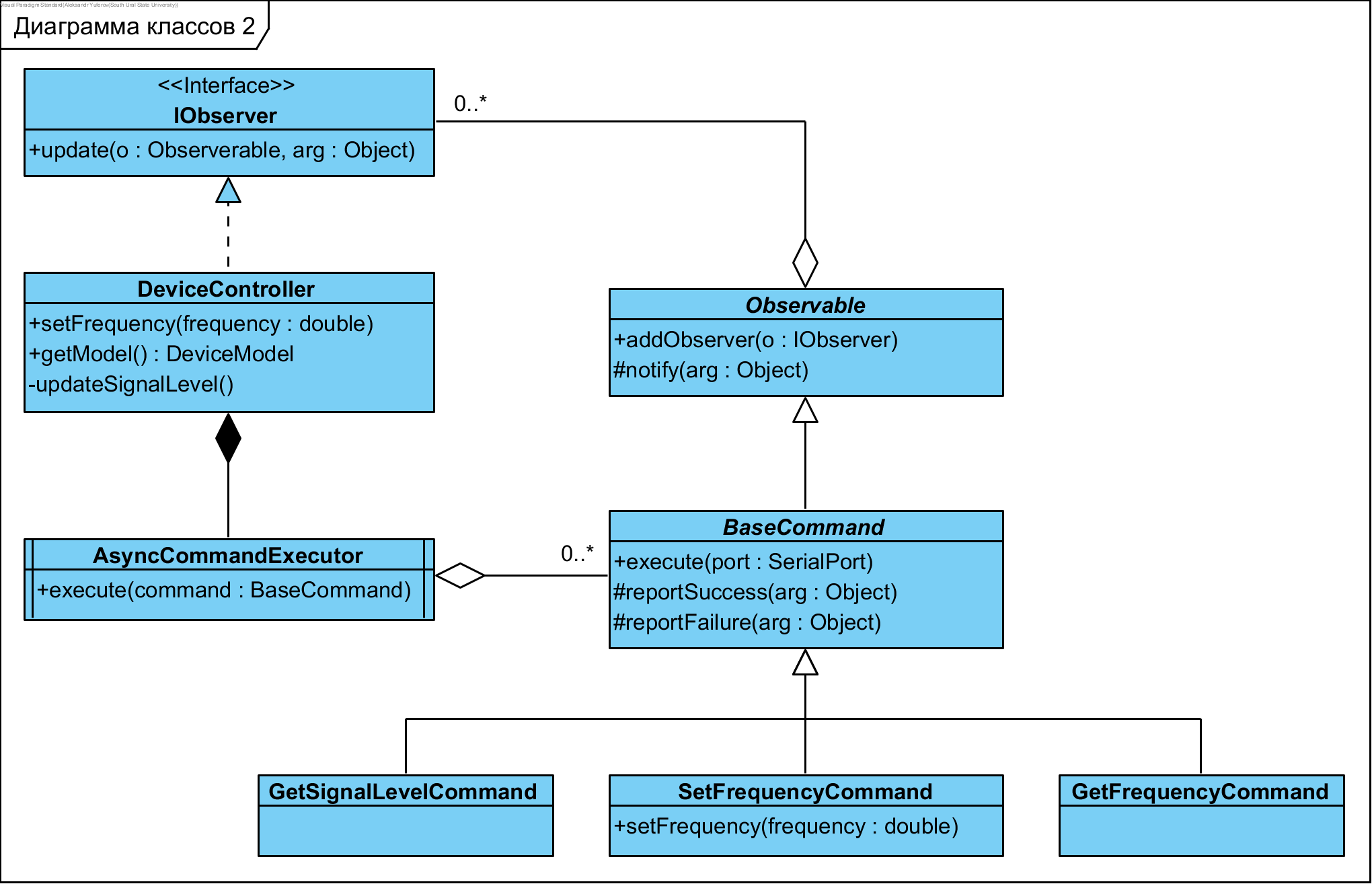


Рис. 2. Диаграмма классов, описывающая классы ответственные за общение с устройством

Логика общения устройства инкапсулирована с помощью шаблона проектирования «команда» (англ. command) в наследниках абстрактного класса BaseCommand: GetFrequencyCommand (команда запроса частоты), SetFrequencyCommand (команда установки нового значения частоты) и GetSignalLevelCommand (команда получения текущего уровня сигнала). Контроллер DeviceController с помощью перечисленных выше команд осуществляет общение с устройством посредством активного класса AsyncCommandExecutor, который исполняет команды в порядке поступления. Класс AsyncCommandExecutor – активный, т.е. исполняет свою логику в отдельном потоке. Это связано с тем фактом, что операции ввода и вывода длительные и могут привести к зависанию пользовательского интерфейса.

# 3. Диаграммы компонентов и размещения

На рисунке 3 представлена диаграмма размещения компонентов системы.

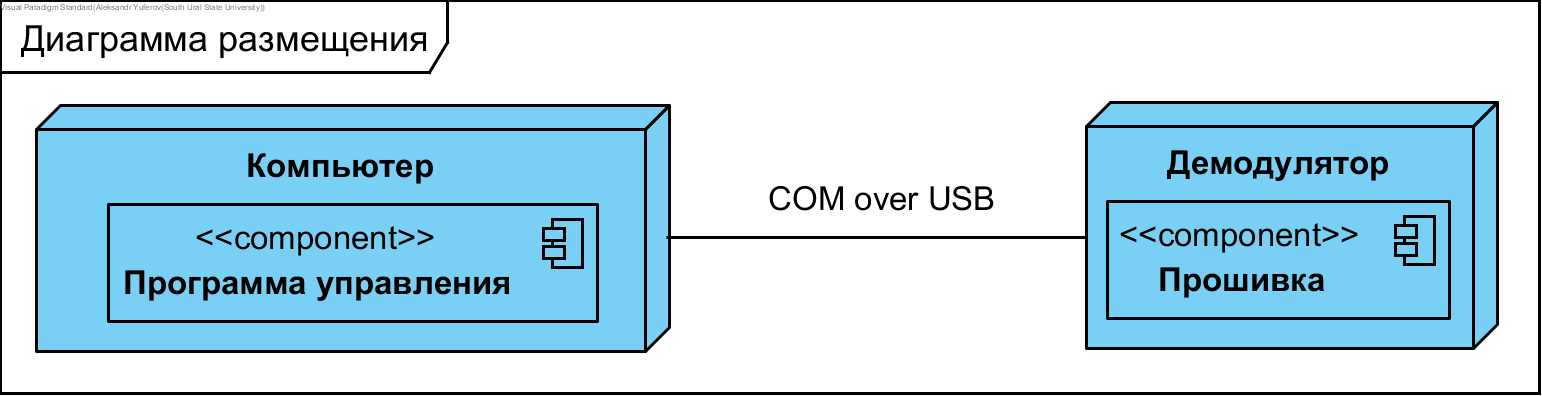


Рис. 3. Диаграмма размещения компонентов системы

С помощью диаграммы видно, что компьютер и устройство соединены по технологии «COM over USB». На компьютере пользователя размещается компонент «Программа управления», который и описан в данной работе, а на устройстве располагается компонент «Прошивка» осуществляющий управление устройством (данный компонент в работе не рассматривается).

На рисунке 4 представлена диаграмма компонентов программы «Программа управления».

Система разбита на четыре компонента:

* GUI – компонент графического пользовательского интерфейса;
* DomainModel – доменная модель программы;
* CommunicationModule – компонент ответственный за прием и передачу сообщений устройству;
* JavaCore – библиотека языка Java.

Компонент GUI зависит от компонента DomainModel. Компонент DomainModel зависит от компонента CommunicationModule. Компоненты GUI, DomainModel и CommunicationModule зависят от компонента JavaCore.

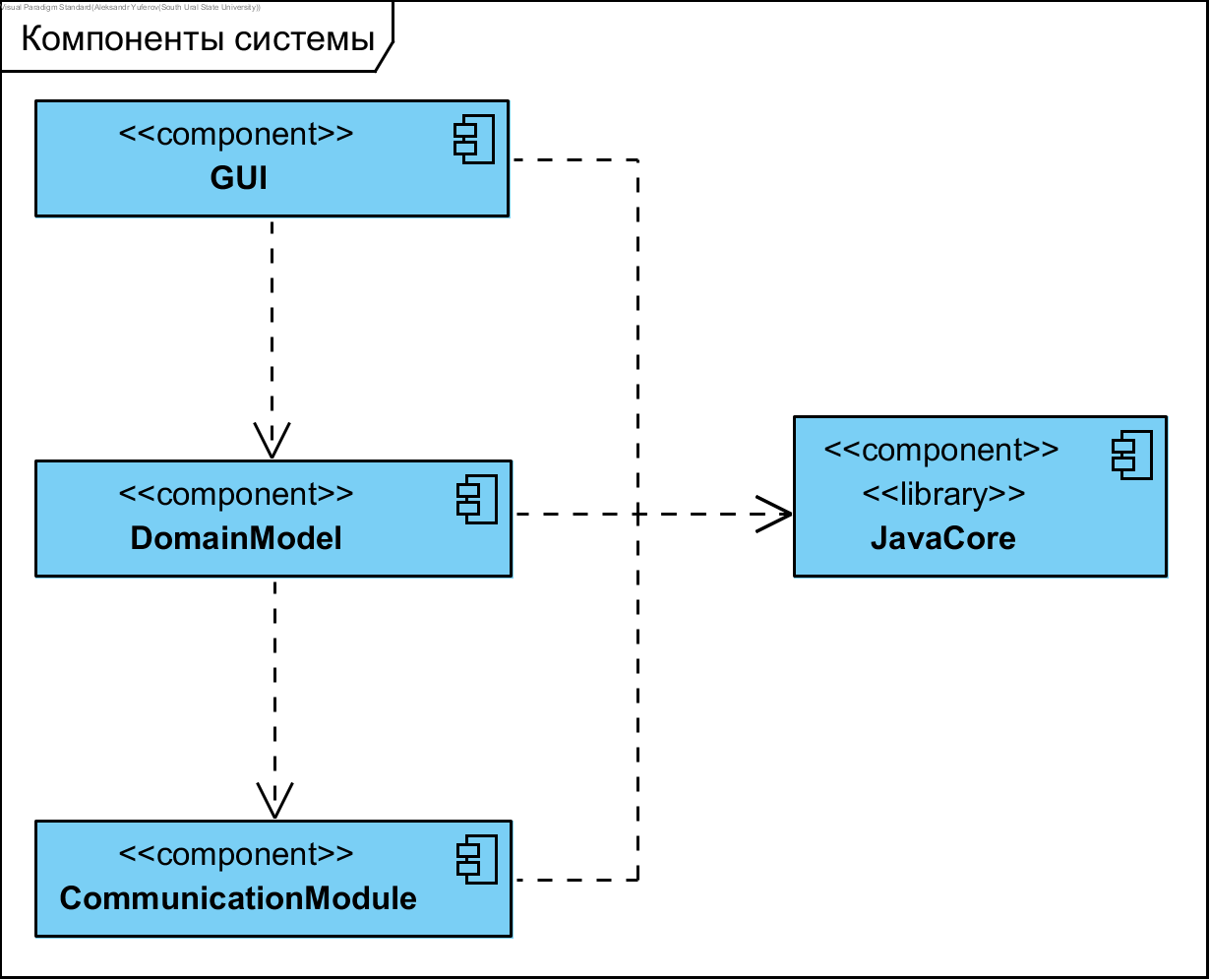


Рис. . Диаграмма размещения компонентов системы

# 4. Диаграмма вариантов использования

На рисунке 5 представлена диаграмма вариантов использования программы.

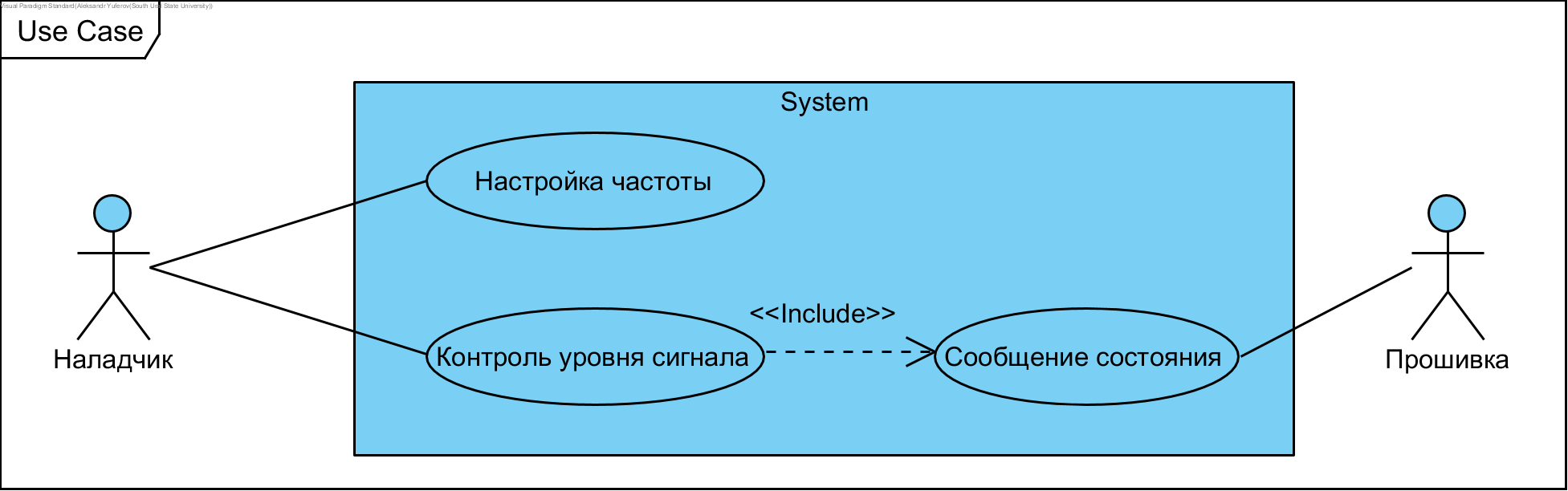


Рис. 5. Диаграмма вариантов использования программы

С системой взаимодействуют два актора: «Наладчик» и «Прошивка».

Наладчик может выполнять две задачи: настройку частоты и осуществлять контроль уровня сигнала.

Прошивка демодулятора сообщает о текущем состоянии устройства. Эти данные отображаются пользователю во время контрольного наблюдения за устройством.

# 5. Диаграмма взаимодействия

На рисунке 6 представлена диаграмма взаимодействия системы, описывающая процесс изменения частоты пользователем.

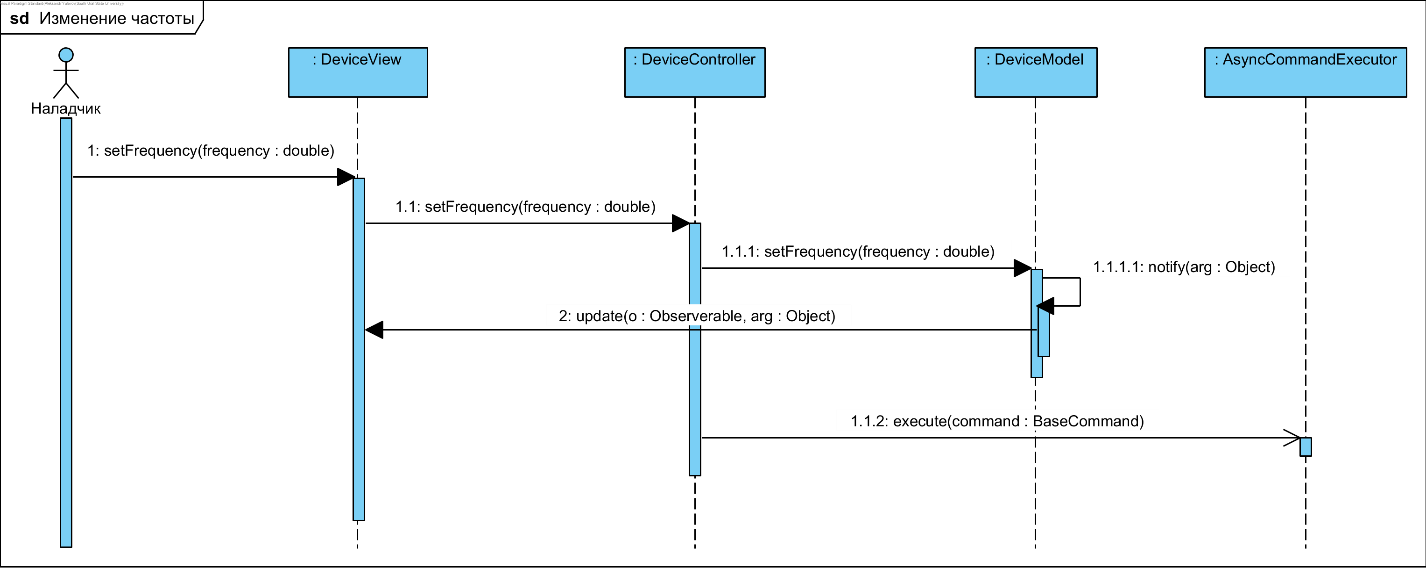


Рис. 6. Диаграмма взаимодействия системы, описывающая процесс изменения частоты пользователем

Пользователь меняет частоту, используя форму настроек на экране (объект класса DeviceView). Представление делегирует обязанности изменения частоты контроллеру DeviceController. Контроллер меняет данные в модели DeviceModel на новые. Модель оповещает своих «наблюдателей» об изменении, в числе которых находится и представление. После изменения модели, контроллер отправляет команду устройству на изменение частоты с помощью объекта класса AsyncCommandExecutor.

# 6. Диаграмма состояний

На рисунке 7 представлена диаграмма состояний объектов класса DeviceController.

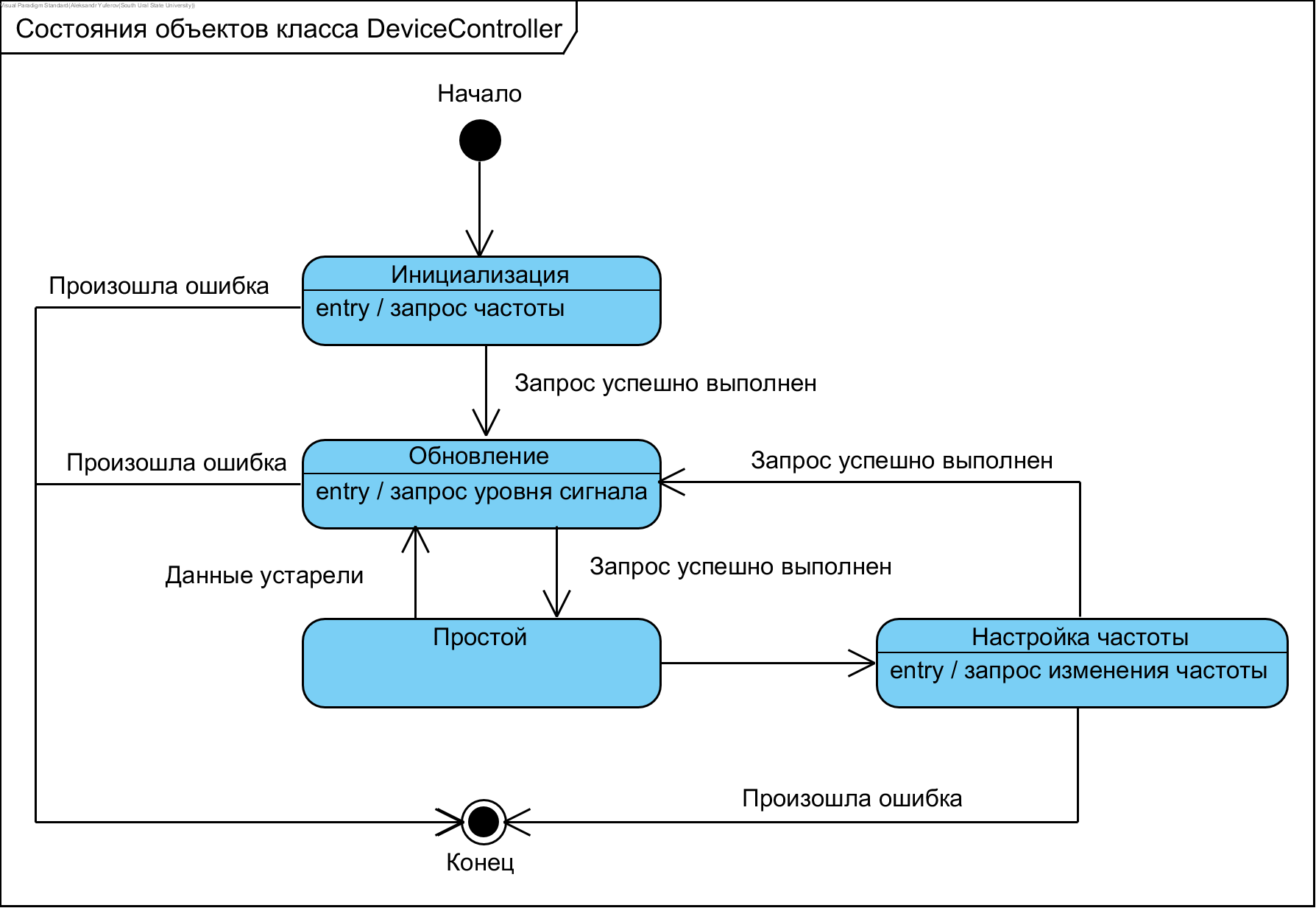


Рис. 7. Диаграмма состояний объектов класса DeviceController

Первое состояние – «Инициализация», попадая в которое объект запрашивает установленную частоту у устройства. Если данные о частоте были успешно получены, то дальше объект переходит в состояние «Обновление», при попадании в которое у устройства запрашиваются данные о текущем уровне сигнала. Если данные были успешно получены, то контроллер переходит в состояние «Простой», из которого он может выйти двумя путями – дождаться устаревания данных или изменения пользователем частоты.

# 7. Диаграмма деятельности

На рисунке 8 представлена диаграмма деятельности, описывающая выполнение команды.

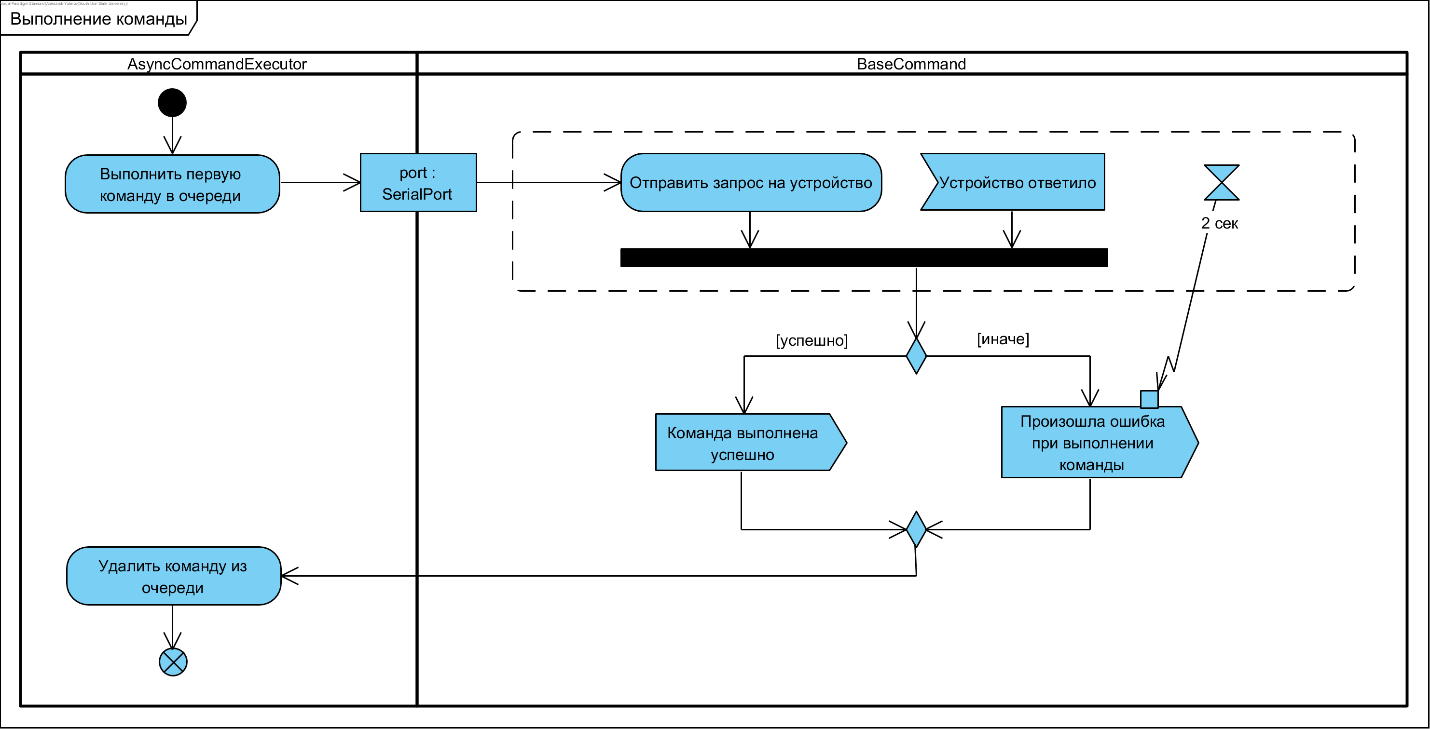


Рис. 8. Диаграмма деятельности, описывающая выполнение команды

В объектах класса AsyncCommandExecutor выстраивается очередь из команд, посылаемых устройству, для выполнения берется первая в очереди команда. Для отправки запроса на устройство используется объект класса SerialPort, который контролирует COM-порт. После отправки на устройства запроса начинается ожидание ответного сигнала. Если запрос выполнен успешно, посылается сигнал об успешном выполнении команды, иначе посылается сигнал об ошибке. Также если отправка и ожидание ответа длятся более 2 секунд, то запрос считается выполненным с ошибкой и посылается сигнал об ошибке во время выполнения запроса. Далее команда считается выполненной, и она удаляется из очереди. После этого поток выполнения команды считается оконченным.

# Заключение

В данной работе была спроектирована программа управления демодулятором с помощью UML-диаграмм. Были разработаны следующие типы диаграмм: диаграммы классов, диаграмма размещения, диаграмма вариантов использования, диаграмма последовательности, диаграмма состояний и диаграмма деятельности. Все поставленные задачи в ходе выполнения курсовой работы были выполнены.