

Концептуальная  
модель uml.

Виды диаграмм.

{

Диаграммы UML есть та основная накладываемая на модель структура, которая облегчает создание и использование модели.

▫ Диаграмм (diagram) — это графическое представление некоторой части графа модели.

# Классификация диаграмм

- ▣ Диаграмма использования (Use Case diagram).
- ▣ Диаграмма классов (Class diagram).
- ▣ Диаграмма объектов (Object diagram).
- ▣ Диаграмма состояний (State chart diagram).
- ▣ Диаграмма деятельности (Activity diagram).
- ▣ Диаграмма последовательности (Sequence diagram).
- ▣ Диаграмма кооперации (Collaboration diagram).
- ▣ Диаграмма компонентов (Component diagram).
- ▣ Диаграмма размещения (Deployment diagram).

# Условная классификация диаграмм

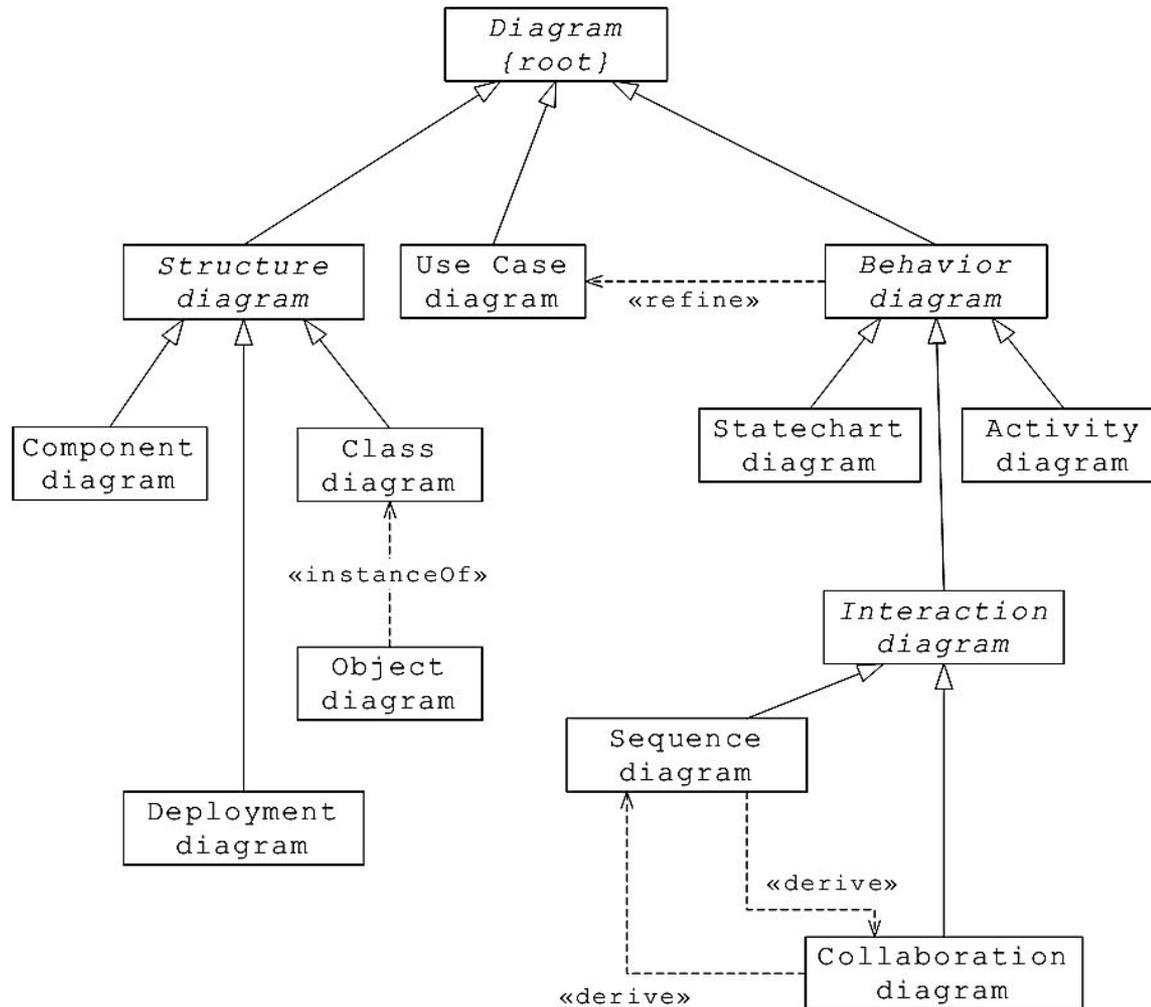


Рис. 1.7. Иерархия типов диаграмм для UML 1

## Типы диаграмм для заголовка

Название диаграммы	Тег (стандартный)	Тег (предлагаемый)
Диаграмма использования	use case или uc	use case
Диаграмма классов	class	class
Диаграмма автомата	state machine или stm	state machine
Диаграмма деятельности	activity или act	activity
Диаграмма последовательности	interaction или sd	sd
Диаграмма коммуникации	interaction или sd	comm
Диаграмма компонентов	component или cmp	component
Диаграмма размещения	не определен	deployment
Диаграмма объектов	не определен	object
Диаграмма внутренней структуры	class	class или component
Обзорная диаграмма взаимодействия	interaction или sd	interaction
Диаграмма синхронизации	interaction или sd	timing
Диаграмма пакетов	package или pkg	package

# ОБЩИЕ ДИАГРАММЫ

- ▣ Все диаграммы UML можно условно разбить на две группы, первая из которых — общие диаграммы.
- ▣ Общие диаграммы практически не зависят от предмета моделирования и могут применяться в любом программном проекте без оглядки на предметную область, область решений и т. д.

# Диаграмма использования

- ▣ Диаграмма использования (use case diagram) — это наиболее общее представление функционального назначения системы.
- ▣ Диаграмма использования призвана ответить на главный вопрос моделирования: что делает система во внешнем мире?

На диаграмме использования применяются два типа основных сущностей: варианты использования (1) и действующие лица (2), между которыми устанавливаются следующие основные типы отношений:

- ▣ ассоциация между действующим лицом и вариантом использования (3);
- ▣ обобщение между действующими лицами (4);
- ▣ обобщение между вариантами использования (5);
- ▣ зависимости между вариантами использования (6).

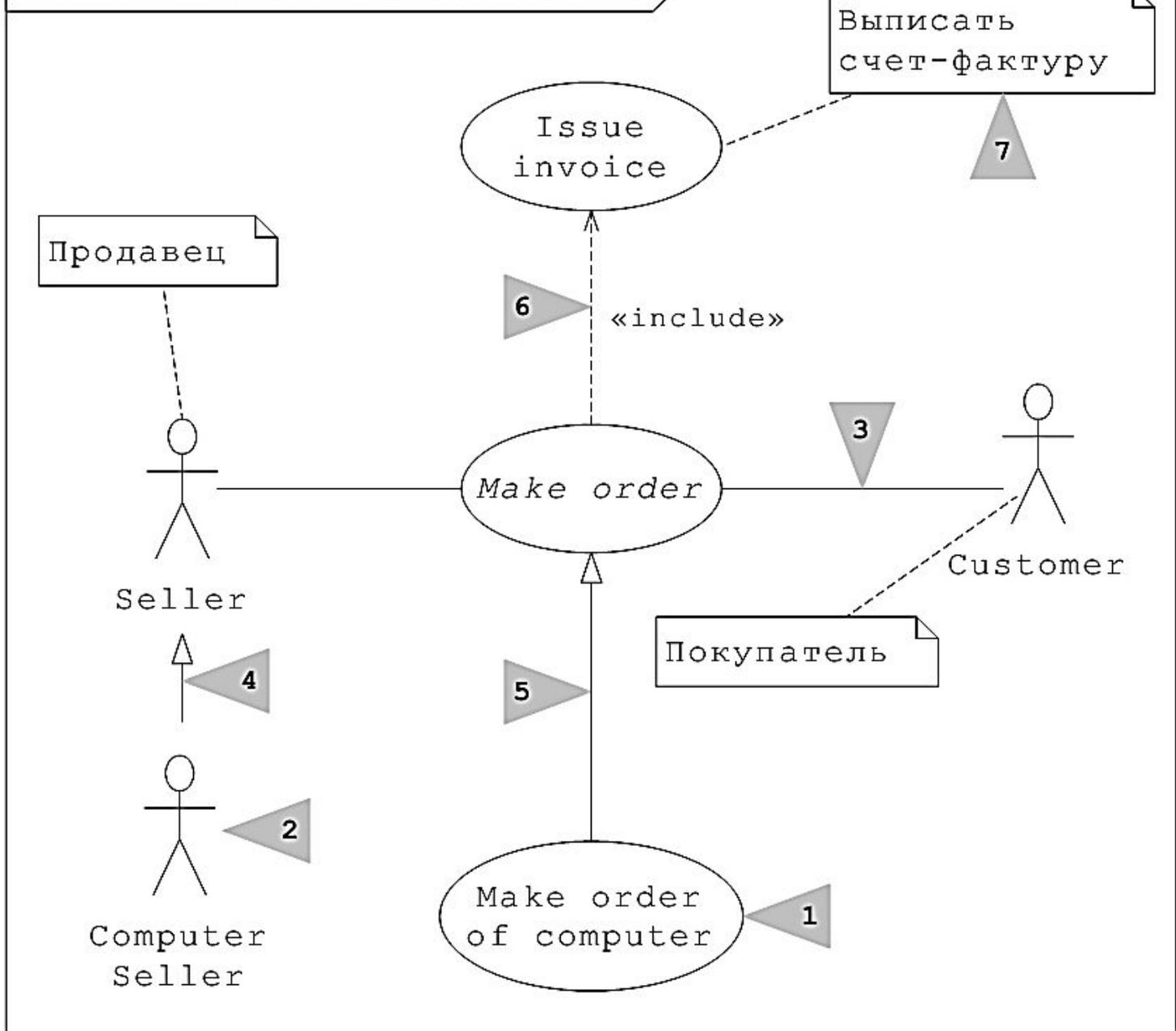


Рис. 1.11. Нотация диаграммы использования

# Диаграмма классов

- Диаграмма классов (class diagram) — основной способ описания структуры системы.
- На диаграмме классов применяется один основной тип сущностей: классы (1) (включая многочисленные частные случаи классов: интерфейсы, примитивные типы, классы4ассоциации и многие другие), между которыми устанавливаются следующие основные типы отношений:
  - ассоциация между классами (2) (с множеством дополнительных подробностей);
  - обобщение между классами (3);
  - зависимости (различных типов) между классами (4) и между классами и интерфейсами.

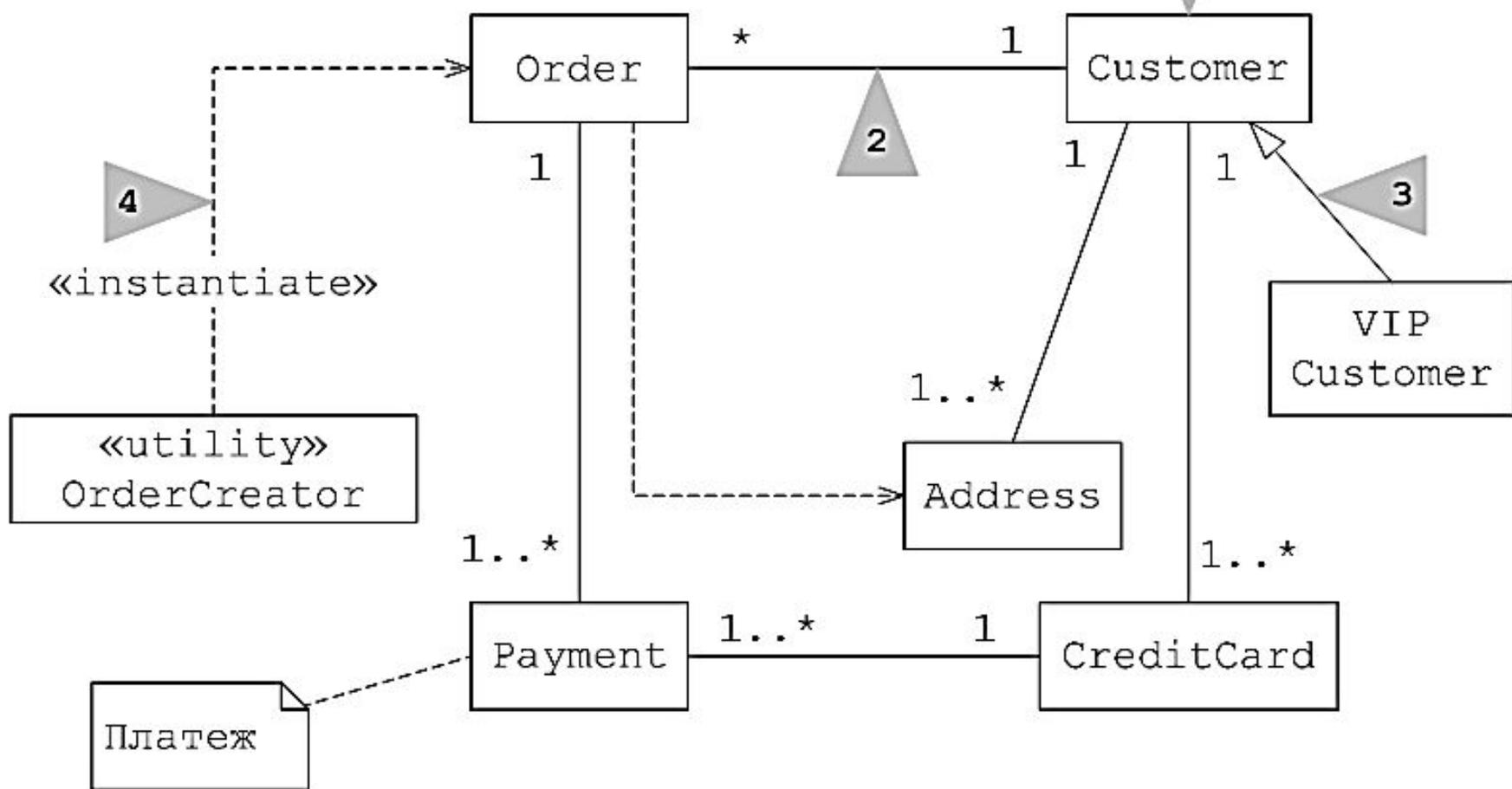


Рис. 1.12. Нотация диаграммы классов

# Диаграмма состояний

Диаграмма состояний (автомата) в UML 1 (state chart diagram) — это один из способов детального описания поведения в UML.

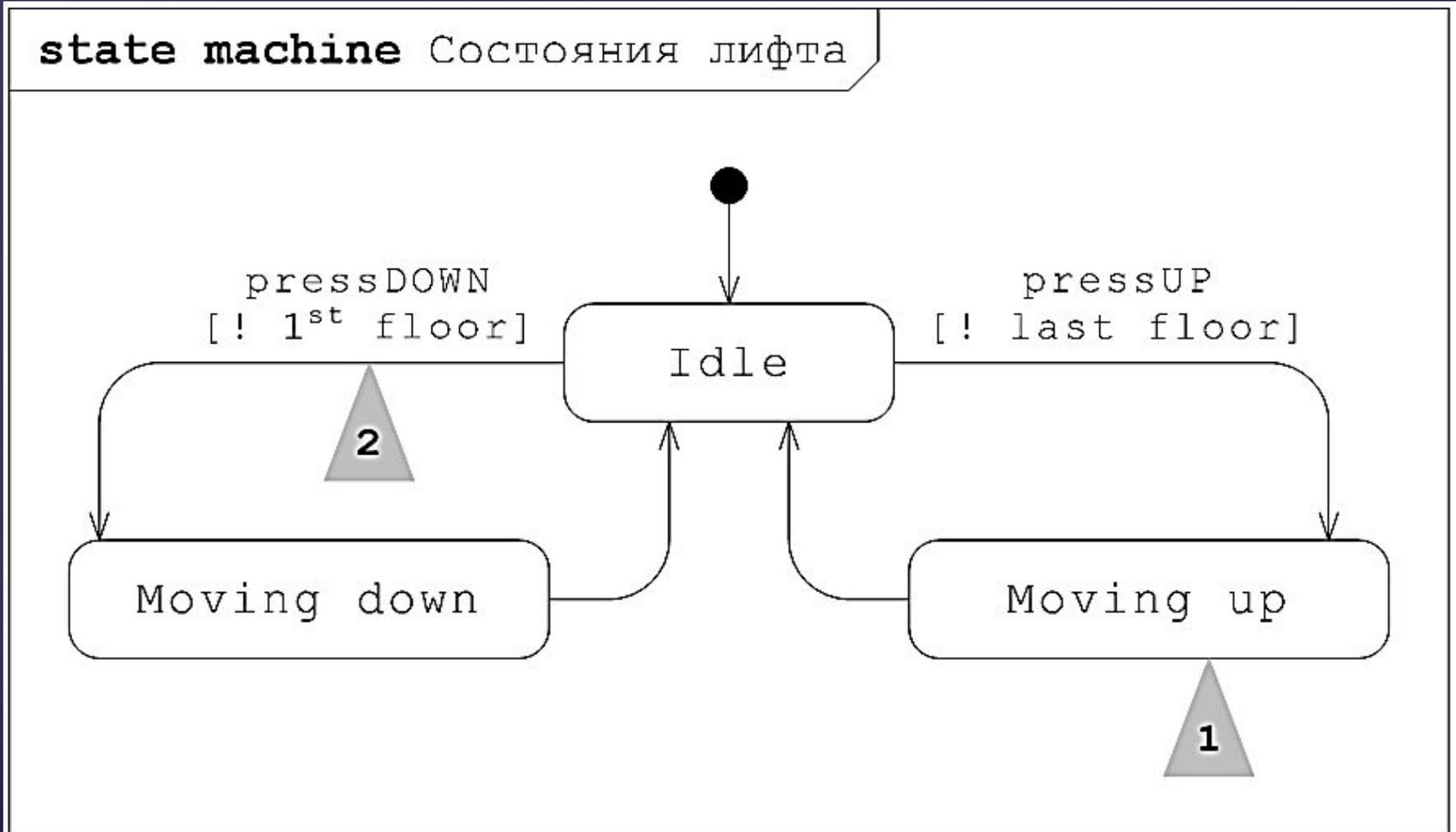


Рис. 1.13. Нотация диаграммы автомата

# Диаграмма деятельности

- ▣ Диаграмма деятельности (activity diagram) — еще один способ описания поведения, который визуально напоминает старую добрую блок-схему алгоритма.

На диаграмме деятельности применяют один основной тип сущностей — действие (1), и один тип отношений — переходы (2) (передачи управления). Также используются такие конструкции как развилки, слияния, соединения, ветвления (3), которые похожи на сущности.

**activity** Сборка приложения

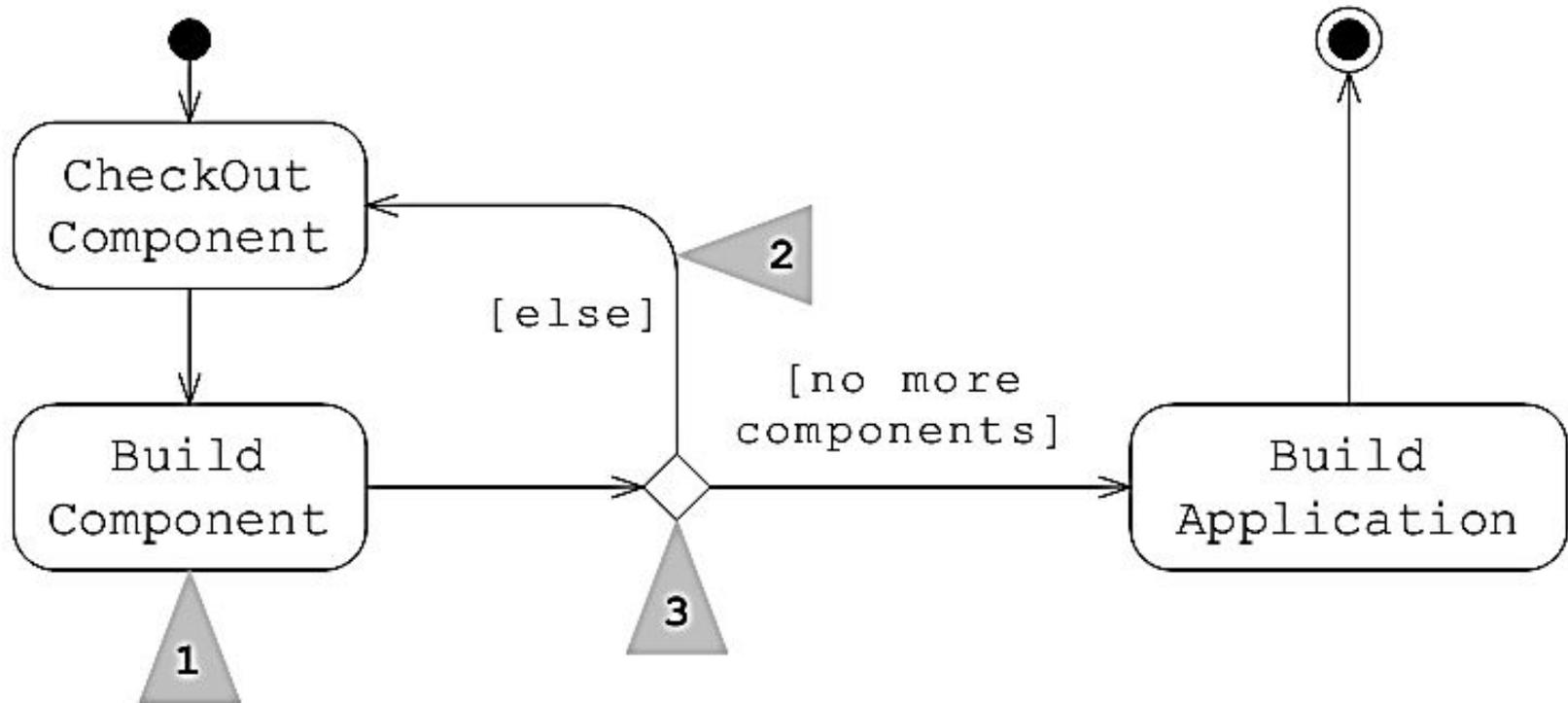


Рис. 1.14. Нотация диаграммы деятельности

# Диаграмма последовательности

Диаграмма последовательности (sequence diagram) — это способ описания поведения системы "на примерах".

Фактически, диаграмма последовательности — это запись протокола конкретного сеанса работы системы (или фрагмента такого протокола).

На диаграмме последовательности применяют один основной тип сущностей — экземпляры взаимодействующих классификаторов (1) (в основном классов, компонентов и действующих лиц), и один тип отношений — связи (2), по которым происходит обмен сообщениями (3).

**sd** Печать через сервер печати

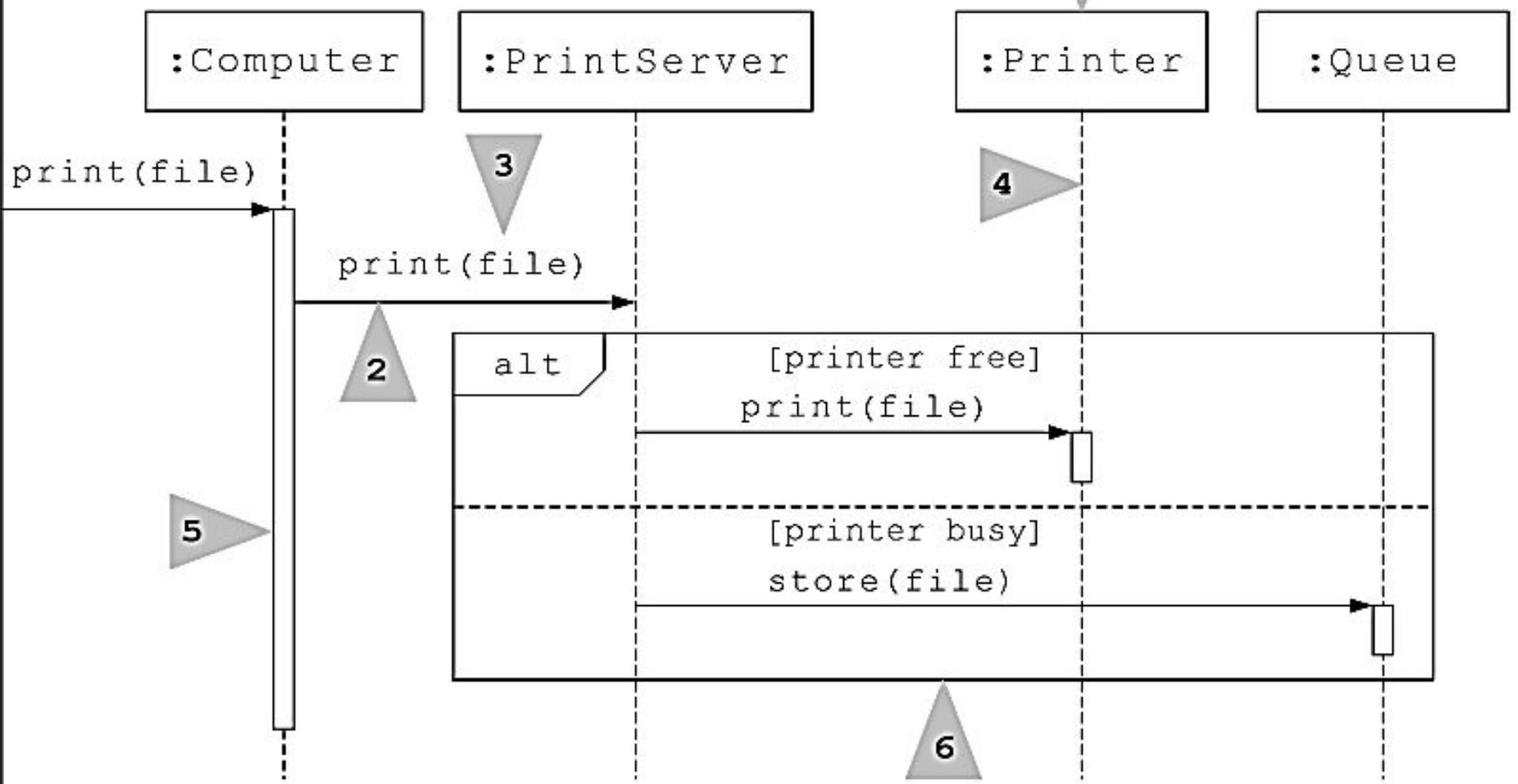


Рис. 1.15. Нотация диаграммы последовательности

# Диаграмма коммуникации

Диаграмма коммуникации (communication diagram) — способ описания поведения, семантически эквивалентный диаграмме последовательности.

Фактически, это такое же описание последовательности обмена сообщениями взаимодействующих экземпляров классификаторов, только выраженное другими графическими средствами. Таким образом, на диаграмме коммуникации также как и на диаграмме последовательности применяют один основной тип сущностей — экземпляры взаимодействующих классификаторов (1) и один тип отношений — связи (2). Однако здесь акцент делается не на времени, а на структуре связей между конкретными экземплярами.

**comm** Печать через сервер печати

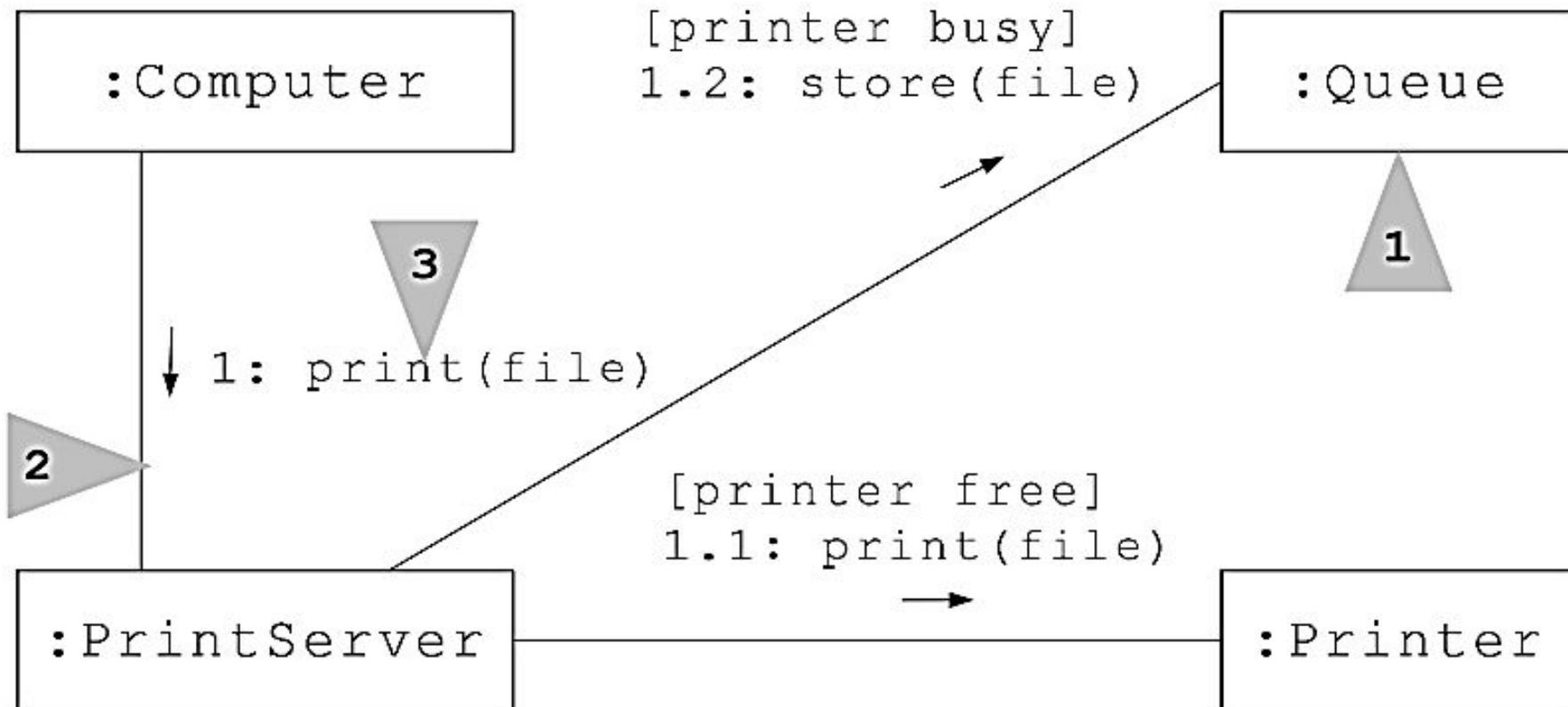


Рис. 1.16. Нотация диаграммы коммуникации

# Диаграмма КОМПОНЕНТОВ

Диаграмма компонентов (component diagram) — показывает взаимосвязи между модулями (логическими или физическими), из которых состоит моделируемая система.

Основной тип сущностей на диаграмме компонентов — это сами компоненты (1), а также интерфейсы (2), посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения:

- реализации между компонентами и интерфейсами (компонент реализует интерфейс);
- зависимости между компонентами и интерфейсами (компонент использует интерфейс) (3).

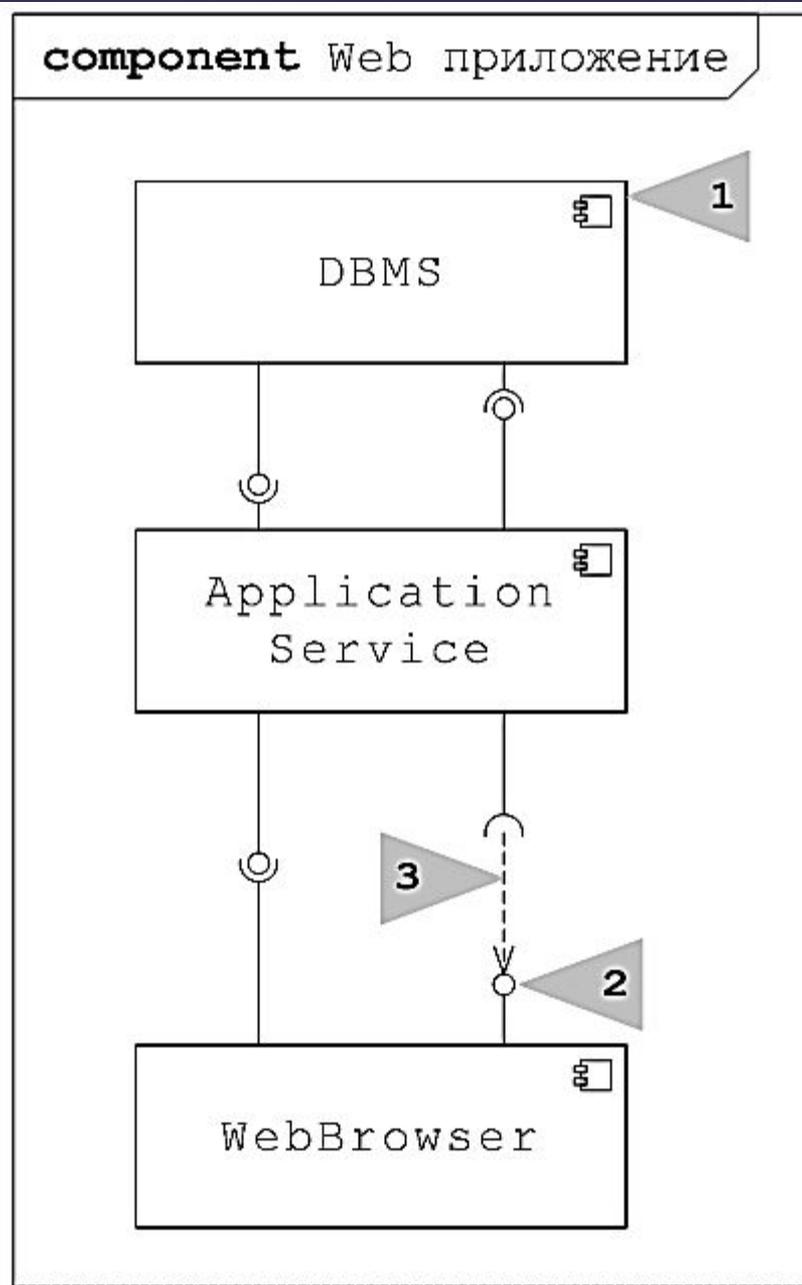
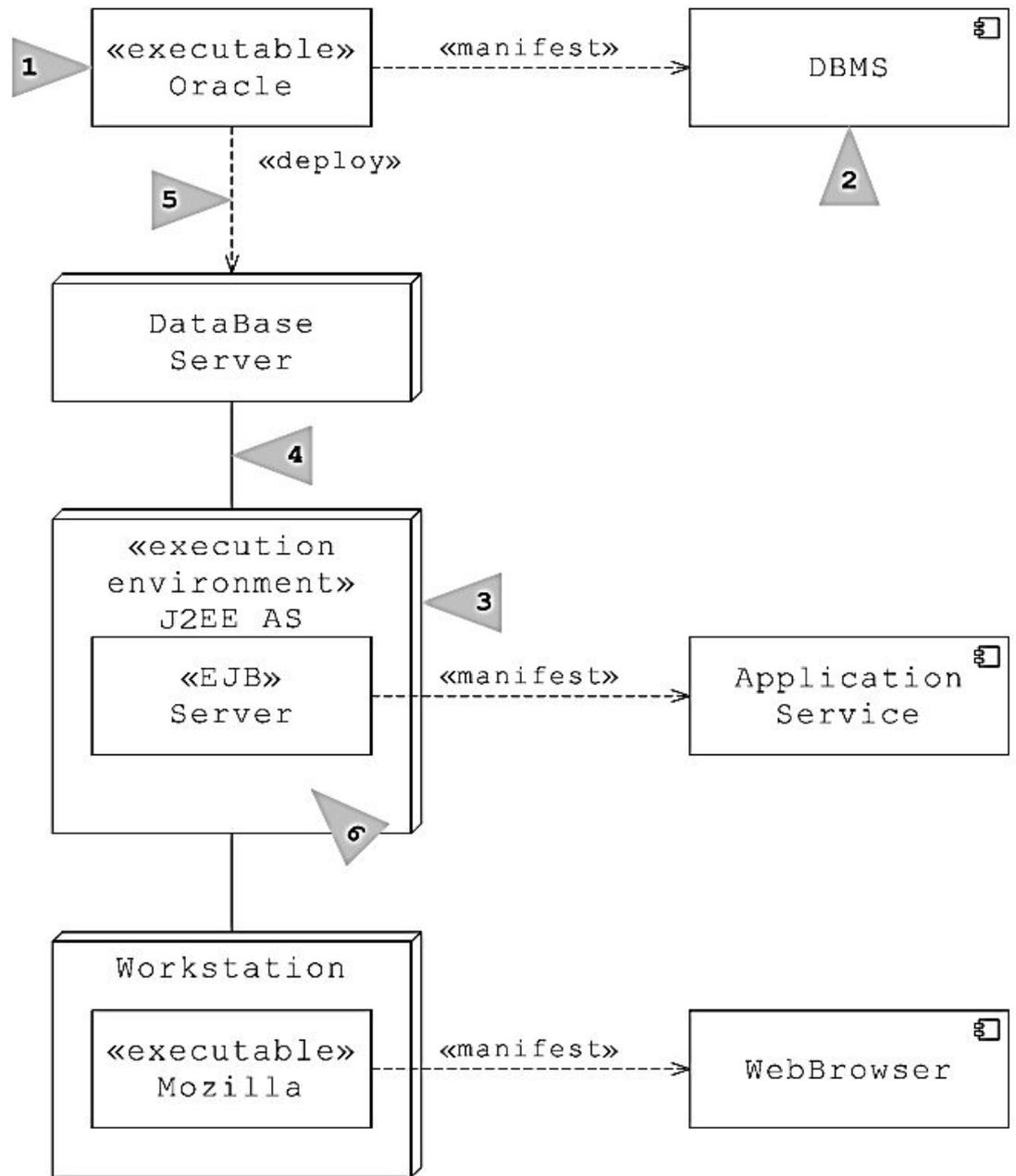


Рис. 1.17. Нотация диаграммы компонентов

# Диаграмма размещения

Диаграмма размещения (deployment diagram) наряду с отображением состава и связей элементов системы показывает, как они физически размещены на вычислительных ресурсах во время выполнения.



# СПЕЦИАЛЬНЫЕ ДИАГРАММЫ

## Диаграмма объектов

Диаграмма объектов (object diagram) — является экземпляром диаграммы классов.

На диаграмме объектов применяют один основной тип сущностей: объекты (1) (экземпляры классов), между которыми указываются конкретные связи (2) (чаще всего экземпляры ассоциаций).

Диаграммы объектов имеют вспомогательный характер — по сути это примеры (можно сказать, дампы памяти), показывающие, какие имеются объекты и связи между ними в некоторый конкретный момент функционирования системы.

**object** Структура системы обработки заказов

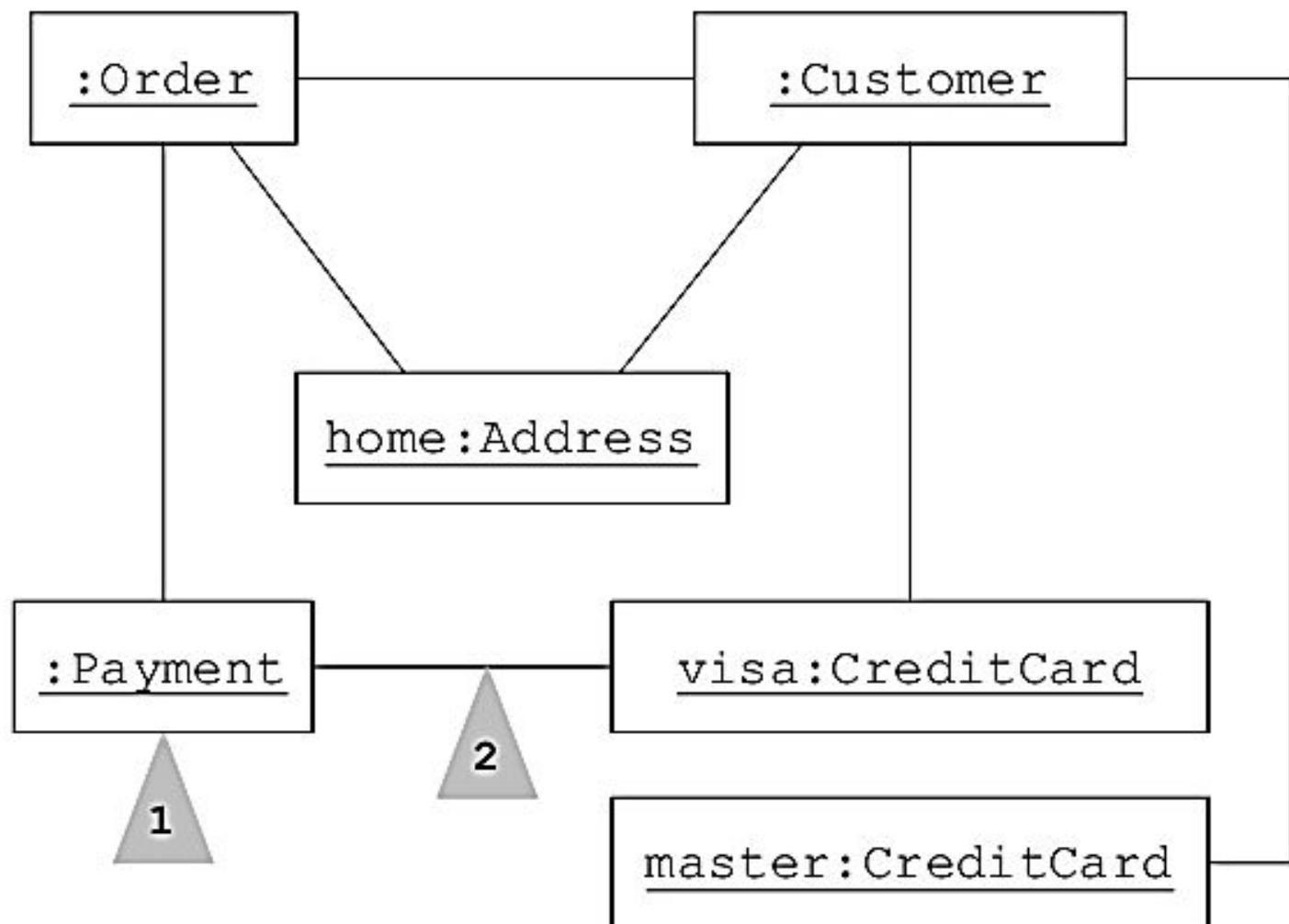


Рис. 1.19. Нотация диаграммы объектов

# Диаграммы вариантов

## ИСПОЛЬЗОВАНИЯ

**Диаграммы вариантов использования** показывают **взаимодействия** между **вариантами использования** и **действующими лицами**, отражая функциональные требования к системе с точки зрения пользователя.

Цель построения – **документирование функциональных требований** в общем виде (требование – простота).

**Вариант использования** – последовательность **действий** (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом).

**Вариант использования** описывает **типичное взаимодействие** между **пользователем и системой** и отражает представление о **поведении системы** с точки зрения пользователя.

# Достоинства модели вариантов использования:

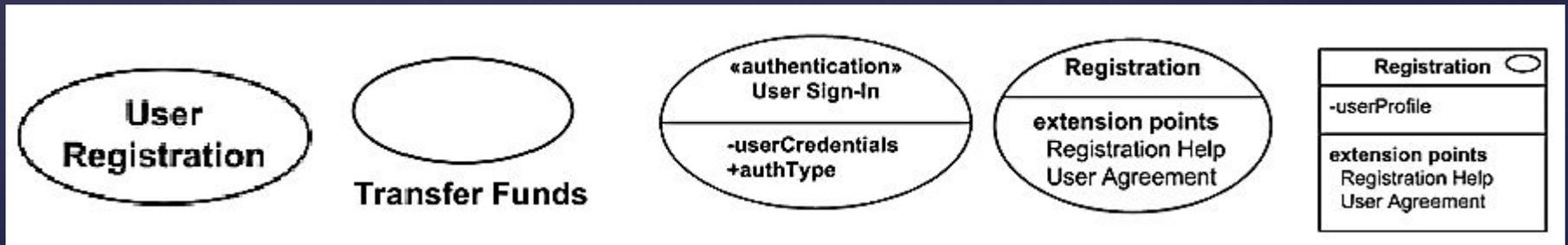
- ▣ определяет пользователей и границы системы;
- ▣ определяет системный интерфейс;
- ▣ удобна для общения пользователей с разработчиками;
- ▣ используется для написания тестов;
- ▣ является основой для написания пользовательской документации;
- ▣ хорошо вписывается в любые методы проектирования (как объектно-ориентированные, так и структурные).

# Вариант использования

Обозначается на диаграмме **эллипсом**, внутри - название.

Цель варианта использования - **определить законченный аспект** или фрагмент **поведения некоторой сущности** без раскрытия её внутренней структуры.

Сущность: система или любой элемент модели, который обладает собственным поведением.



# Актеры

**Актер** - **внешняя** по отношению к моделируемой системе **сущность**, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей.

Каждый актер - некая отдельная роль относительно конкретного варианта использования.

Стандартное графическое обозначение - **фигурка человечка**, под ней - имя.

**Актеры взаимодействуют** с системой посредством **обмена сообщениями** с вариантами использования.

**Сообщение** - запрос актером определенного **сервиса системы** и **получение этого сервиса**. Это взаимодействие может быть выражено посредством ассоциаций между отдельными актерами и вариантами использования или классами.



**Web Client**

«actor»  
**Customer**

+ name: Name  
+ address: Address



**Bank**

# Отношения

Между элементами диаграммы вариантов использования могут существовать различные **отношения**, которые описывают взаимодействие экземпляров актеров и вариантов использования.

Стандартные виды отношений между актерами и вариантами использования:

- **ассоциации** (association relationship);
- **расширения** (extend relationship);
- **обобщения** (generalization relationship);
- **включения** (include relationship).

# Отношение ассоциации

**Ассоциация** устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

Обозначение - **сплошная линия** между актером и вариантом использования.

Линия может иметь условные обозначения:

- имя;
- кратность.

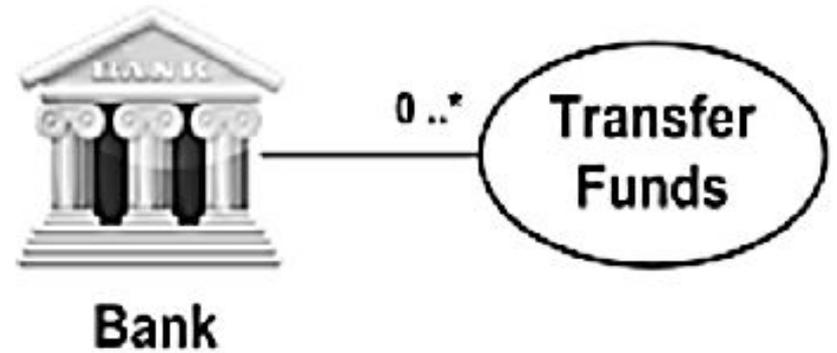
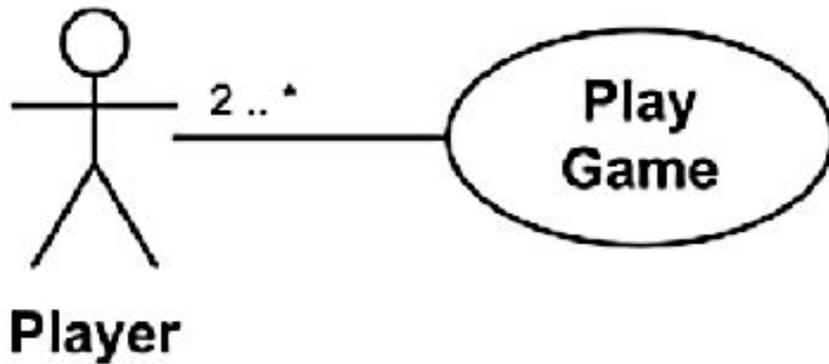
**Кратность (multiplicity)** указывается рядом с обозначением компонента диаграммы, который является участником данной ассоциации, и характеризует **количество экземпляров данного компонента**, которые могут выступать в качестве **элементов данной ассоциации**.

Обозначение - в форме одной или нескольких **цифр** и символа **звездочка**.

# Распространенные формы записи кратности отношения ассоциации:

- ▣ **целое неотрицательное число (включая 0).** Указание кратности, которая является строго фиксированной для элемента соответствующей ассоциации. Количество экземпляров актеров или вариантов использования равно указанному числу;
- ▣ **два целых неотрицательных числа, разделенные двумя точками.** Соответствует нотации для множества или интервала целых чисел, которая применяется в некоторых языках программирования для обозначения границ массива элементов. Множество целых неотрицательных чисел, следующих в последовательно возрастающем порядке;
- ▣ **два символа, разделенные двумя точками.** Первый из них - целое неотрицательное числом или 0, второе - специальный символом «\*», обозначает произвольное конечное целое неотрицательное число, значение которого неизвестно на момент задания соответствующего отношения ассоциации;
- ▣ **СИМВОЛ «\*»** - сокращение записи интервала «0..\*». Кратность отношения ассоциации по умолчанию - 1.

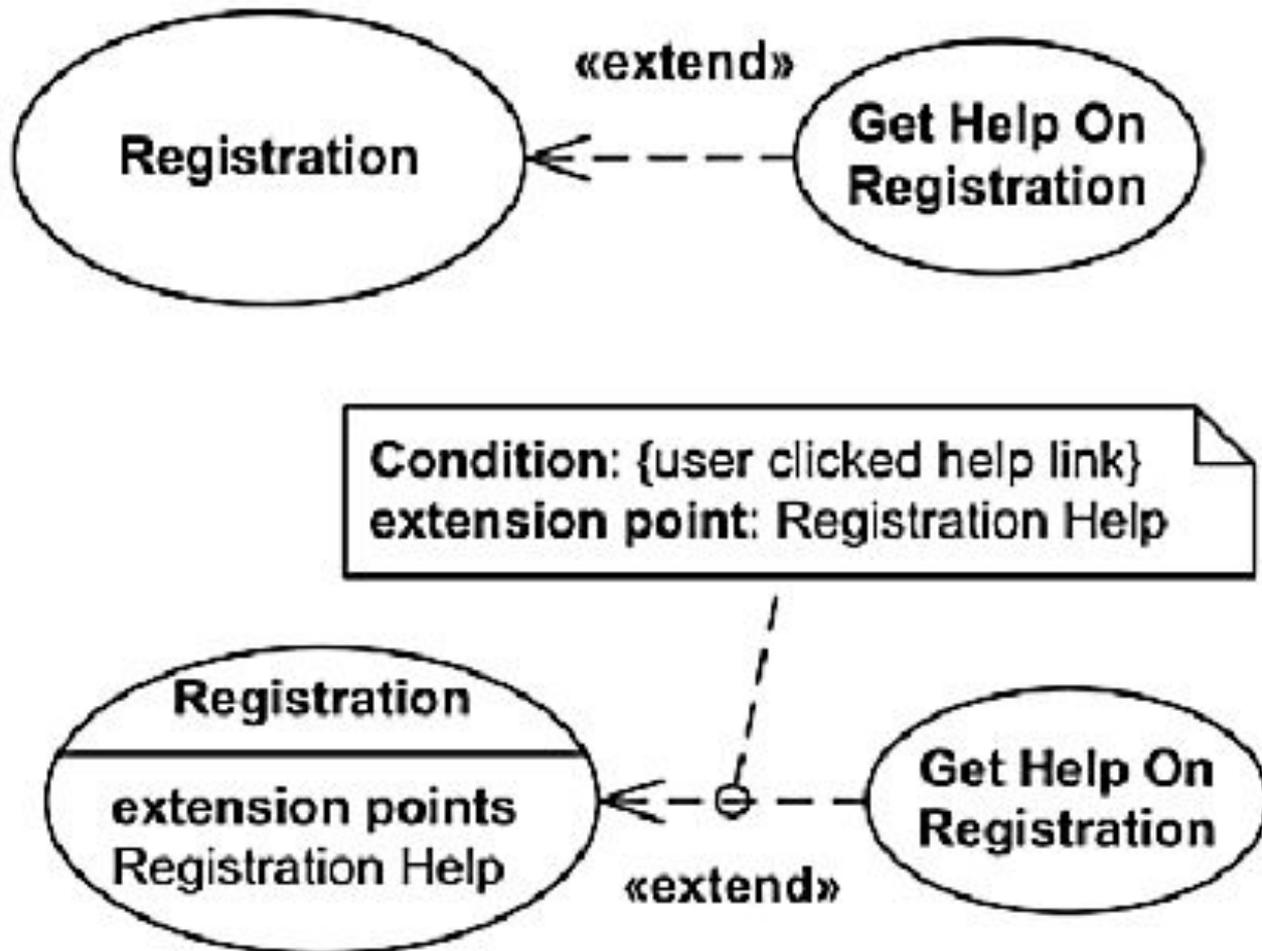
# Отношение ассоциации



# Отношение расширения

- ▣ **Отношение расширения** - определяет **взаимосвязь экземпляров** отдельного варианта использования с более **общим вариантом**, свойства которого определяются на основе способа совместного объединения данных экземпляров.
- ▣ Обозначение - **пунктирная линией со стрелкой** (вариант отношения зависимости), направленной от того варианта использования, который является **расширением** для исходного варианта использования. Линия со стрелкой помечается ключевым словом «extend» (расширяет).
- ▣ **Один вариант** использования может быть **расширением для нескольких базовых вариантов**, а также иметь в качестве собственных расширений несколько других вариантов.
- ▣ Базовый вариант использования может дополнительно никак не зависеть от своих расширений.

# Отношения. Отношение расширения



# Отношение обобщения

Отношение **обобщения** служит для указания того факта, что некоторый вариант использования **A** может быть обобщен до варианта использования **B**.

В этом случае вариант **A** - специализацией варианта **B**.

**B** - предок или родитель по отношению **A**, а вариант **A** – потомок по отношению к варианту использования **B**.

**Потомок наследует все свойства и поведение своего родителя**, может быть дополнен **новыми свойствами и особенностями поведения**.

Графическое обозначение - **сплошная линия со стрелкой в форме незакрашенного треугольника**, которая указывает на родительский вариант использования.

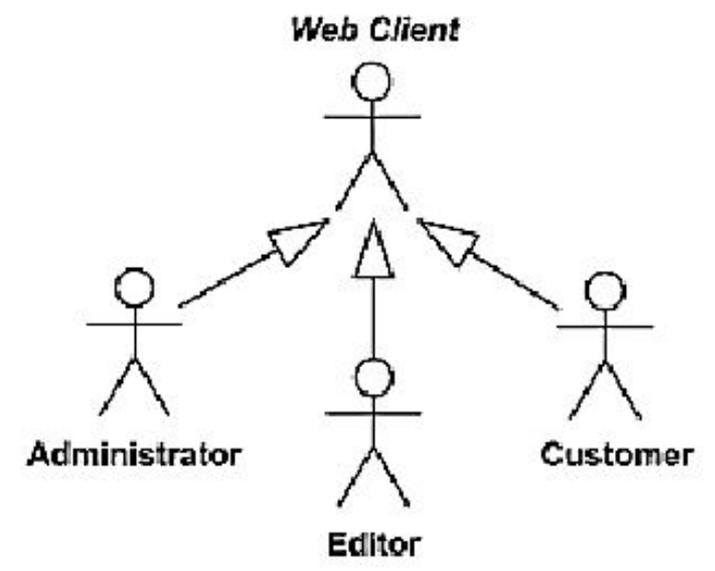
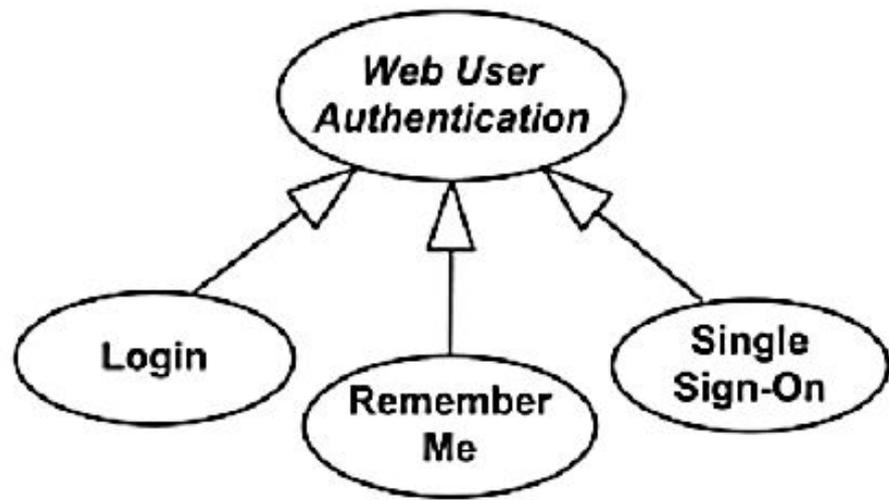
Один вариант использования может иметь несколько родительских вариантов. В этом случае реализуется **множественное наследование свойств и поведения отношения предков.**

Один вариант использования может быть предком для нескольких дочерних вариантов, что соответствует **таксономическому характеру отношения обобщения.**

**Между отдельными актерами** также может существовать отношение **обобщения.**

Данное отношение - направленное и указывает на факт специализации одних актеров относительно других.

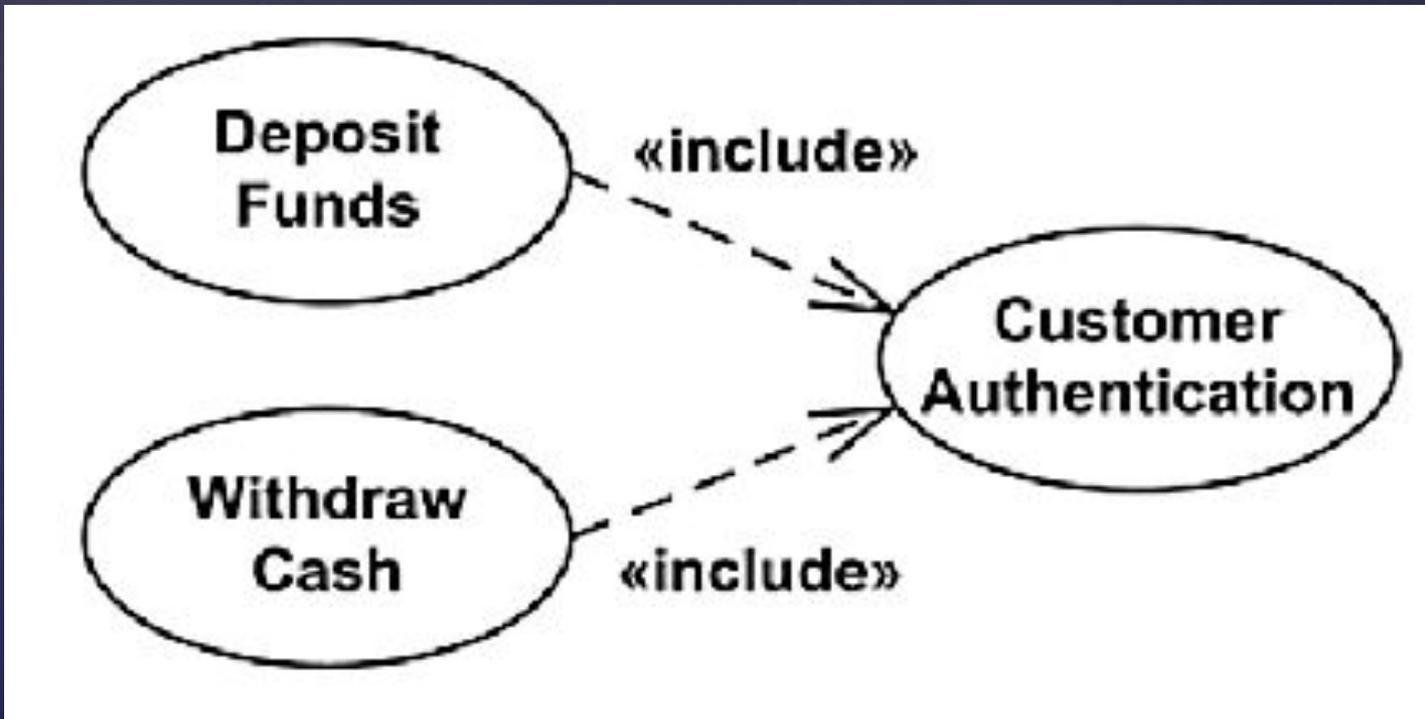
## Отношения. Отношение обобщения



# Отношение включения

- Отношение **включения** между двумя вариантами использования указывает, что некоторое **заданное поведение** для одного варианта использования **включается в качестве составного компонента в последовательность поведения другого варианта использования.**
- Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты.
- **Включаемый вариант** использования может быть **независимым от базового варианта** в том смысле, что он предоставляет ему некоторое **инкапсулированное поведение**, детали реализации которого скрыты и могут быть перераспределены между несколькими включаемыми вариантами использования.
- **Базовый вариант** может зависеть только от **результатов выполнения включаемого** в него поведения, но не от структуры включаемых в него вариантов.

- Отношение включения, направленное от варианта использования **A** к варианту использования **B**, указывает, что каждый экземпляр варианта **A** включает в себя **функциональные свойства**, заданные для варианта **B**.
- Графическое обозначение - **пунктирная линия со стрелкой**, которая помечается ключевым словом «include» (включает).



# Примеры построения диаграмм вариантов использования

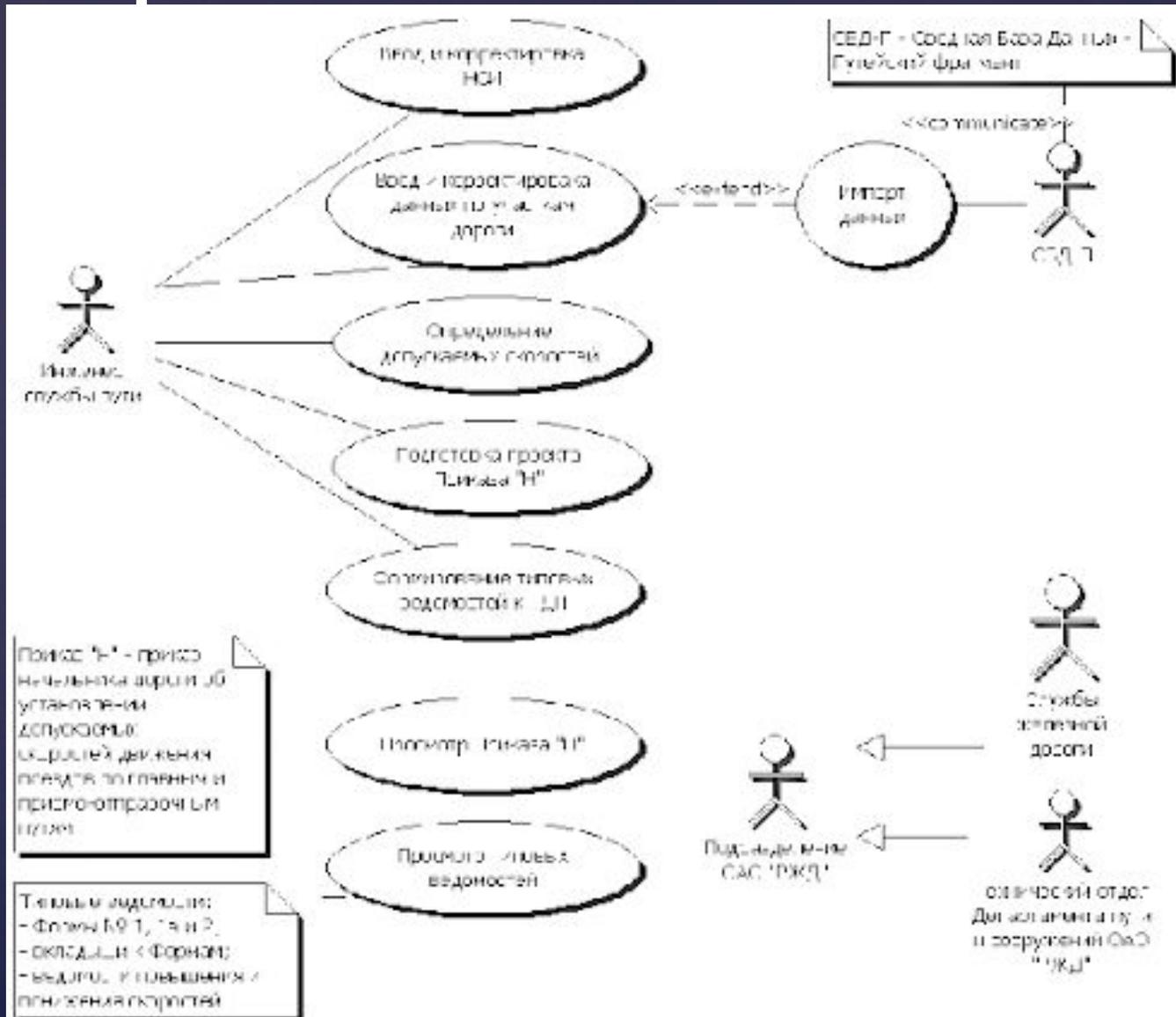


Рис. 12.9. Контекстная диаграмма вариантов использования системы определения допусков для скоростей

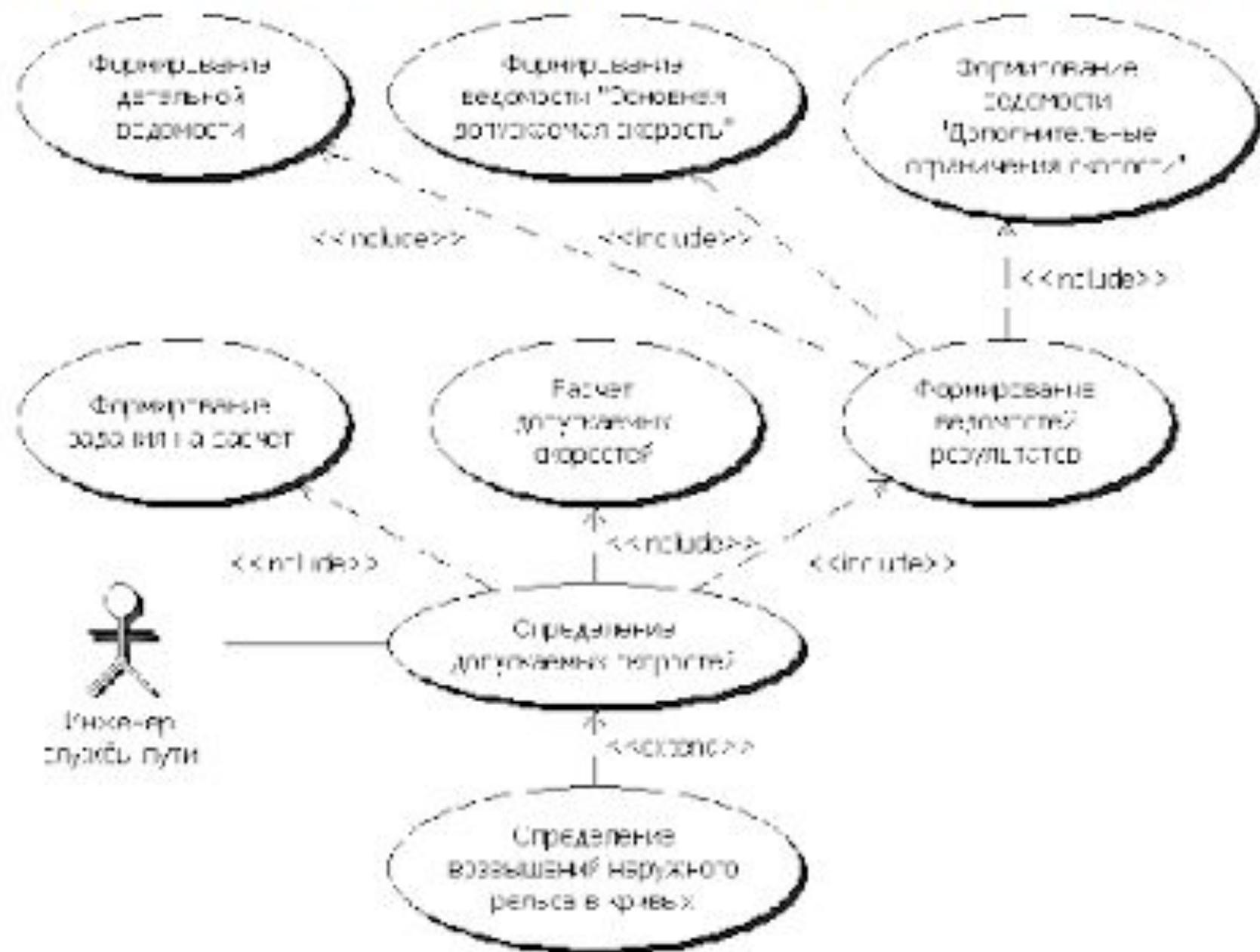


Рис. 12.10. Диаграмма декомпозиции варианта использования «Определение допустимых скоростей»

# Создание диаграммы

