

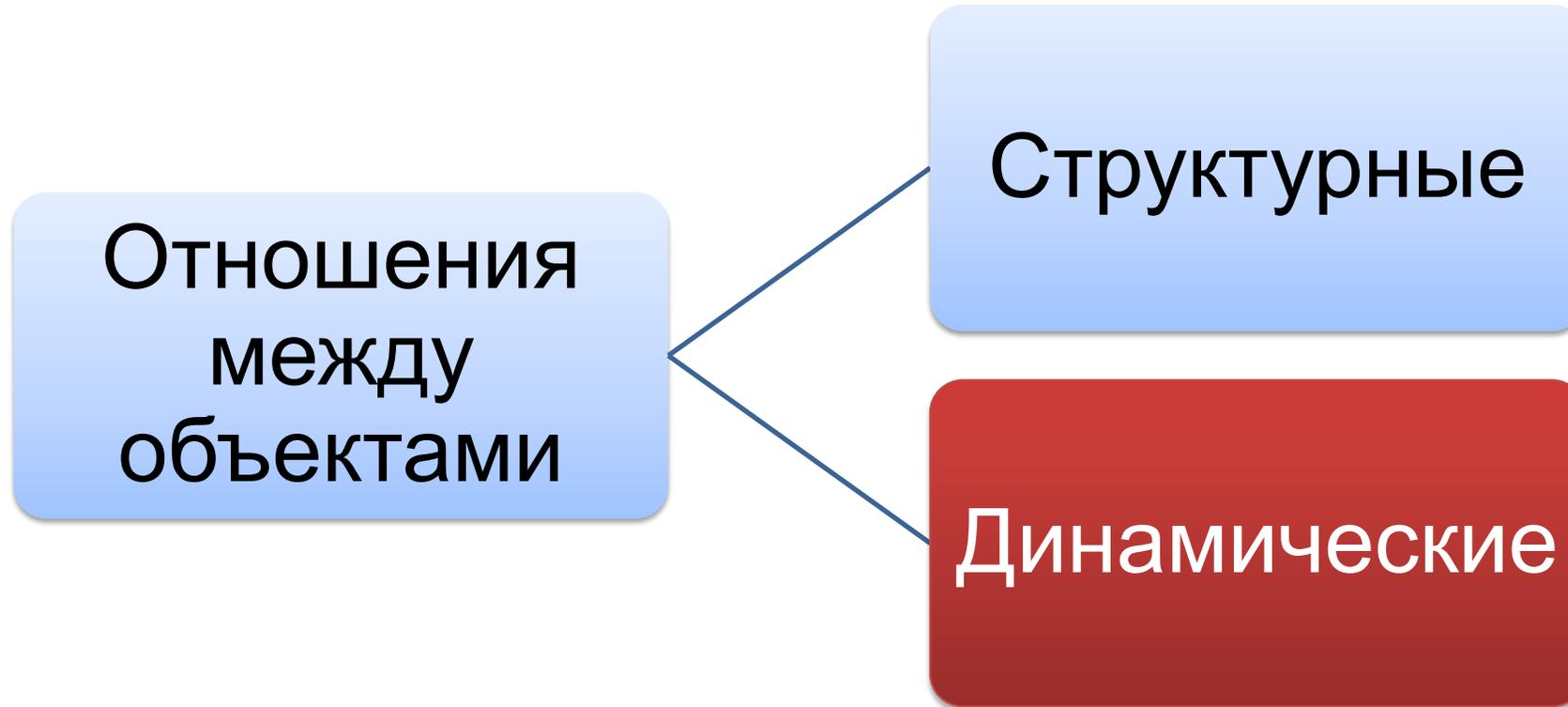
Моделирование динамических отношений на UML

1. Диаграммы поведения на UML
2. Диаграмма прецедентов (Use Case diagram)
3. Диаграмма последовательностей (Sequence diagram)
4. Диаграммы взаимодействий (Interaction diagram)
5. Диаграмма деятельности (Activity diagram)
6. Диаграмма состояний (Statechart diagram)

Преподаватель:

Ботов Дмитрий Сергеевич

Отношения между объектами



Диаграммы поведения на UML

Диаграмма прецедентов

- описывает функциональность, организацию поведения системы

Диаграмма взаимодействия

- описывает взаимодействие объектов в системе

Диаграмма состояний

- описывает изменение состояний в системе в ответ на события

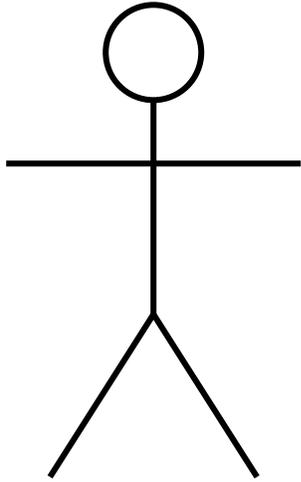
Диаграмма деятельности

- демонстрирует передачу управления от одной деятельности к другой

Применение диаграммы прецедентов (Use case diagram)

- Прецеденты служат UML-представлением **функций системы**
- Прецеденты описывают взаимодействия между **системой и пользователями**, уделяя основное внимание тому, что система «делает» для пользователя
- Модель прецедентов описывает **функциональное поведение** системы в целом или ее отдельных подсистем
- Модель прецедентов **НЕ специфицирует, КАК** функции будут реализованы в системе

Актор



Имя
актера

Актор – это находящийся **вне описываемой системы пользователь** или **другая система**, **взаимодействующие с ней**

Акторами на диаграмме прецедентов могут быть:

- **Роли пользователей** в системе
- **Внешние системы и устройства**, взаимодействующие с моделируемой системой (интегрированные с ней)

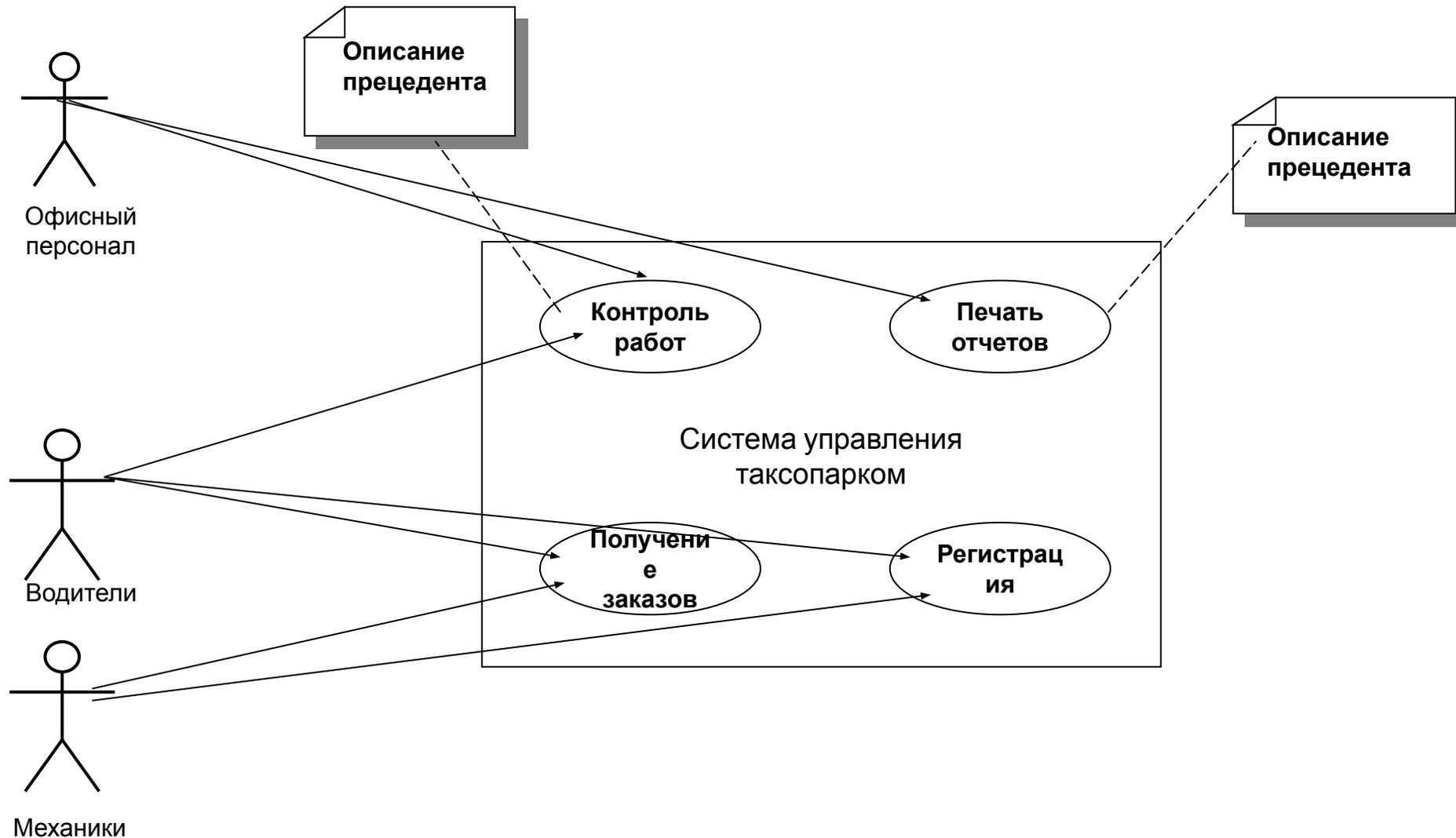
Прецедент

Прецедент - описание **множества последовательностей действий** (включая варианты), выполняемых системой для того, чтобы **актор** мог получить **определенный результат**.



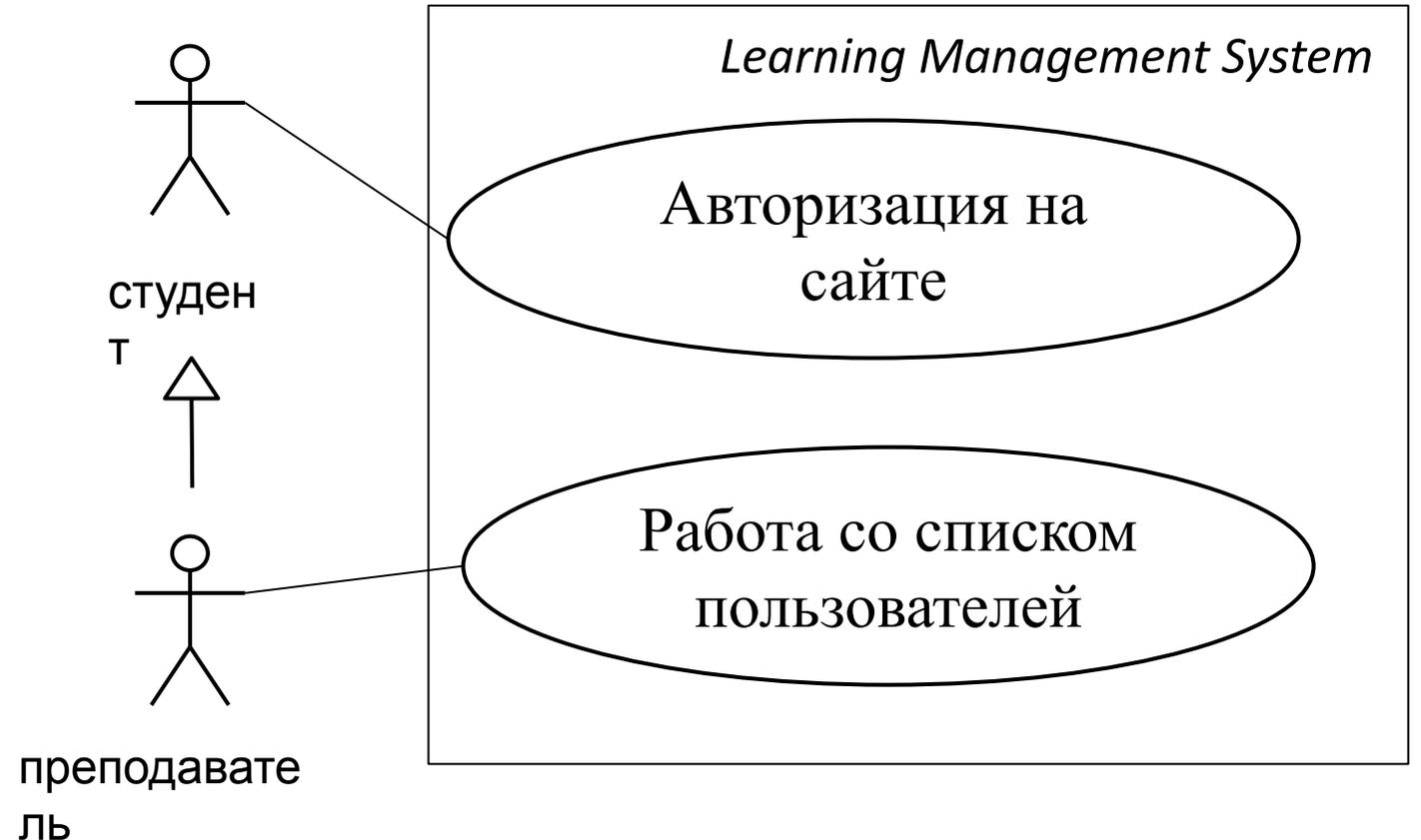
- Отражает типичное взаимодействие пользователя и компьютерной системы
- Охватывает некоторую очевидную для пользователя функцию
- Может быть как небольшим, так и достаточно крупным
- Решает некоторую дискретную задачу пользователя

Пример диаграммы прецедентов



Отношение обобщения между акторами

- Применяется для того, чтобы перенять от родительского актора базовые характеристики и прецеденты, а затем добавить к ним особенности.
- Обобщение специализирует актора или создаёт его разновидность.

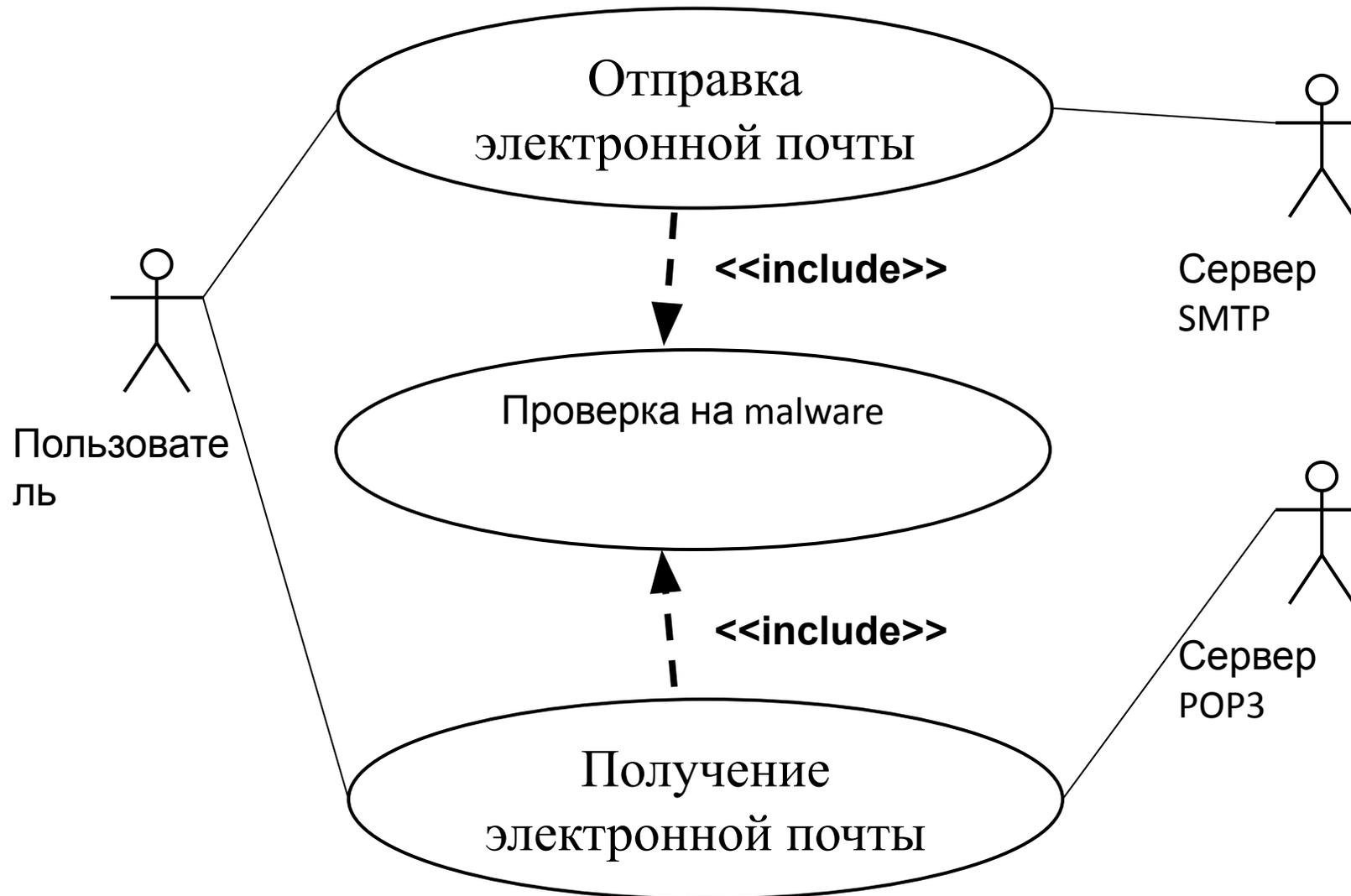


Связь «включение» между прецедентами

- Обозначается `<<include>>`
- Применяется, когда имеется какой-либо фрагмент поведения системы, который повторяется более, чем в одном варианте использования
- Отпадает необходимость дублировать одинаковые варианты поведения

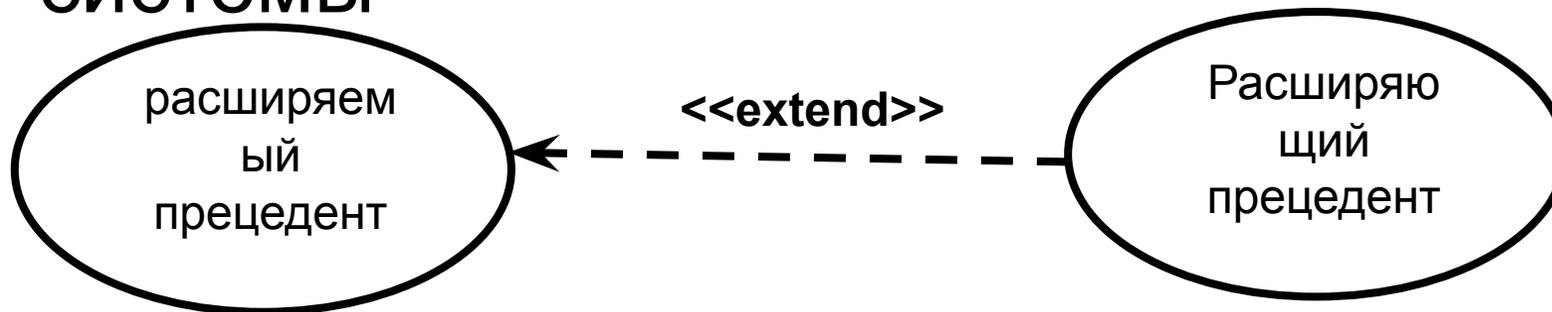


Пример включения прецедентов

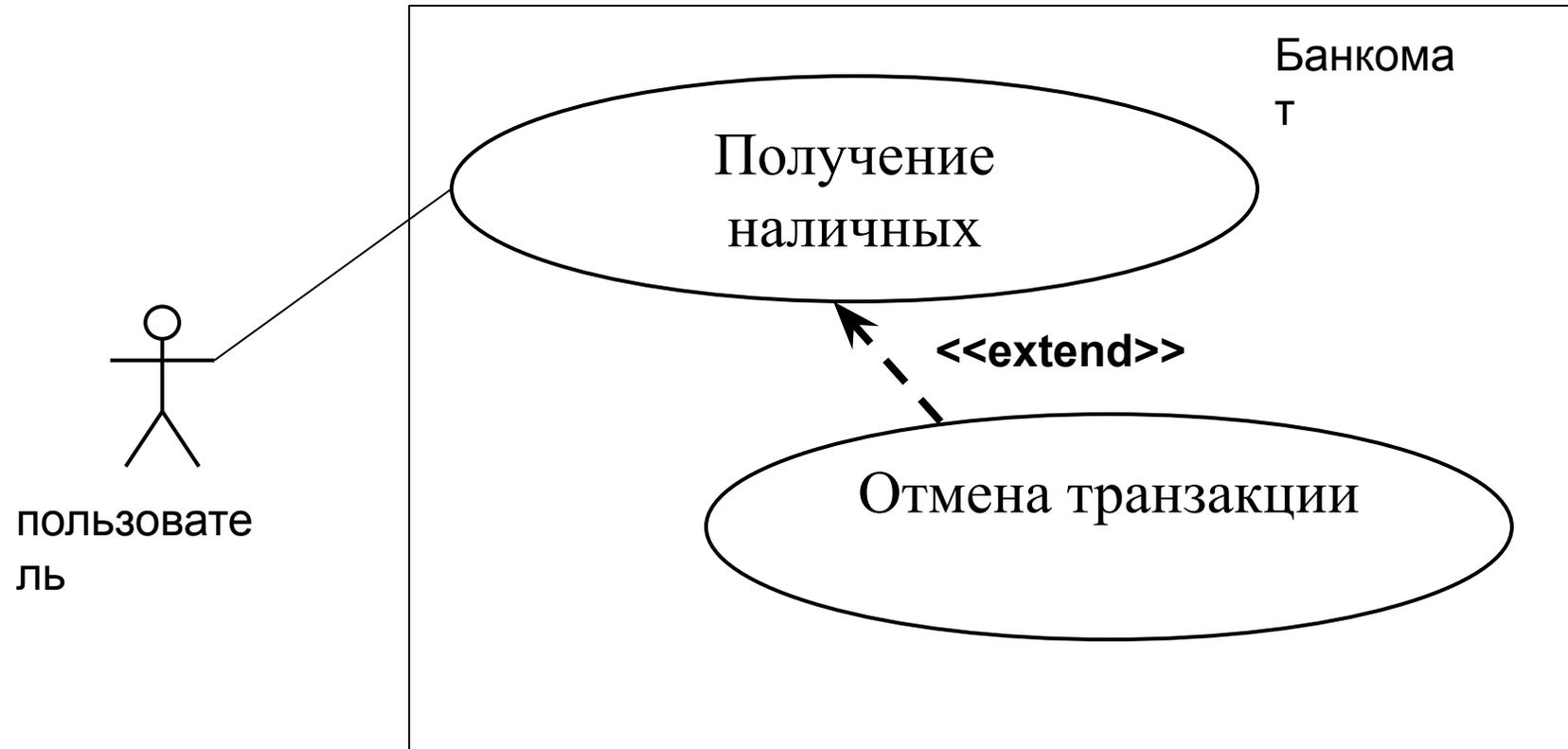


Связь «расширение» между прецедентами

- Обозначается **<<extend>>**
- Применяется в тех случаях, когда один вариант использования подобен другому, но несет несколько большую нагрузку
- Следует применять, когда вы описываете альтернативные варианты в типичном поведении системы



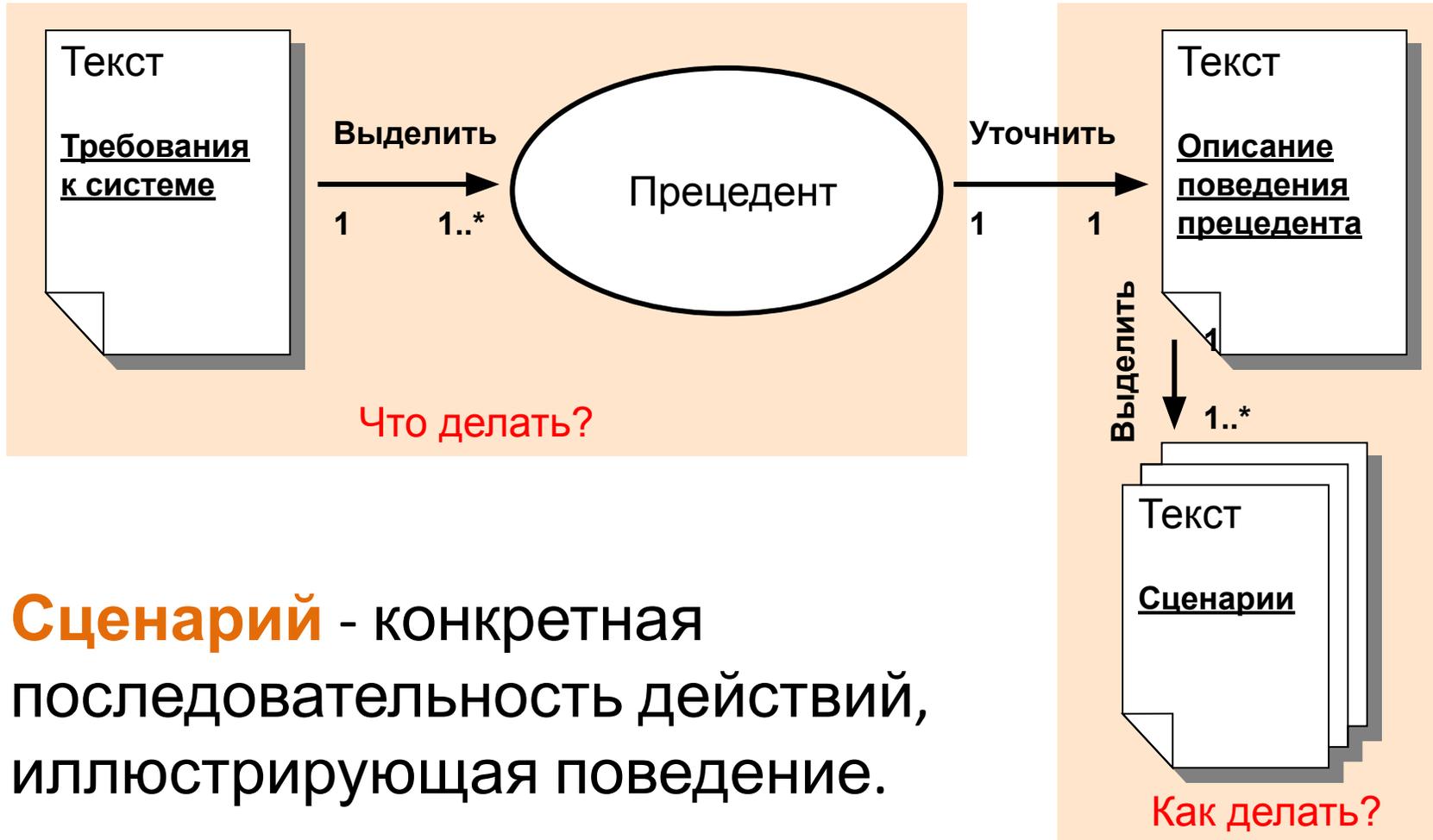
Пример применения расширения прецедентов



Применение связей между прецедентами



Документирование прецедентов



Сценарий - конкретная последовательность действий, иллюстрирующая поведение.

Формы сценариев

- Простая текстовая форма:

Пользователь вставляет карточку.

Система запрашивает PIN.

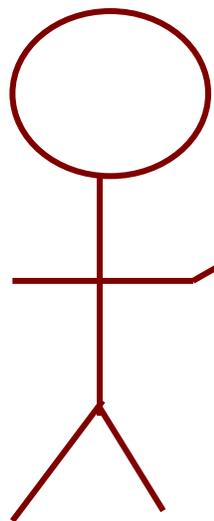
Пользователь вводит PIN.

Система проверяет PIN.

- Таблица из двух столбцов:

<u>Действия пользователя</u>	<u>Реакция системы</u>
Вставка карточки	Считывание номера карточки Запрос PIN
Ввод PIN	Проверка PIN

Упражнение



Опишите сценарии
работы автомата
по приготовлению кофе

Диаграммы взаимодействий

Диаграмма взаимодействий описывает **взаимодействия**, состоящие из множества объектов и отношений между ними, включая сообщения, которыми они обмениваются.

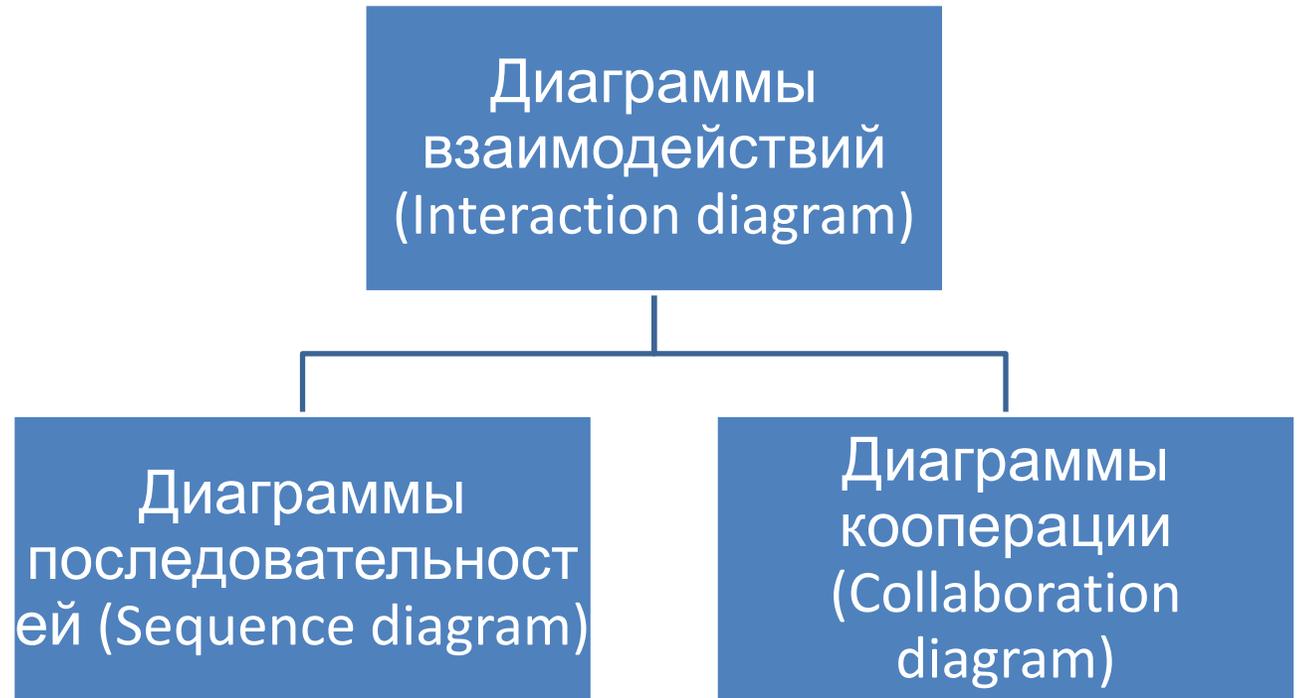
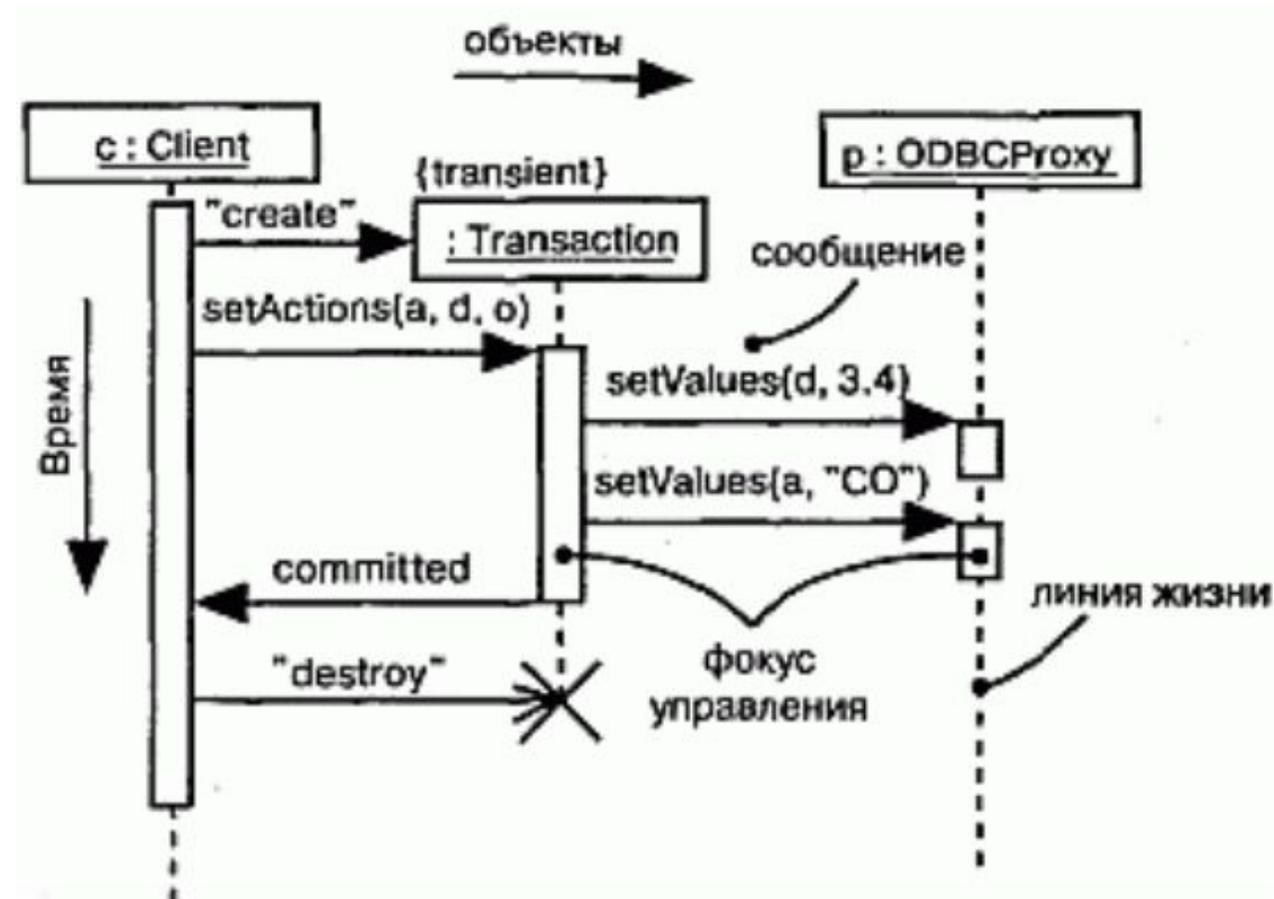
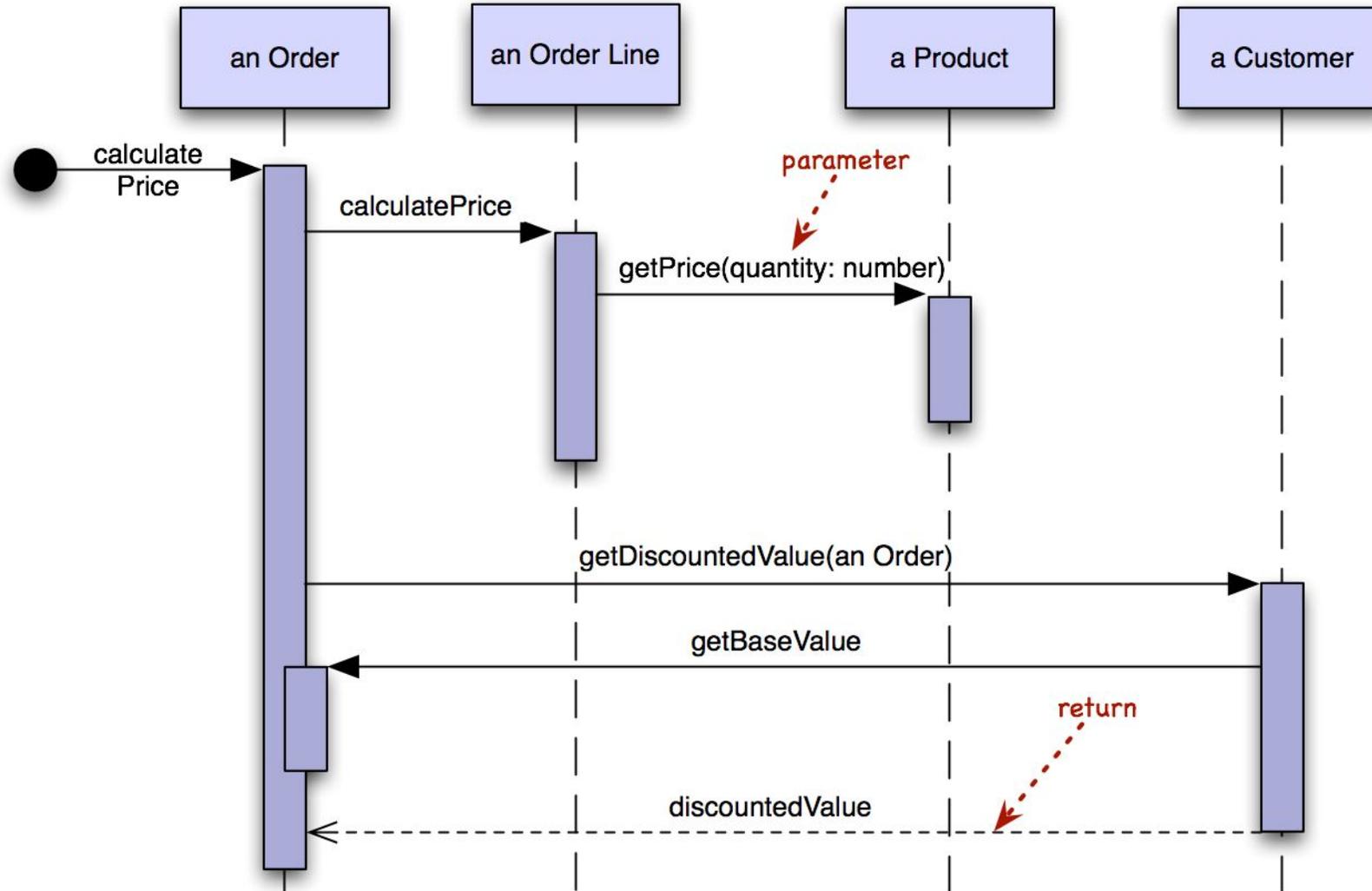


Диаграмма последовательностей (Sequence diagram)

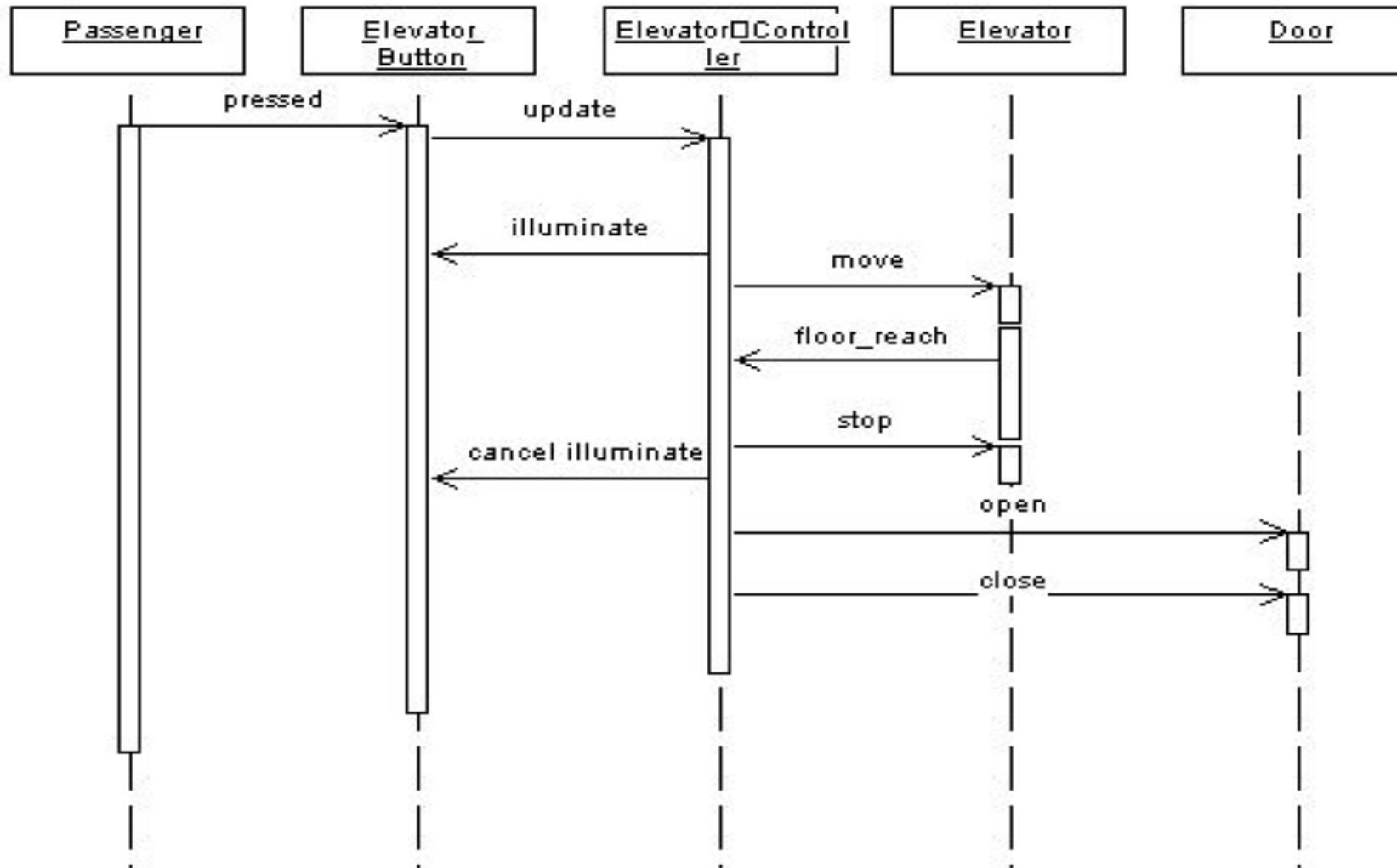
- Отображает взаимодействие объектов в динамике
- Акцентирует внимание на **временной упорядоченности сообщений**
- Иллюстрирует логику развития сценария использования (use case)



Пример диаграммы последовательностей (1)



Пример диаграммы последовательностей (2)



Пример диаграммы последовательностей (3)

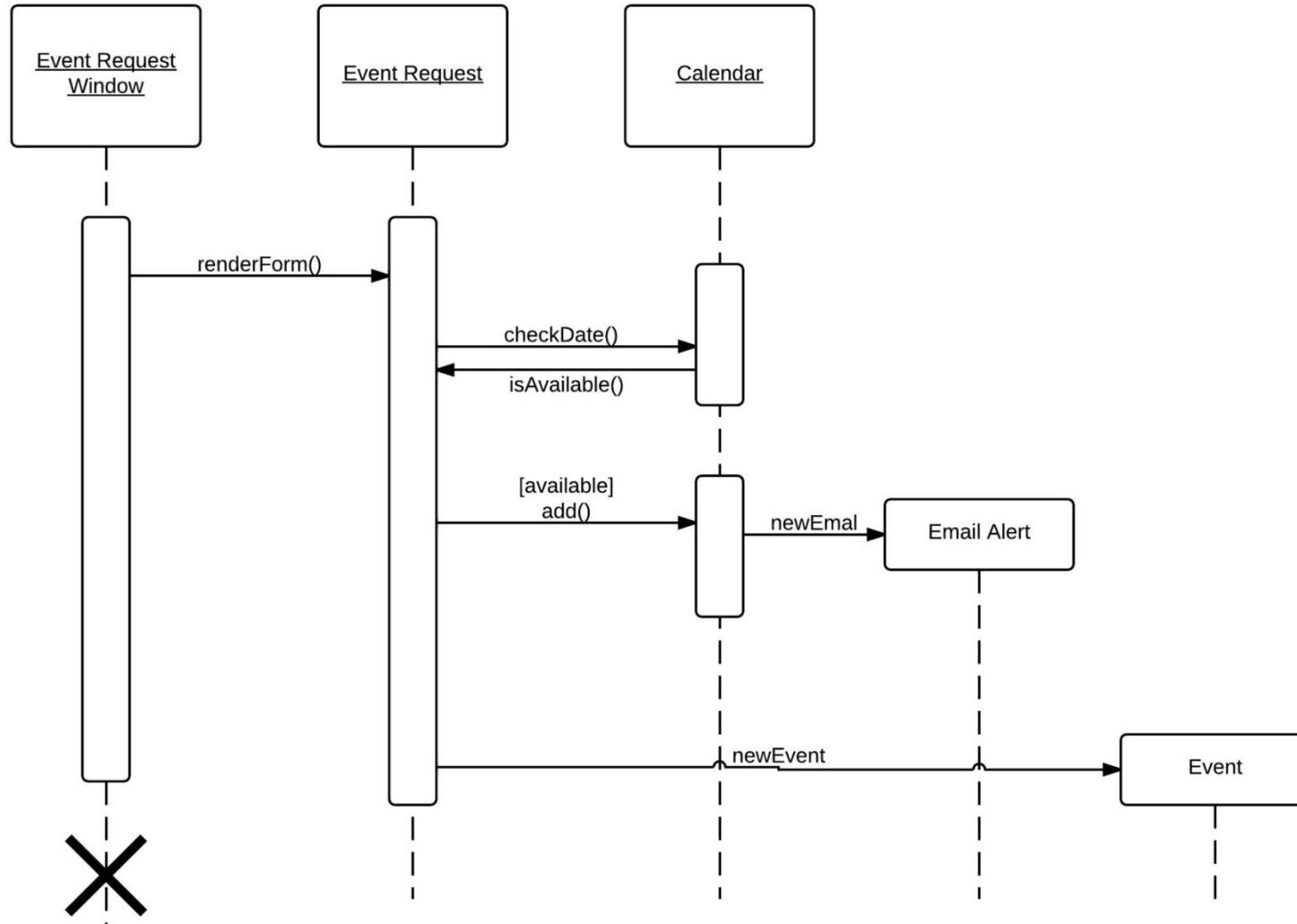
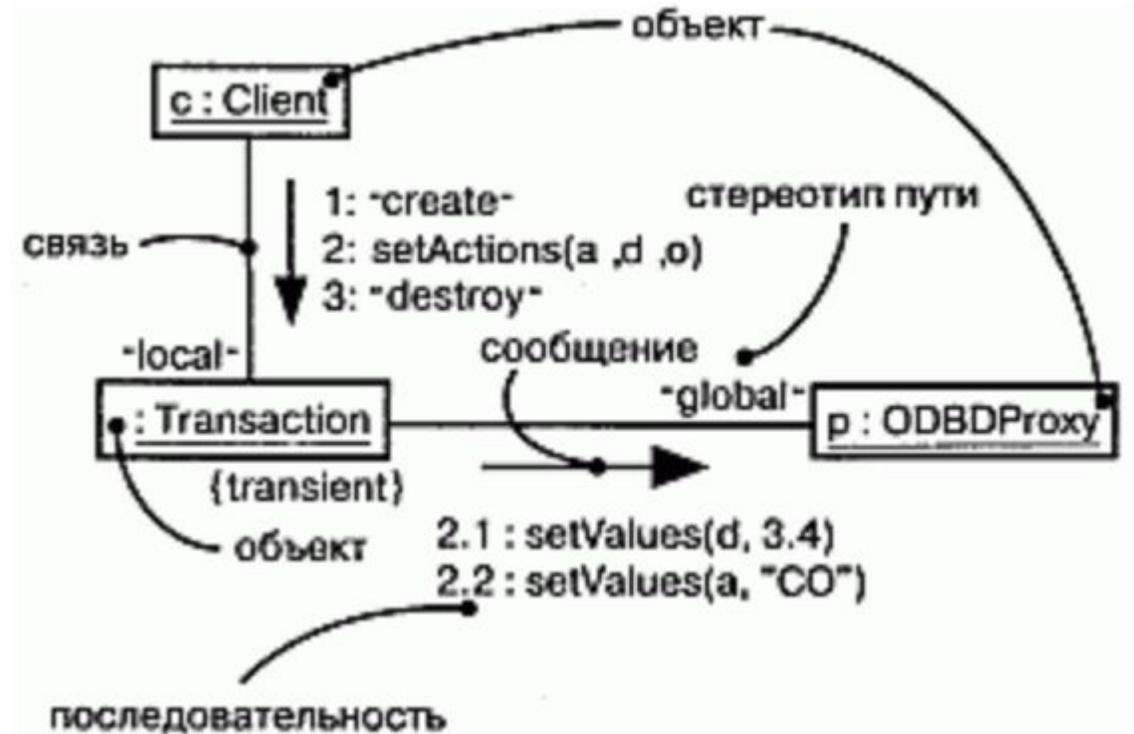


Диаграмма кооперации (Interaction diagram)

- акцентирует внимание на **структурной организации объектов**, принимающих участие во взаимодействии
- описывает **пути взаимодействия** объектов и **очередность передачи сообщений**
- Семантически эквивалентна **диаграмме последовательностей**



Пример диаграммы взаимодействия

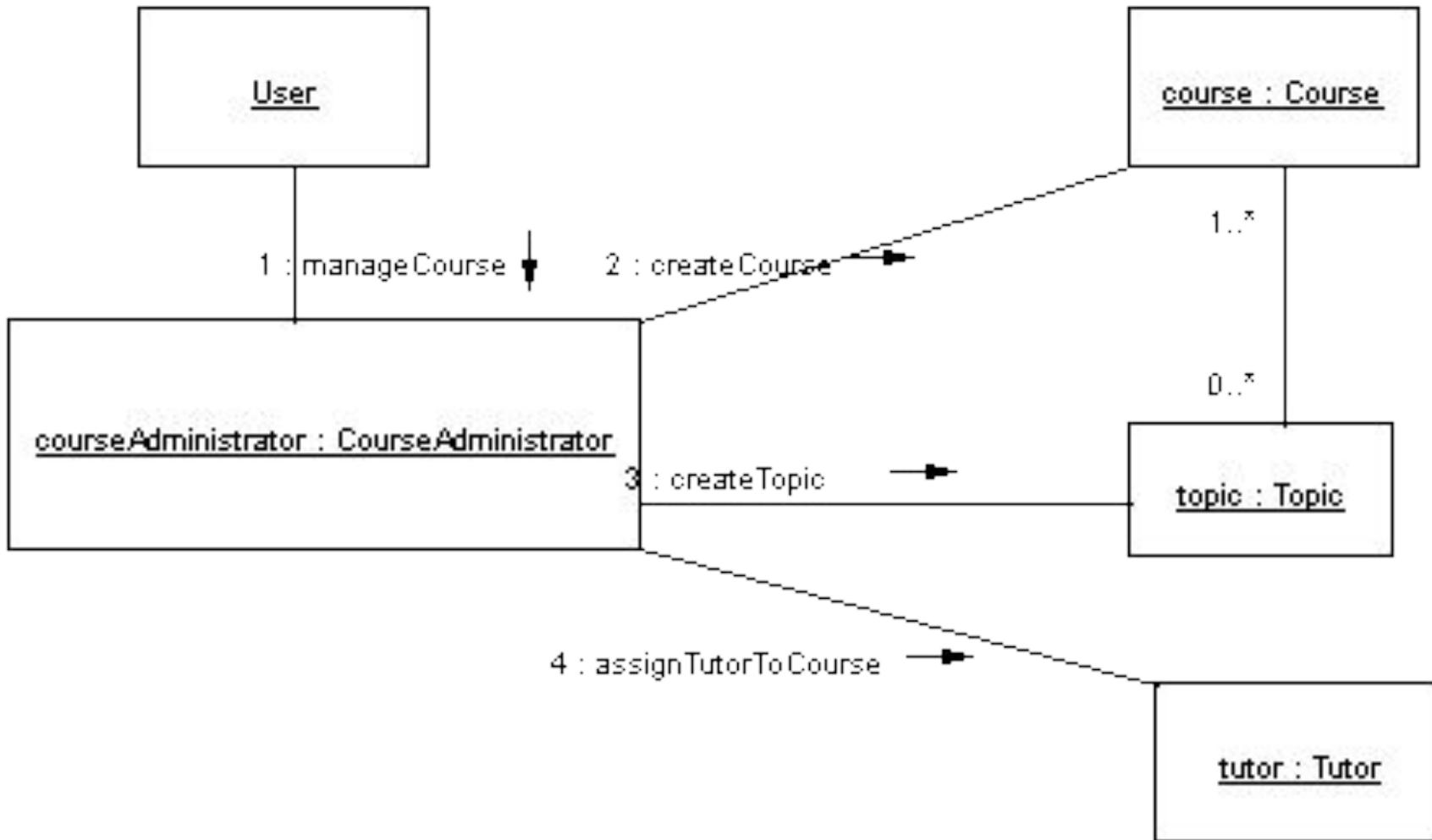
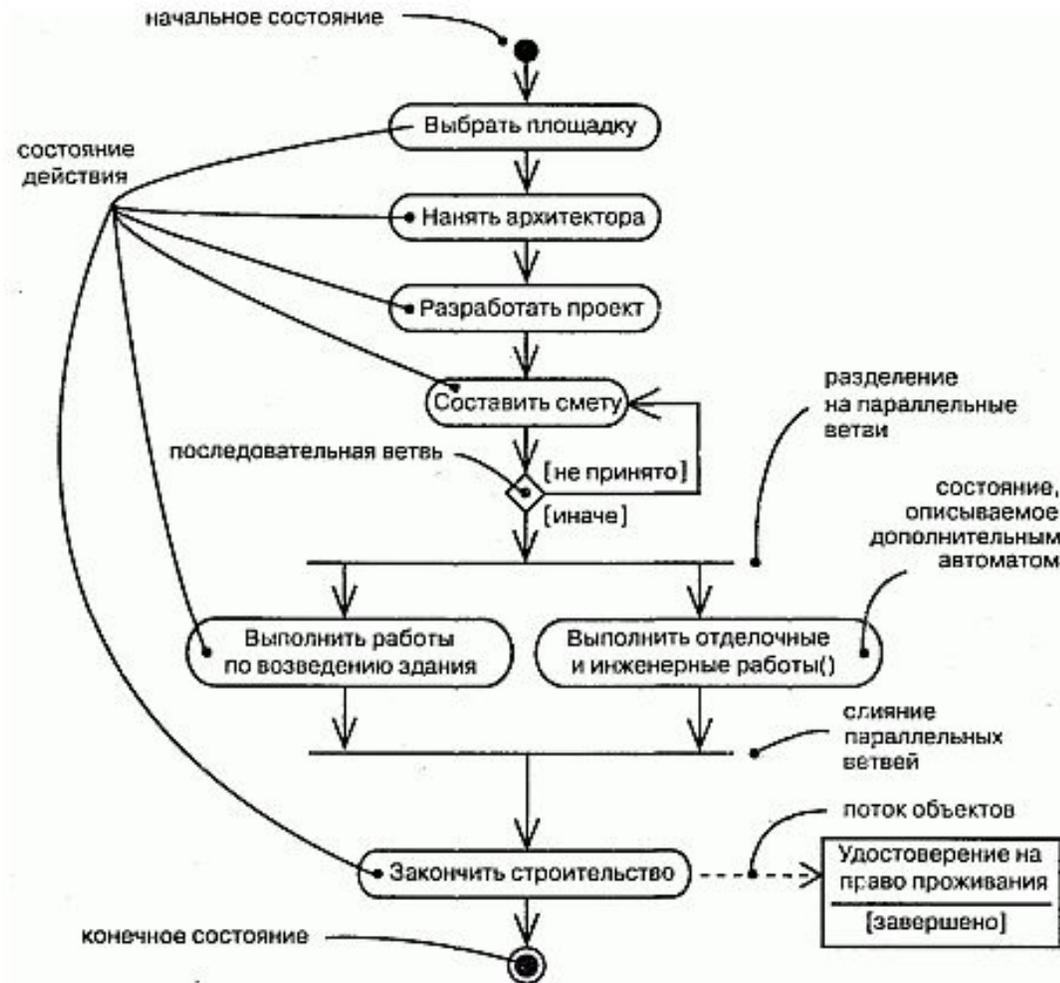


Диаграмма деятельности (Activity diagram)

- *Диаграмма деятельности* показывает **поток переходов** от одной деятельности к другой.
- Такие диаграммы детализируют особенности алгоритмической и логической реализации операций, выполняемых системой
- Напоминают **блок-схемы алгоритма**



Пример диаграммы деятельности

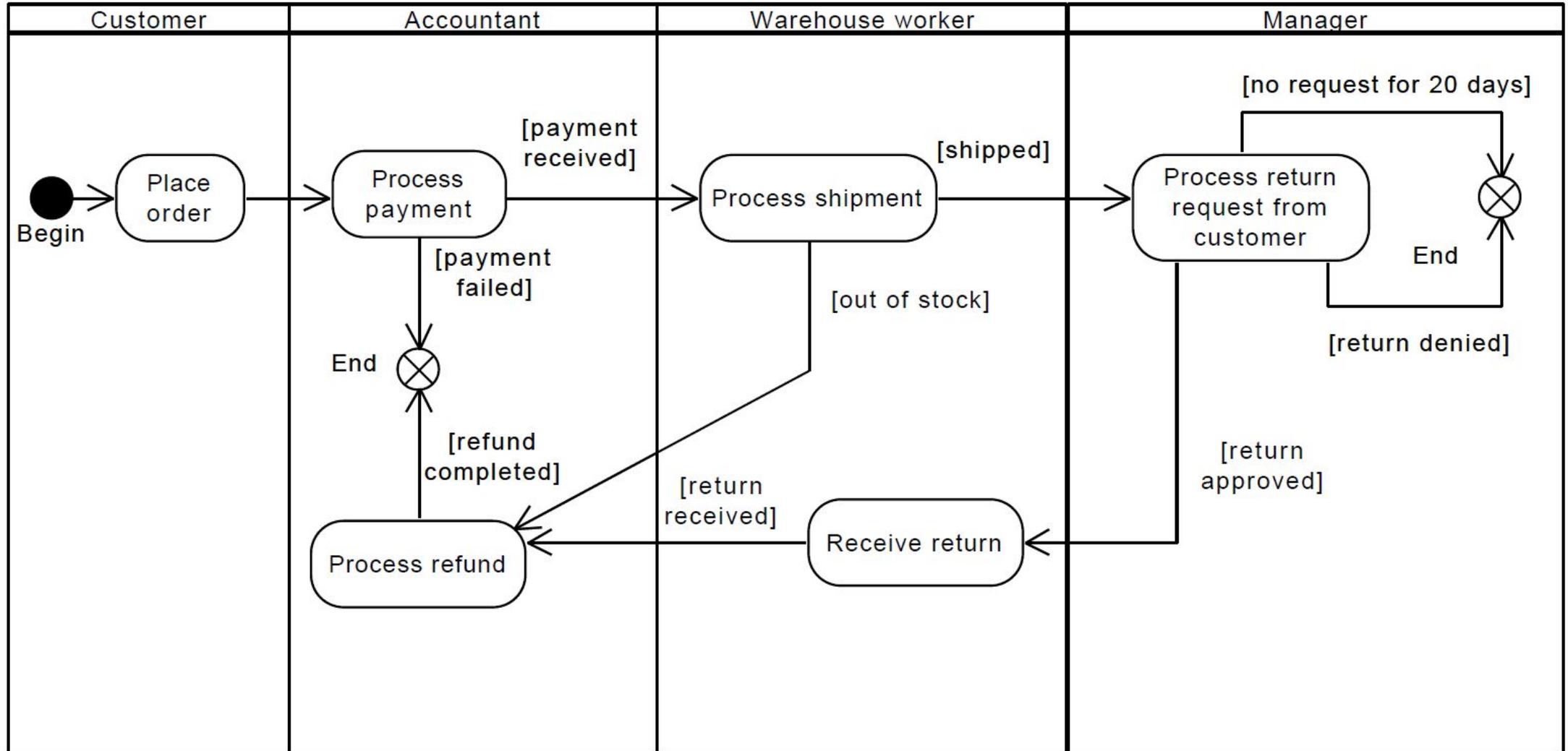
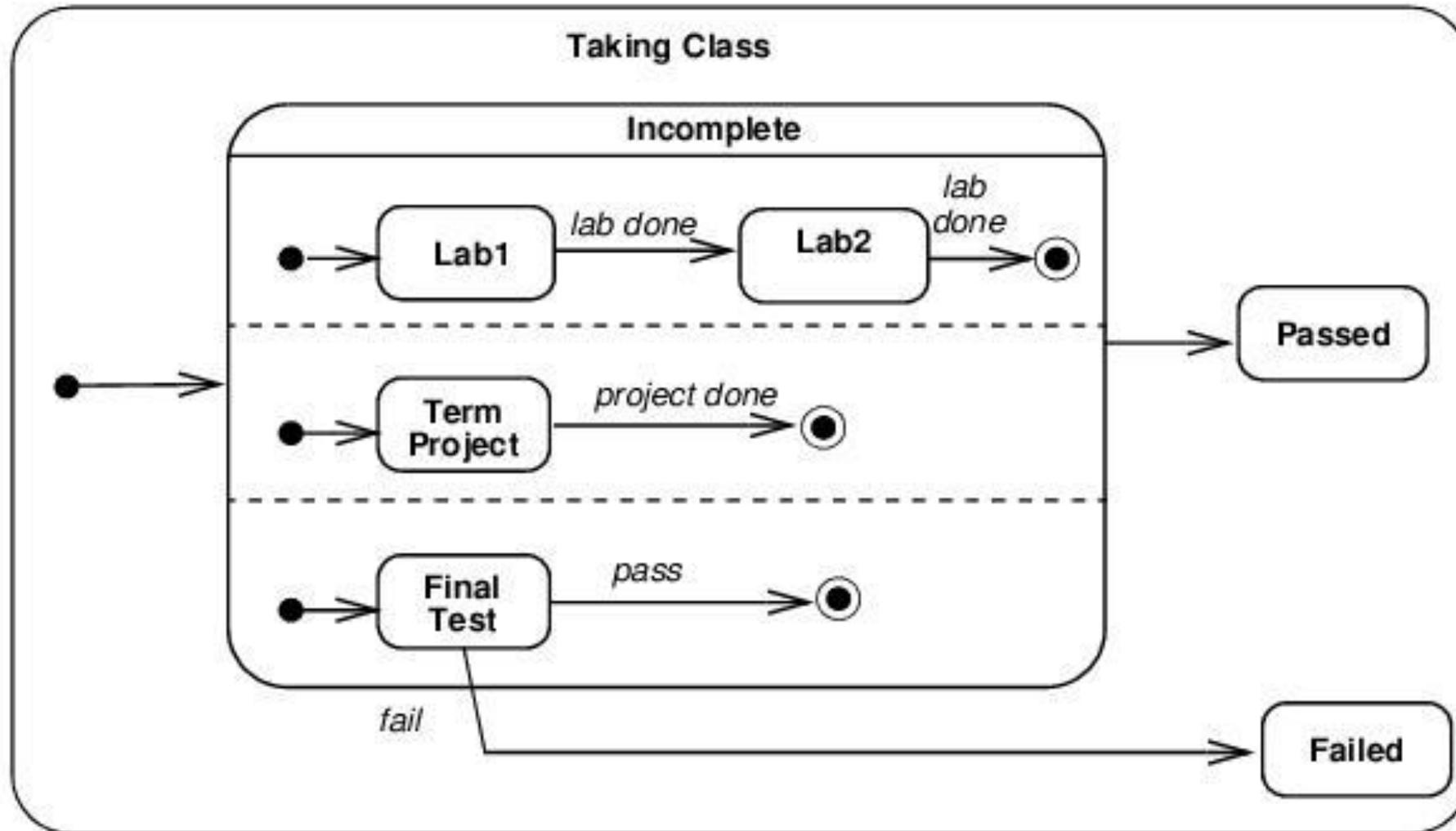


Диаграмма состояний (Statechart diagram)

- Объекты имеют как поведение, так и **состояние**
- В каждый момент времени объект находится в некотором определенном состоянии
- Диаграмма состояний показывает, как объект переходит из одного состояния в другое
- Пример:



Пример диаграммы состояний



Резюме: рассмотренные вопросы

- Какие существуют UML-диаграммы для моделирование динамических аспектов системы?
- Как описать функциональность системы с помощью **диаграммы прецедентов**?
- Как смоделировать взаимодействие объектов во времени с помощью **диаграмм взаимодействия**?
- Как описать алгоритмическую последовательность действий в системе с помощью **диаграммы деятельности**?
- Как описать состояния объектов и переходы между ними с помощью **диаграммы состояний**?