Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование программного обеспечения»

Отчёт по лабораторной работе №7

Объектно-ориентированное моделирование. ДИАГРАММЫ ПОВЕДЕНИЯ UML.

Цель: Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования структуры информационной системы с применением UML.

Выполнил: Драч К.А.

ФИТ 3 курс 1 группа

Преподаватель: Панченко О.Л.

**Введение**

Целью данной работы является изучение методологии объектно-ориентированного моделирования с применением средств UML и применение этой методологии для создания двух ключевых диаграмм поведения. Рассмотренная в проекте тематика - авиакомпания - стала основой для разработки этих диаграмм.

Первая диаграмма поведения моделирует процесс обновления таблицы рейсов, позволяя визуализировать последовательность шагов и действий, необходимых для этого процесса. Эта диаграмма акцентирует внимание на активностях и переходах между ними, что способствует лучшему пониманию логики обновления данных о рейсах.

Вторая диаграмма поведения моделирует процесс покупки билета, предоставляя взгляд на последовательность шагов, которые совершает пользователь при приобретении билета. Эта диаграмма состояний позволяет увидеть изменения состояний системы в ответ на действия пользователя, что полезно для оптимизации пользовательского опыта.

Обе диаграммы не только иллюстрируют ключевые процессы в авиакомпании, но и предоставляют ясное представление о том, как объекты взаимодействуют друг с другом в контексте этих действий. Результаты работы могут быть использованы для более глубокого понимания процессов авиакомпании и оптимизации системы в целом.

1. Описание программных средств

Как и в прошлой лабораторной работе, для построения диаграмм будем использовать Draw.io –бесплатное онлайн-приложение для создания диаграмм и схем.

Название: Draw.io

Версия: Веб-приложение

Разработчик: JGraph Ltd.

Адрес: https://www.draw.io/

Режим использования: Онлайн

Доступность на платформах: любой веб-браузер.

Draw.io позволяет создавать различные типы диаграмм, такие как блок-схемы, организационные диаграммы, UML-диаграммы, сетевые диаграммы и многое другое.

1. Описание практического задания

Первой диаграммой была выбрана диаграмма деятельности – рисунок 1.

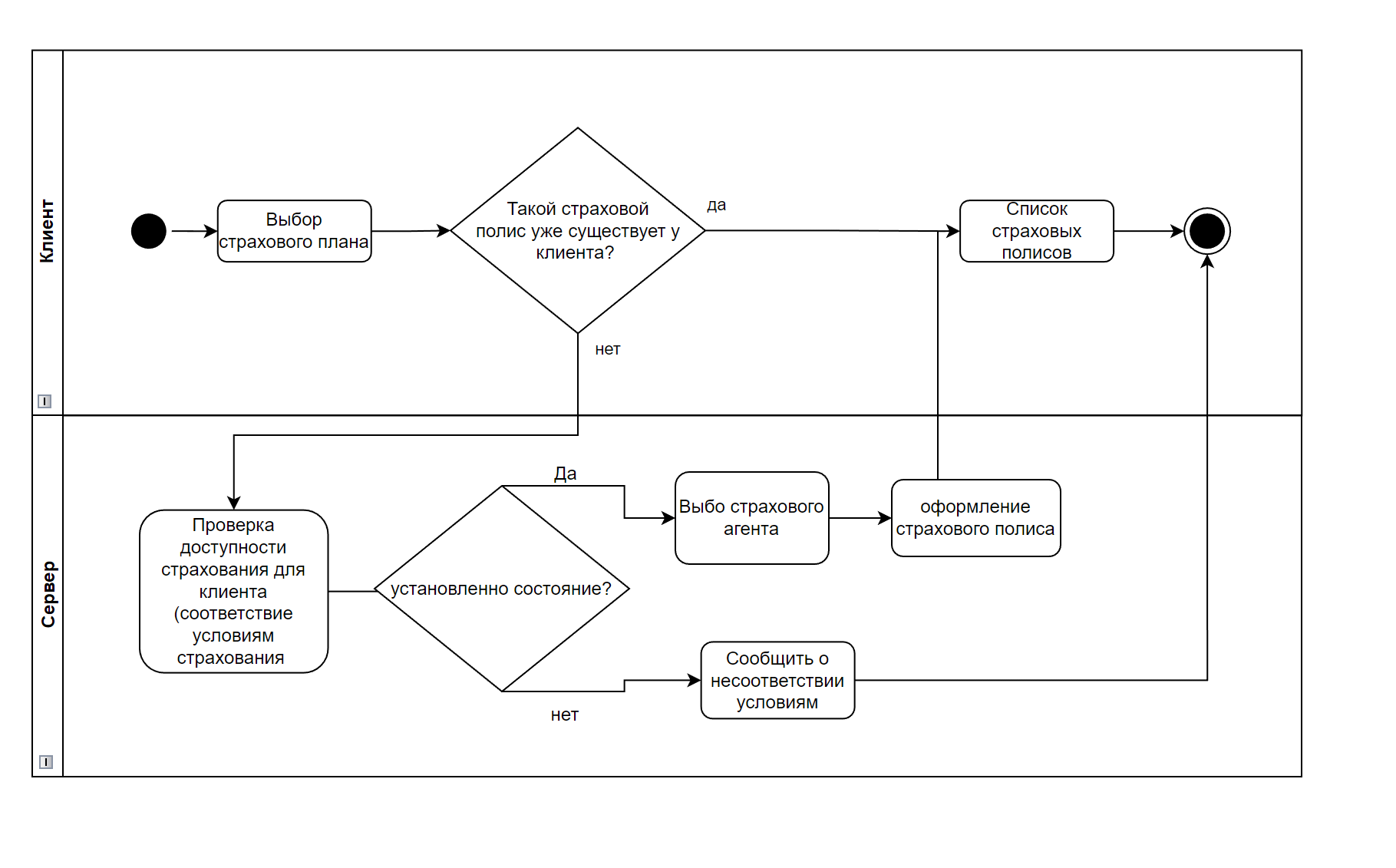


Рисунок 1 – Диаграмма деятельности

Вторая диаграмма – диаграмма состояния:

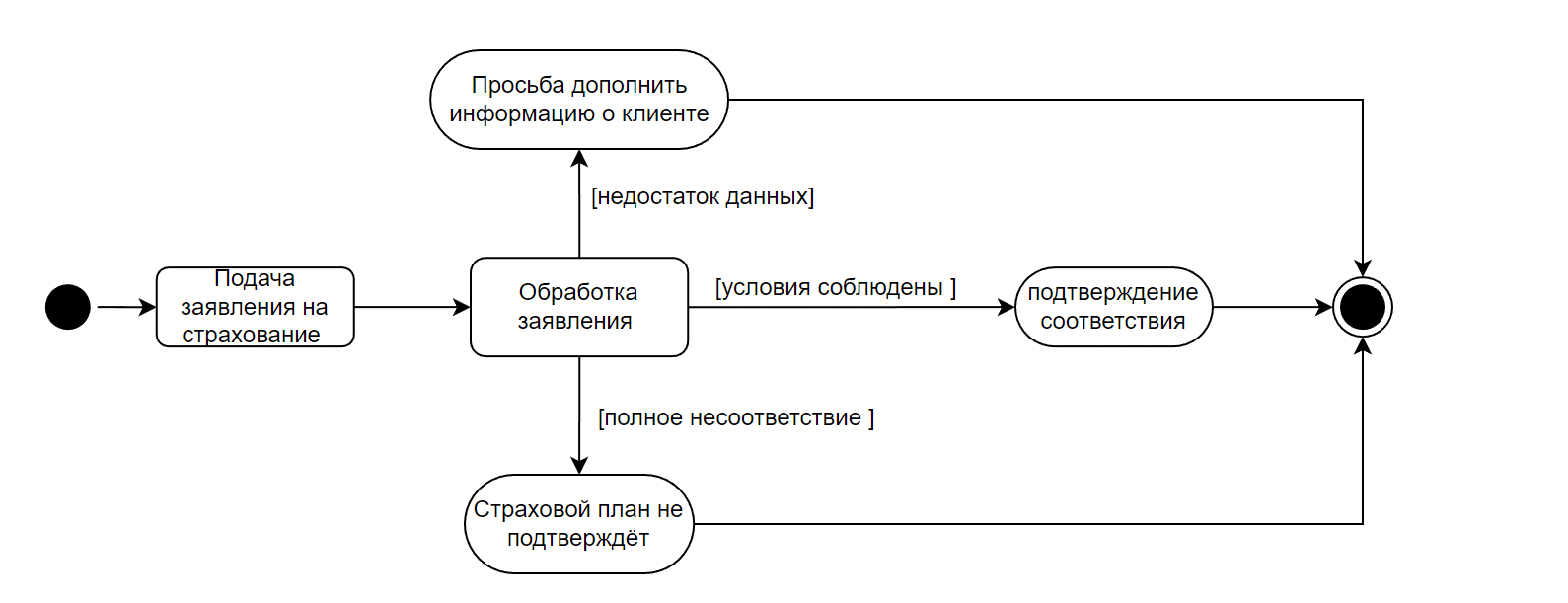
  
Рисунок 2 – Диаграмма состояния

Диаграмма состояний – это, по существу, диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов.

**Вывод:** Я изучил методологию объектно-ориентированного моделирования средствами UML. Получил дополнительные навыки проектирования моделей информационной системы с применением возможностей UML диаграмм поведения в контексте программного средства «Страховая компания».

**Ответы на теоретические вопросы**

1. **Укажите виды диаграмм поведения.**

Всего существует 3 вида диаграмм поведения:

* диаграмма деятельности;
* диаграмма состояний;
* диаграмма вариантов использования.

1. **Опишите назначение диаграммы деятельности.**

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

**3. Опишите основные нотации, которые используются на диаграмме состояний.**

| **Элемент/Нотация** | **Предназначение** |
| --- | --- |
| Пример | Класс (Class) |
| Пример | Состояние (State) |
| Пример | Состояние (StateEx) |
| Пример | Составное состояние (Composite state) |
| Пример | Разделитель (Concurrent state) |
| Пример | История (History) |
| Пример | Глубокая история (Deep history) |
| Пример | Начальное состояние (Start state) |
| Пример | Конечное состояние (Final state) |
| ПримерПример | Синхронизатор/разветвитель (Complex transition) |
| Пример | Переход (Transition) |
| Пример | Сообщение (Event message) |
| Пример | Точка изгиба связей (Point) |
| Пример | Комментарий (Note) |
| Пример | Коннектор комментария (Note connector) |

**4. Укажите виды связей между объектами на диаграмме последовательностей.**

Синхронное сообщение — отправитель передаёт ход управления актёру-получателю, которому необходимо провести в прецеденте некоторое действие. Пока проводимое получателем действие не будет завершено (не будет получено ответное сообщение), отправитель теряет возможность производить какие-либо действия. Графически изображается как сплошная линия со стрелкой в виде закрашенного треугольника, после которой идёт прямоугольник, отражающий деятельность объекта, в конце которого находится ответное сообщение.

Ответное сообщение — данное сообщение является ответом на синхронное сообщение. Обычно, содержит какое-либо возвращаемое изначальному отправителю значение, также возвращающее ему управление (возможность действовать). Графически изображается пунктирной линией с открытой стрелкой.

Асинхронное сообщение — отправитель передаёт ход управления получателю, которому необходимо провести в прецеденте некоторое действие. Основное отличие от синхронного сообщения состоит в том, что отправитель не теряет возможности совершать другие действия. Графически изображается сплошной линией с открытой стрелкой.

Потерянное сообщение — сообщение без адресата.

Найденное сообщение — сообщение без отправителя.

Последние два вида стрелок (взаимодействий) используются крайне редко. В основном они используются для демонстрации взаимодействия имеющихся объектов в данном прецеденте с внешними системами.

**5. Какая диаграмма позволяет моделировать параллельные вычисления?**

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.