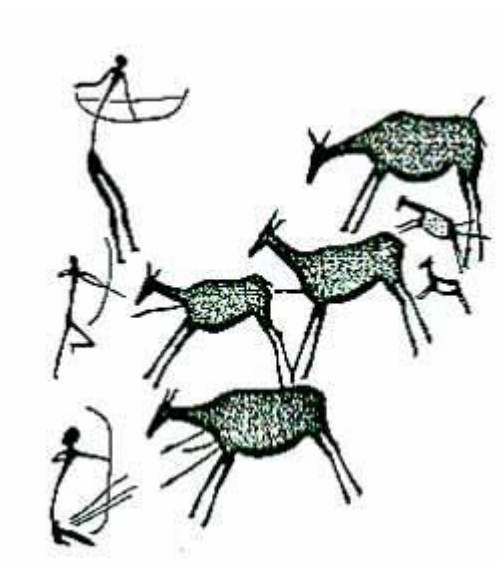


UML и моделирование

Введение



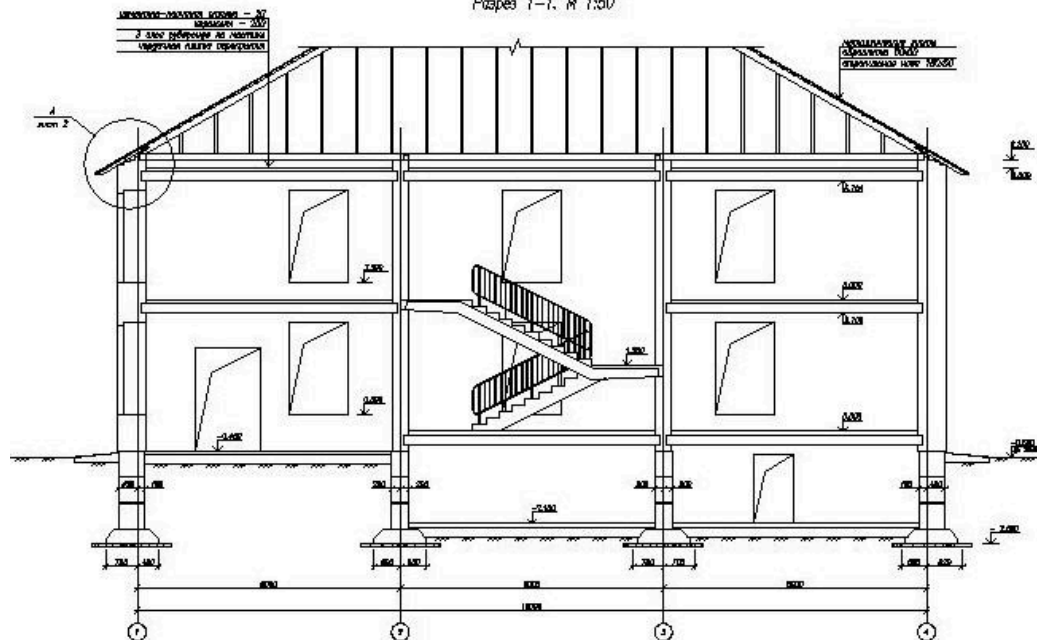
Шахматная нотация

1. e2e4 e7e5
2. f1c4 b8c6
3. d1h5 g8f6??
4. h5xf7#

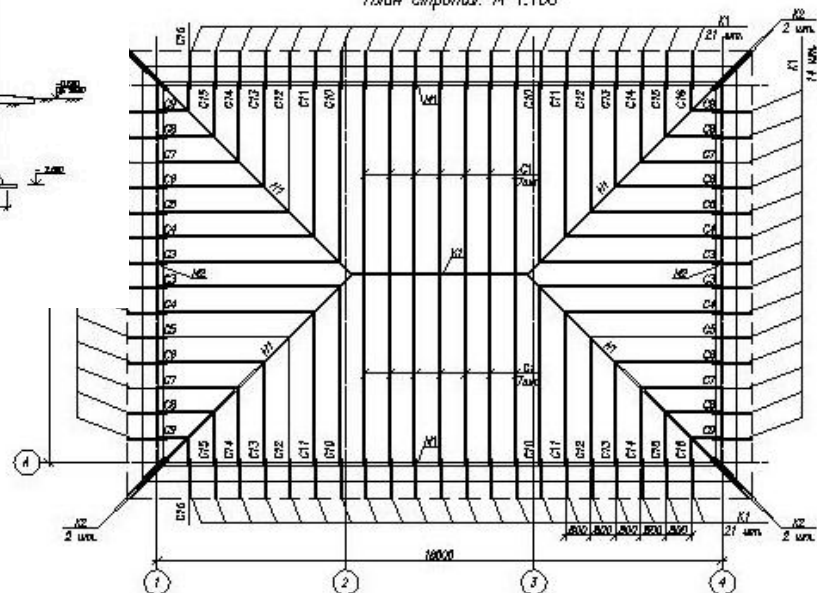


Схемы

Разрез 1-1. М 1:50



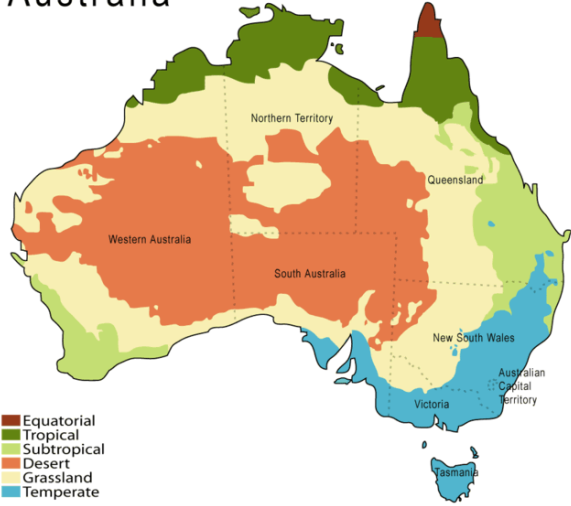
План стропил. М 1:100



Дополняющие модели



Australia



Возможные определения

- **Модель — наиболее абстрактная конструкция, которая выделяет существенную структуру явлений и процессов при отвлечении от несущественных деталей (www.wikipedia.org)**
- **A model is a simplification. It is an interpretation of reality that abstracts the aspects relevant to solving the problem at hand and ignores extraneous detail (Eric Evans)**

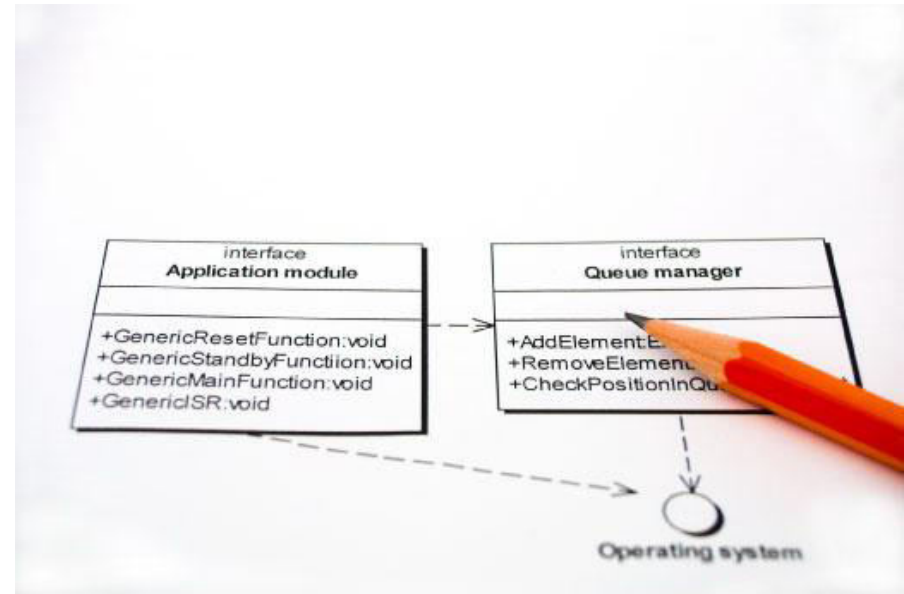
UML

- **UML: Unified Modeling Language**
 - *Унифицированный*
 - *Язык*
 - *Синтаксис + семантика*
 - *Моделирования*
 - *Modelling vs Simulation*

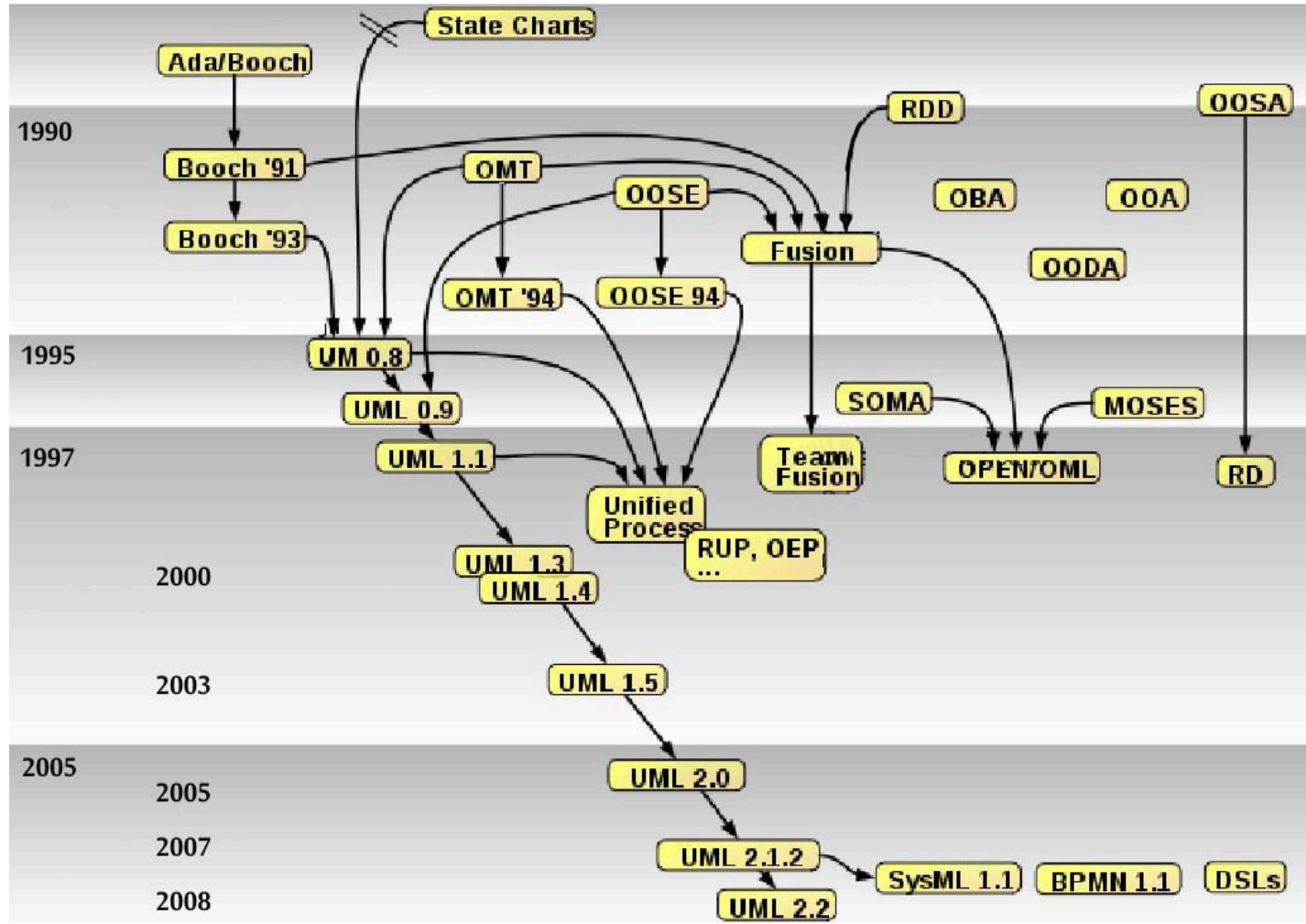


UML

- **Формальный**
- **Искусственный**
- **Обладает графической нотацией**
- **Является стандартом Object Management Group**



Немного об истории



Стандарты

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)



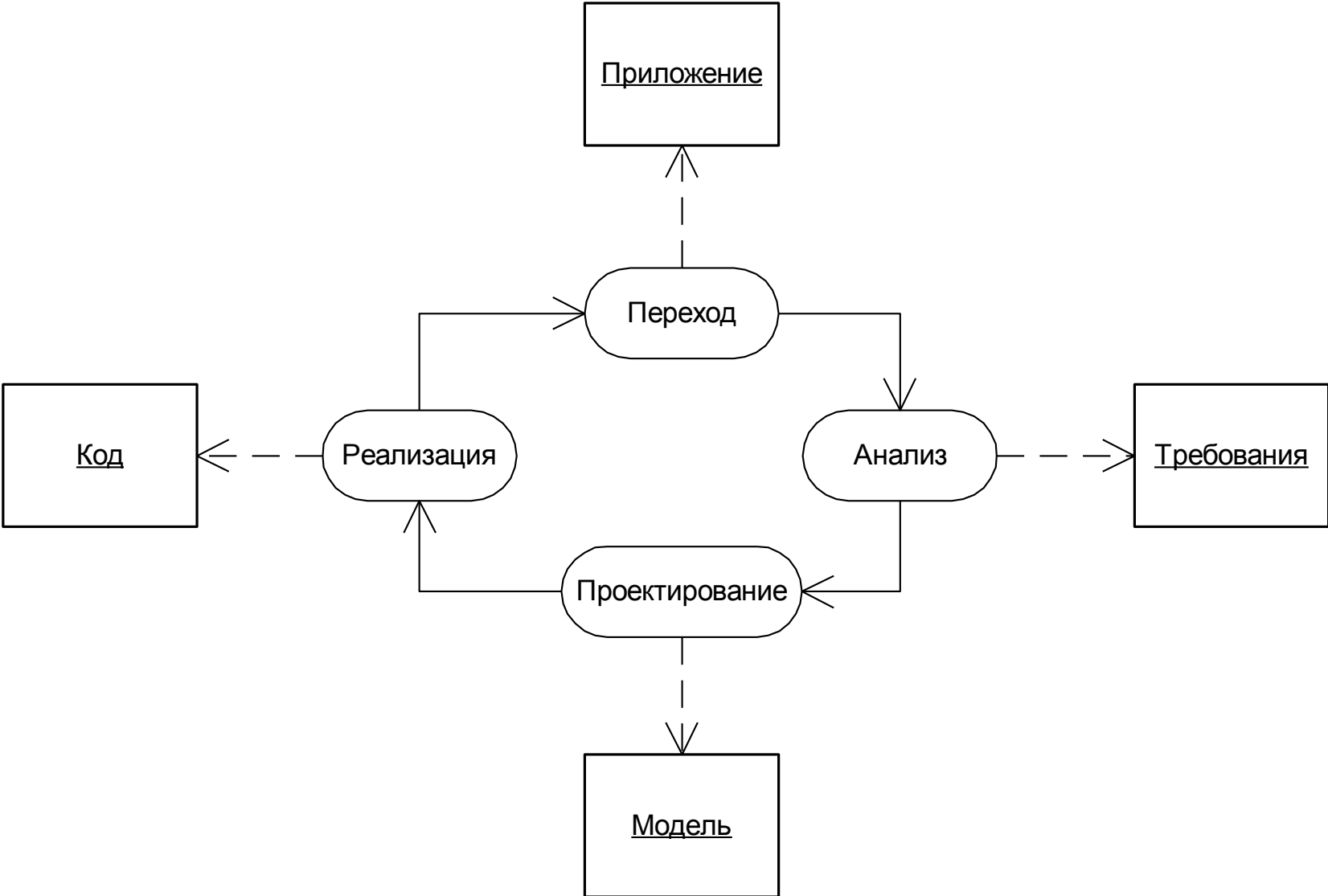
Авторы UML



Языки, повлиявшие на UML

- Блок-схема
- ER – диаграмма
- Диаграммы состояний из теории автоматов
- ...

Процесс разработки



Назначение UML

- **Рисование эскизов !!!**
 - *Удобный способ обмена идеями*
- **Проектирование и документирование !**
 - *Компактная и емкая документация (если нужно)*
- **Артефакт ??**
 - *Executable UML*

Чем не является UML

- Языком программирования
- Процессом разработки

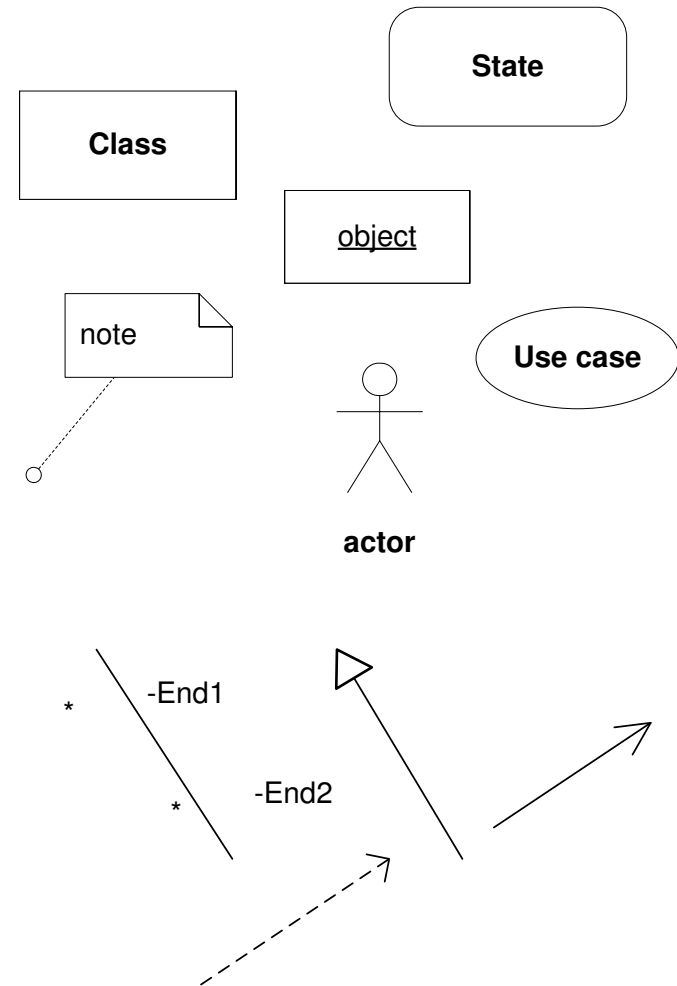
Базовые элементы

■ Сущности

- *классы, объекты, интерфейсы, действующие лица, состояния, операции, примечания, ...*

■ Отношения

- *Ассоциации, зависимости, обобщения, переходы, ...*



Диаграммы UML

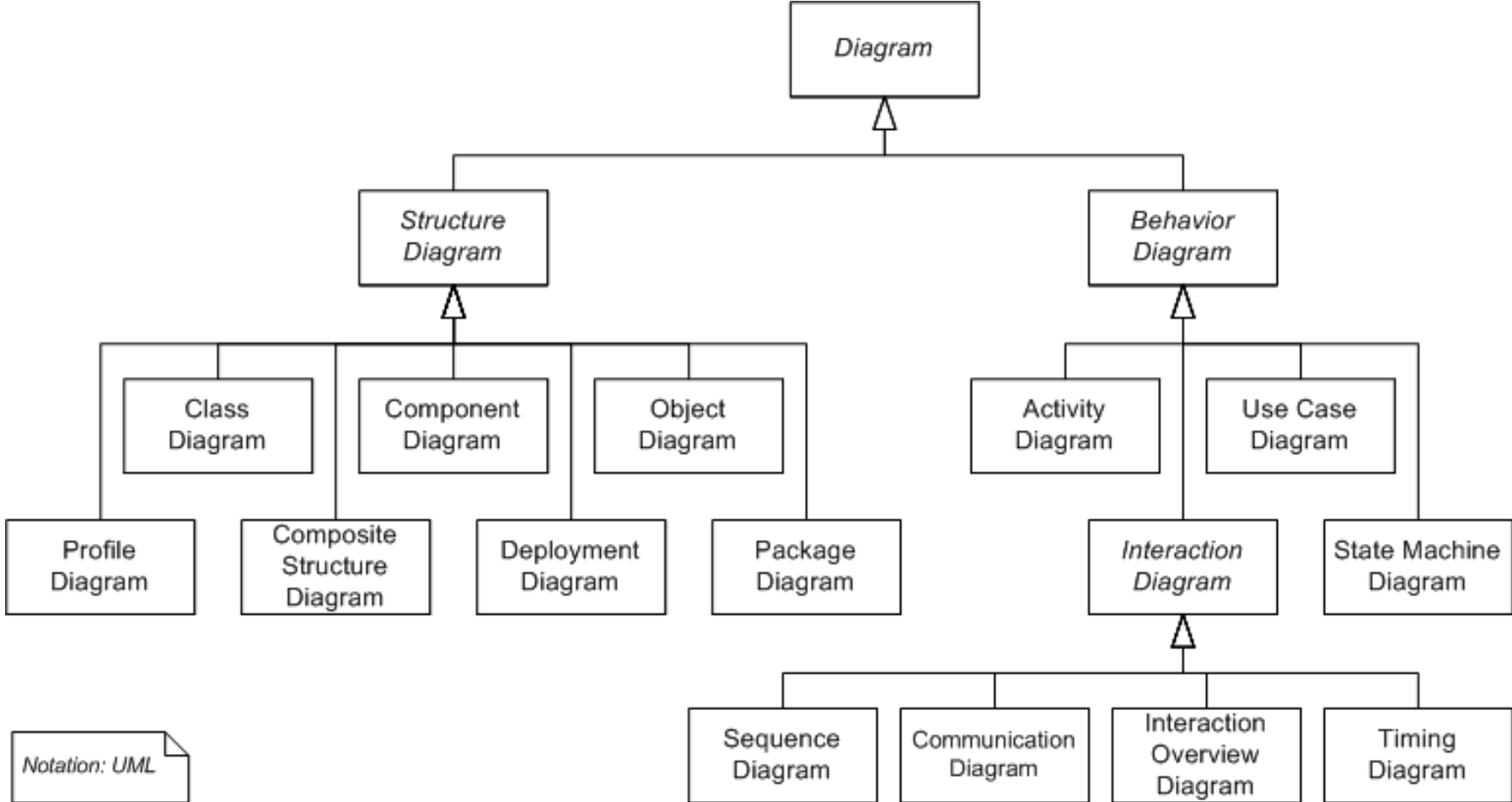
- Структурные

- *Диаграмма классов (class)*
- *Диаграмма объектов (object)*
- *Диаграмма компонентов (component)*
- *Диаграмма развертывания (deployment)*

- Поведенческие

- *Диаграмма использования (use case)*
- *Диаграмма деятельности (activity)*
- *Диаграмма состояний (statechart)*
- *Диаграмма последовательности (sequence)*
- *Диаграмма взаимодействия (communication)*

Диаграммы UML



Диаграммы использования

- Способ изображения требований к системе
- Основные элементы
 - Система
 - Действующее лицо
 - Вариант использования

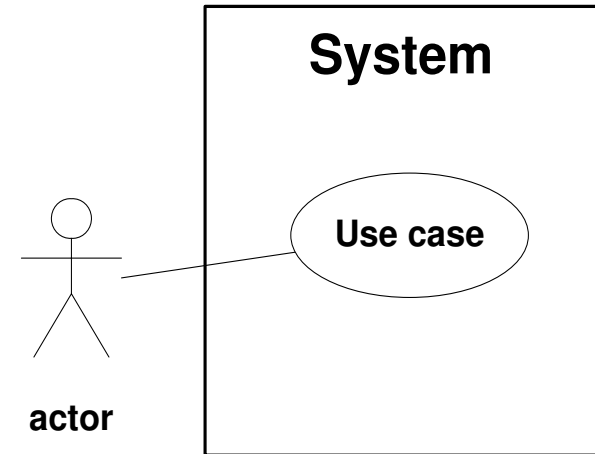
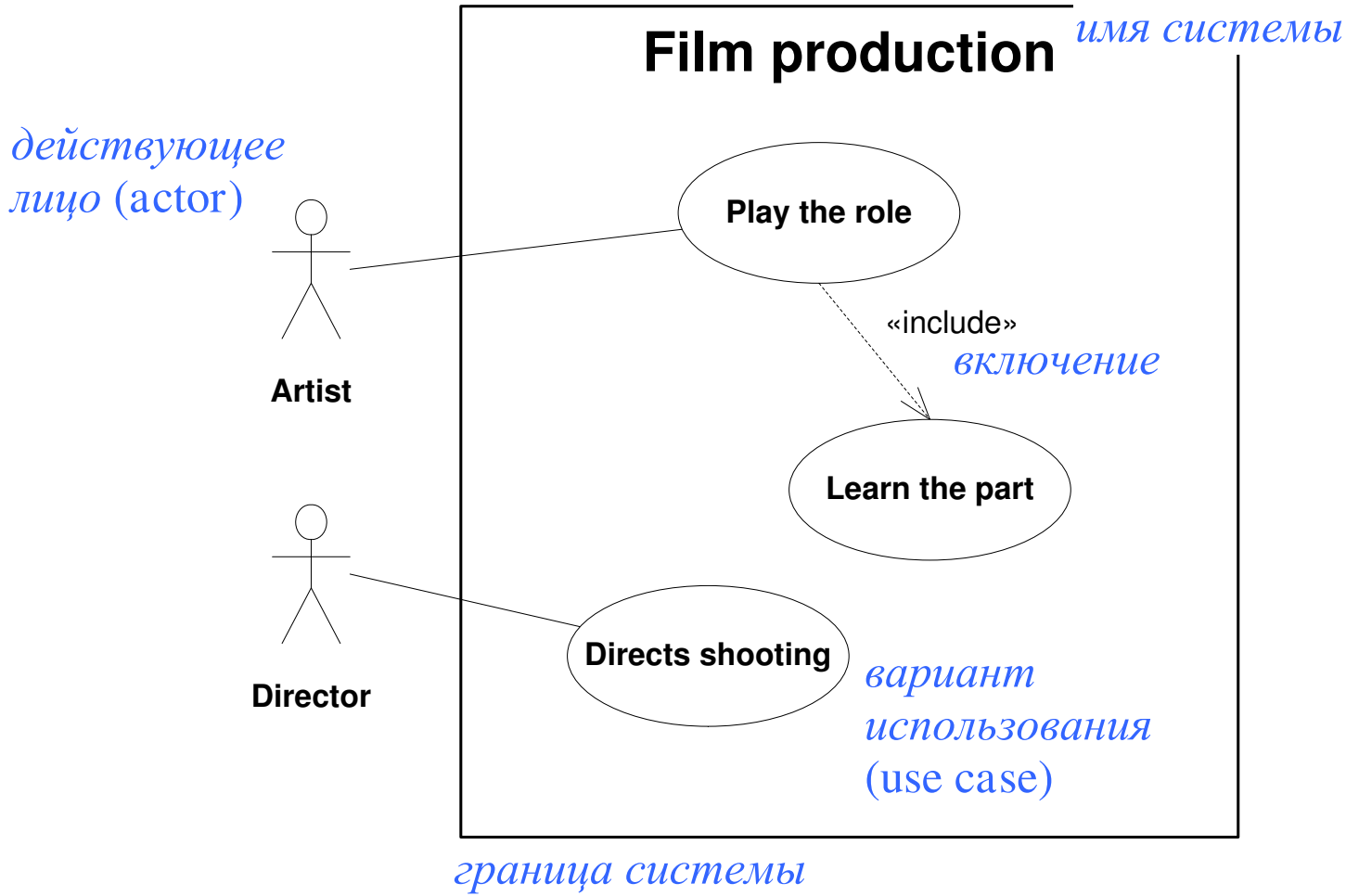


Диаграмма использования



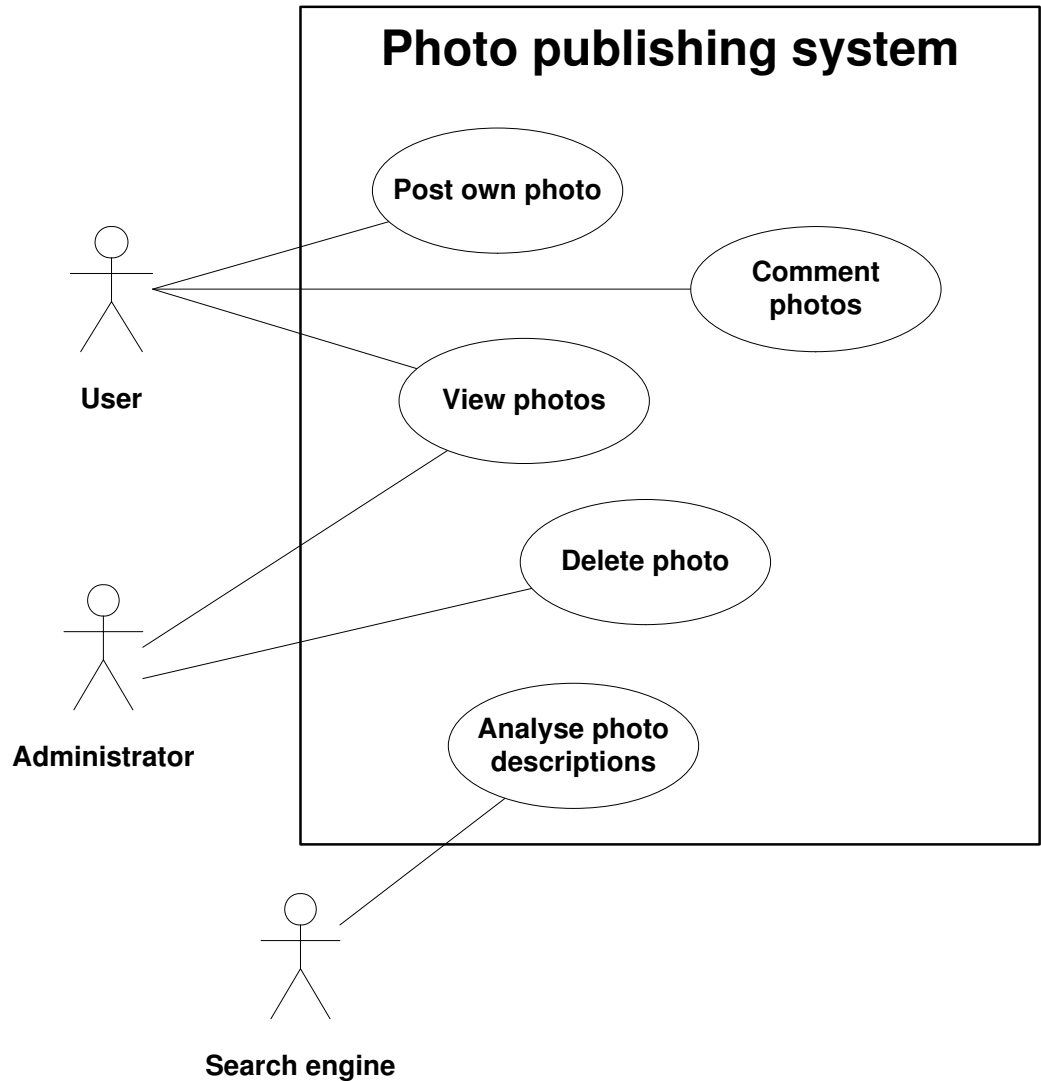
Пример: фотоальбом

- *Пользователь* может добавлять фотографии, просматривать и комментировать чужие
- *Администратор* может просматривать и удалять любые фотографии

Система и действующие лица

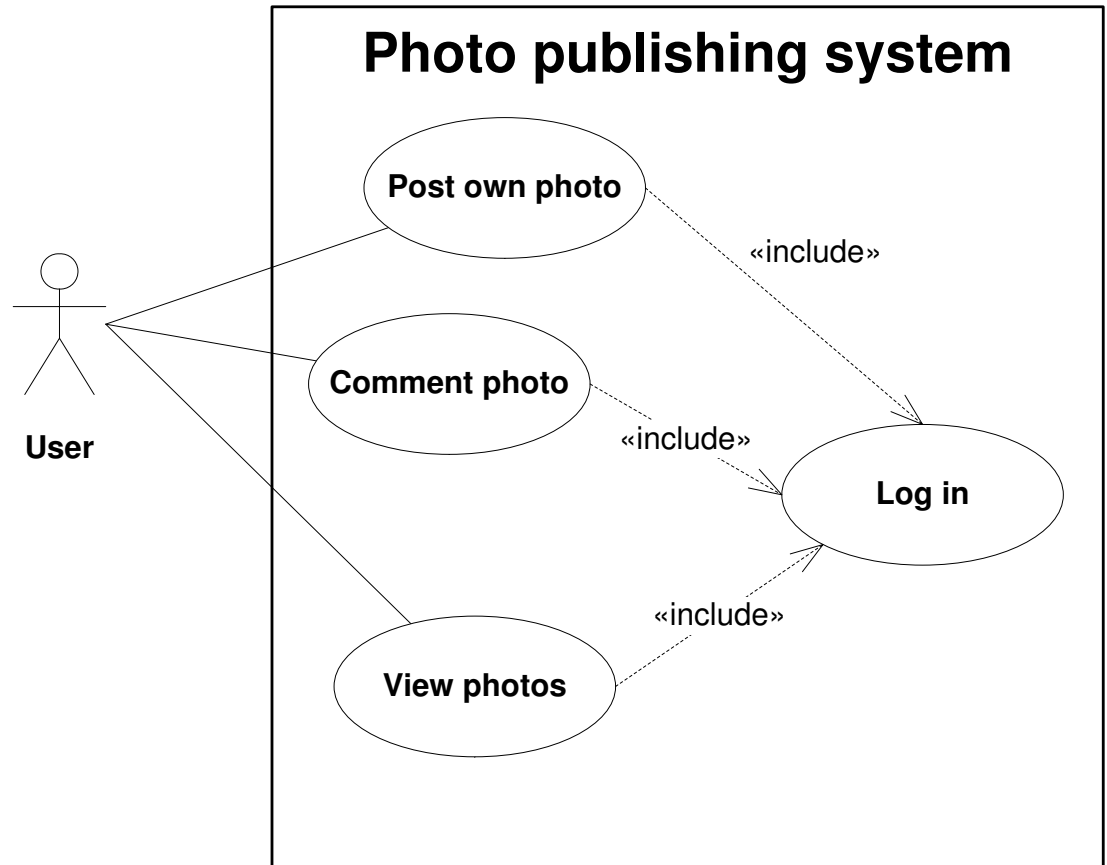
■ Действующие лица

- Категории, не конкретные персонажи
- не обязательно люди
- Находятся вне системы



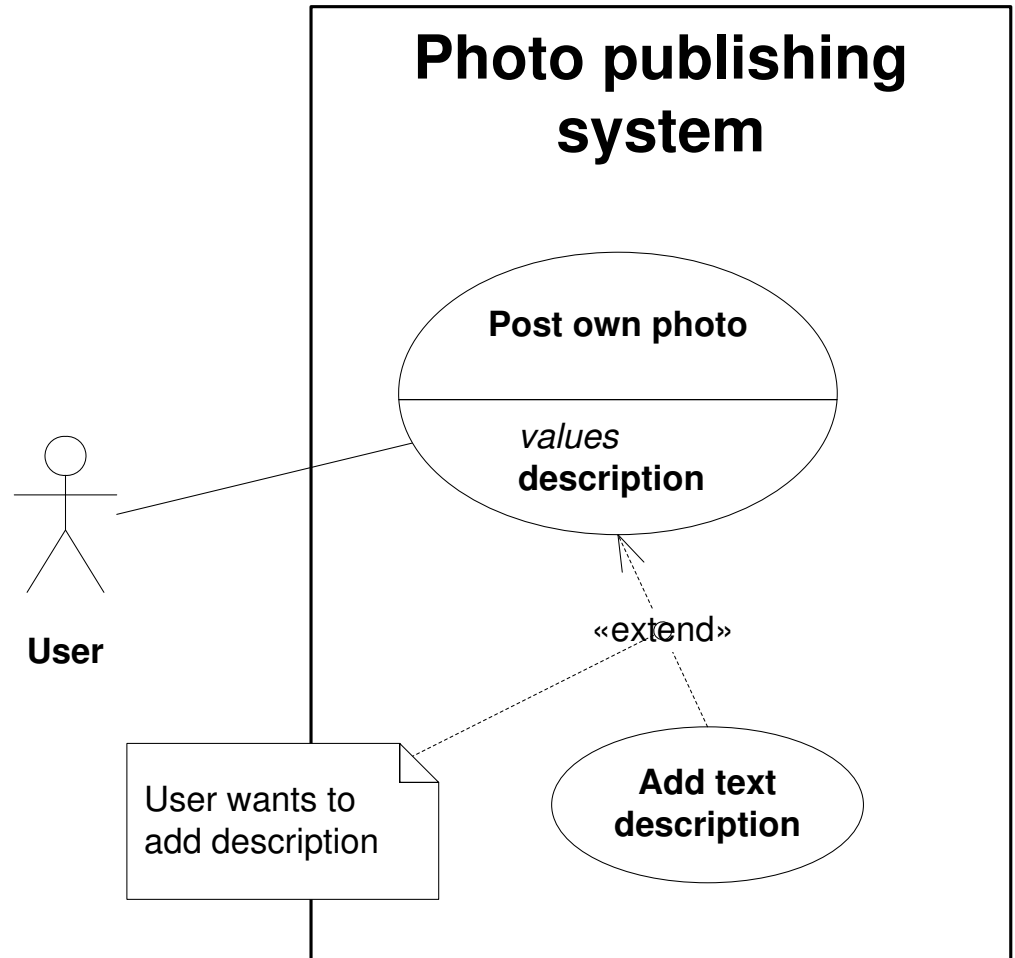
Включение (include)

- Выполнение любого из действий подразумевает обязательный вход в систему



Расширение (extension)

- Публикуя фотографию, пользователь может добавить к ней описание



Текстовое содержание варианта использования

- **Добавление фотографии**
- **Основной успешный сценарий**
 1. Пользователь входит в систему
 2. Выбирает ссылку «Добавить фотографию»
 3. Указывает файл с фотографией
 4. Вводит название фотографии
 5. Выбирает ссылку «Загрузить»
- **Расширения**
 - 4a. *description*. Пользователь хочет добавить описание
 - .1 Добавляет тестовое описание

Преимущества

- **Простые утверждения**
- **Отделены от реализации**
- **Определяют границы системы**

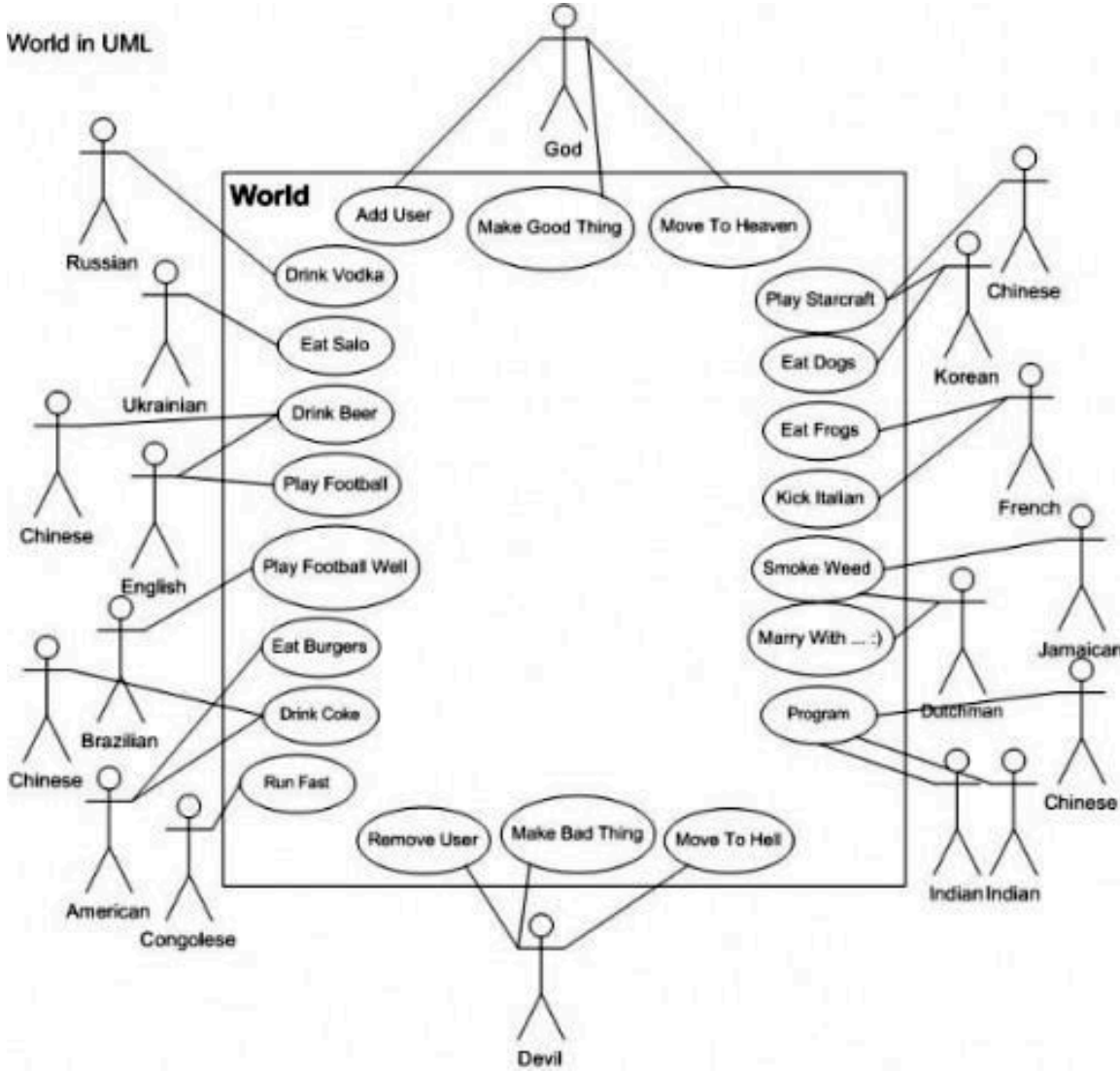
Рекомендации

- Проще начинать с акторов
- Внешние события – хороший источник вариантов использования
- Детализация по мере необходимости
- Не всегда возможно выбрать актора

Выводы

- Помогают идентифицировать пользователей системы и ее основную функциональность
- Каждый вариант использования требует подробных расшифровок

Model of the World

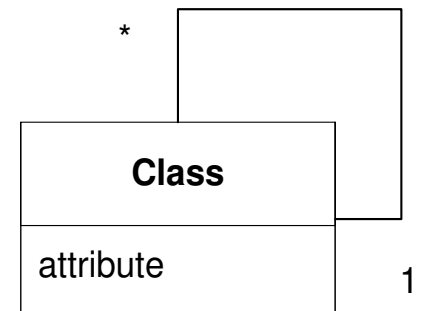


Упражнение

- Нарисуйте диаграмму вариантов использования для банкомата

Диаграммы классов

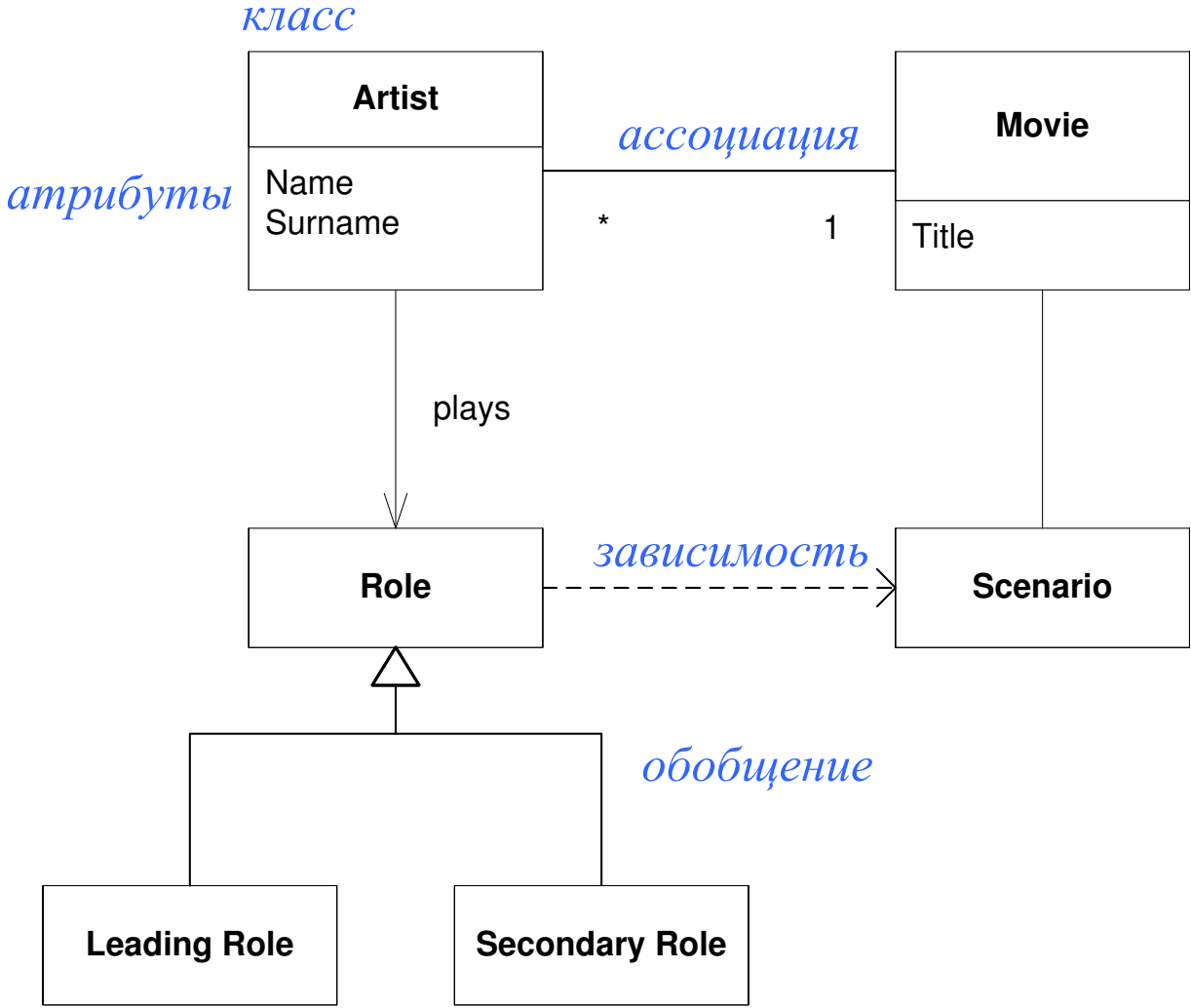
- Описывают сущности (типы объектов) системы и отношения между ними
 - *Моделирование предметной области*
 - *Моделирование программных сущностей*
- Ближайший родственник: ER диаграммы
- Основные элементы
 - *Класс*
 - *Атрибут*
 - *Ассоциация*
 - *Зависимость*



Уровни диаграммы классов

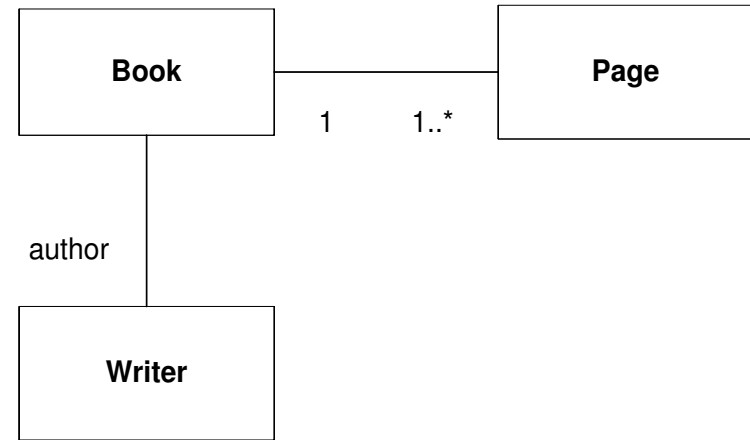
- **Концептуальный**
- **Уровень спецификации**
- **Уровень реализации**

Диаграмма классов

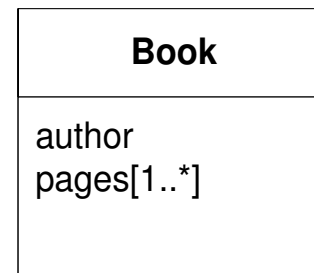


Моделирование свойств

- С помощью ассоциаций

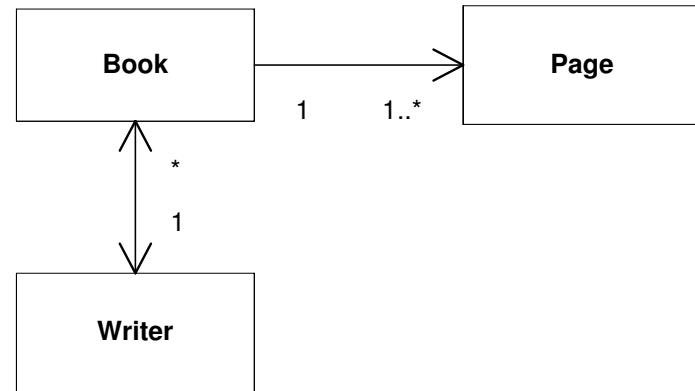


- С помощью атрибутов

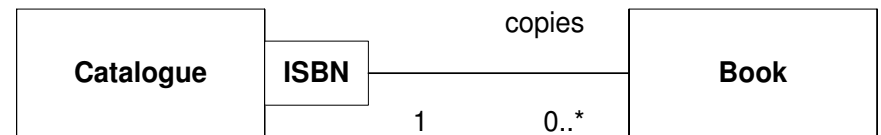


Ассоциации

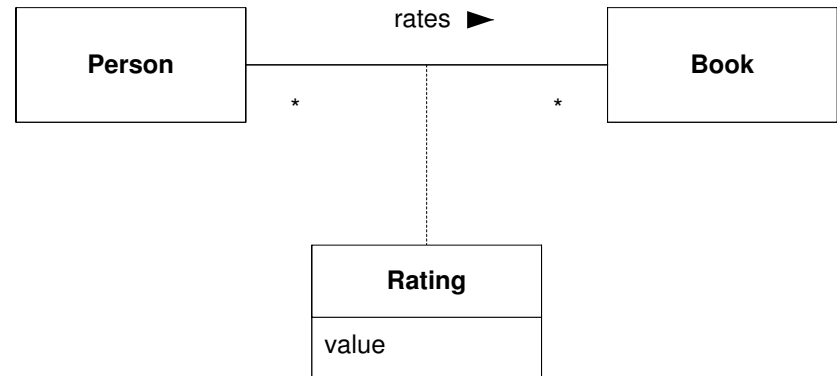
- Направление ассоциации



- Квалифицированная ассоциация

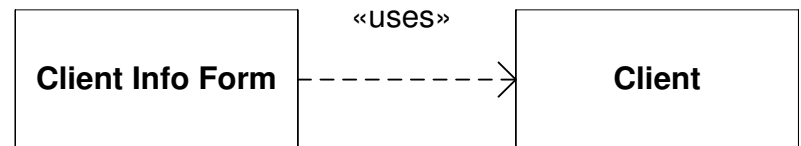
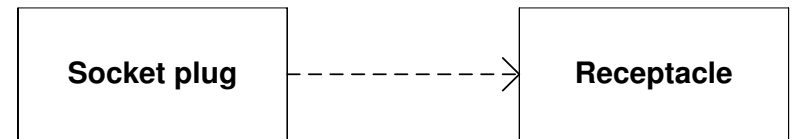
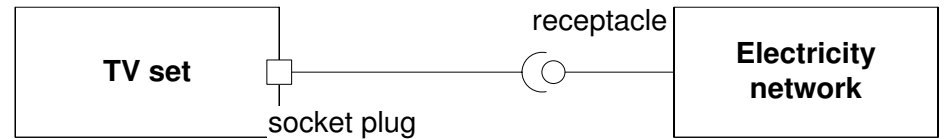


- Класс-ассоциация



Зависимость

- Изменения в определении одного элемента могут вызвать изменения в другом
- Пример: штепсель и розетка
- Пример: форма для ввода данных о клиенте



Моделирование операций

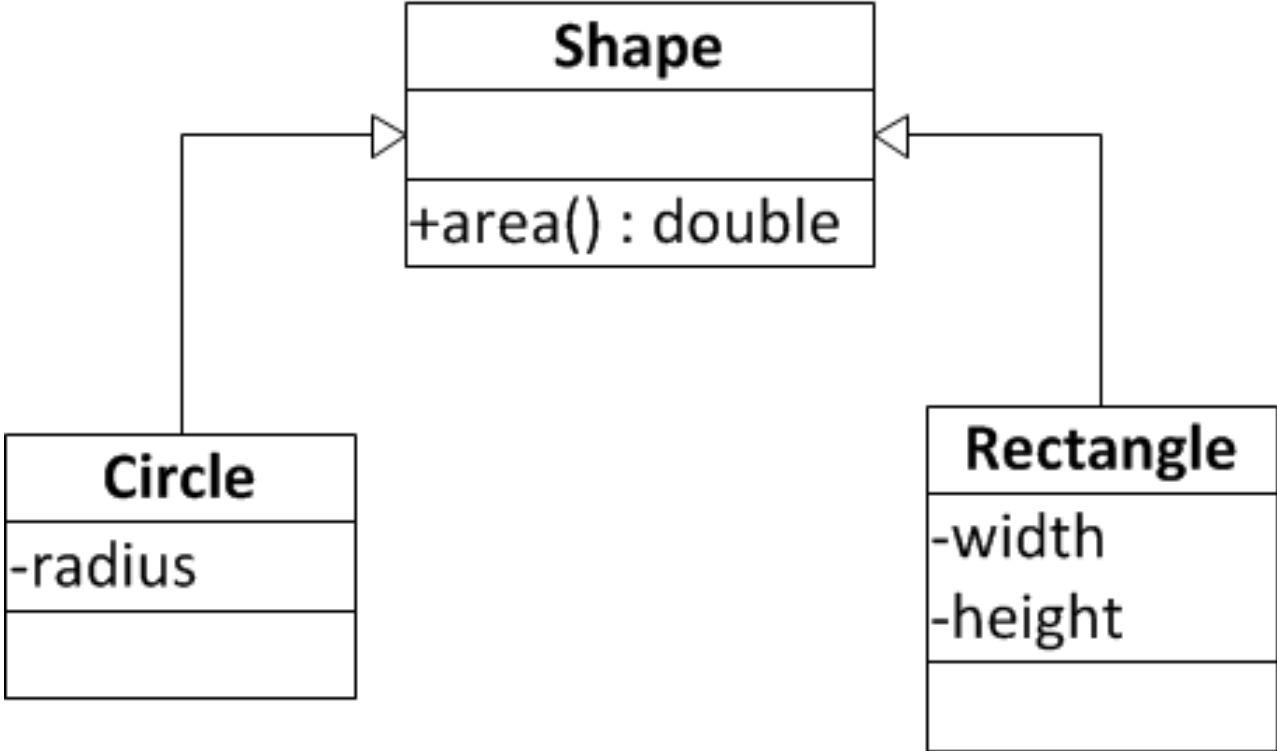
- Соответствуют методам класса
- Полный синтаксис операции:
<видимость><имя>(<список параметров>):
<возвращ. значение>
- Статические функции

Math
<u>+sqrt(in number : double) : double</u>

Bulb
-on: Boolean
+ turnOn + turnOff + isOn: Boolean

Параметр: <направление><имя>:<тип> = <значение по умолчанию>

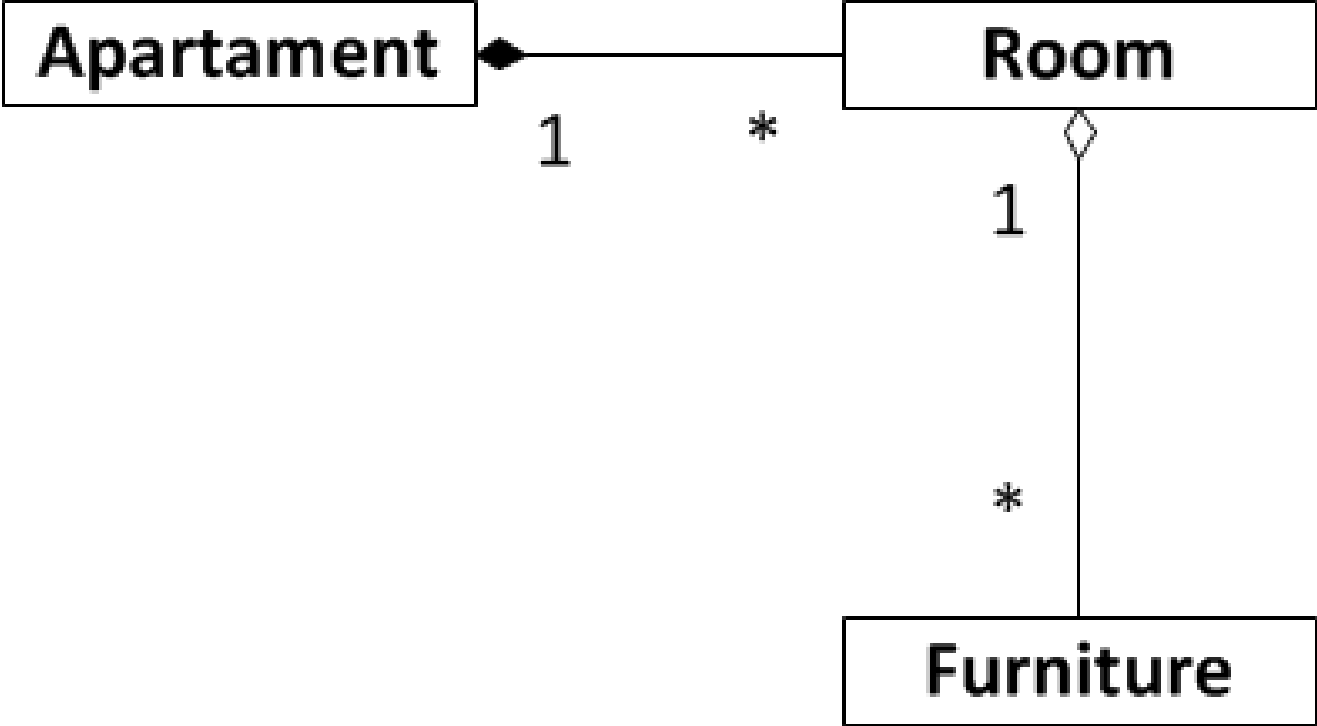
Обобщение



Агрегация и композиция

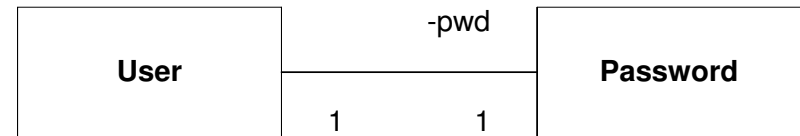
- Квартира имеет несколько комнат
- В квартире находится несколько предметов мебели
- Есть ли разница?

Агрегация и композиция



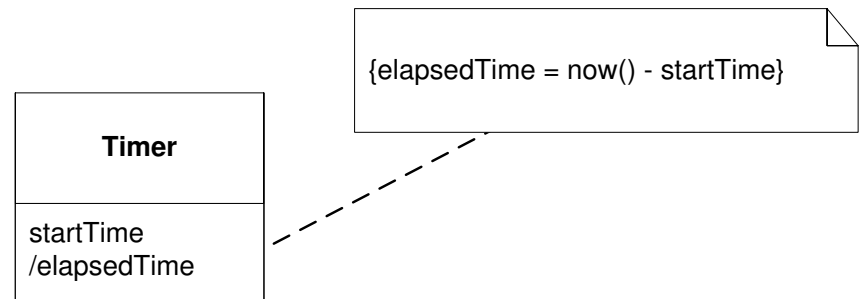
Прочие возможности

■ Видимость



■ Вычисляемые поля, ограничения

- *Текст*
- *Псевдокод*
- *OCL (Object Constraint Language)*



Выводы

- Полезны при изучении терминологии системы и моделировании предметной области
- Хорошо подходят для моделирования статической структуры программы объектно-ориентированных языков
- Полнота диаграммы вредит наглядности

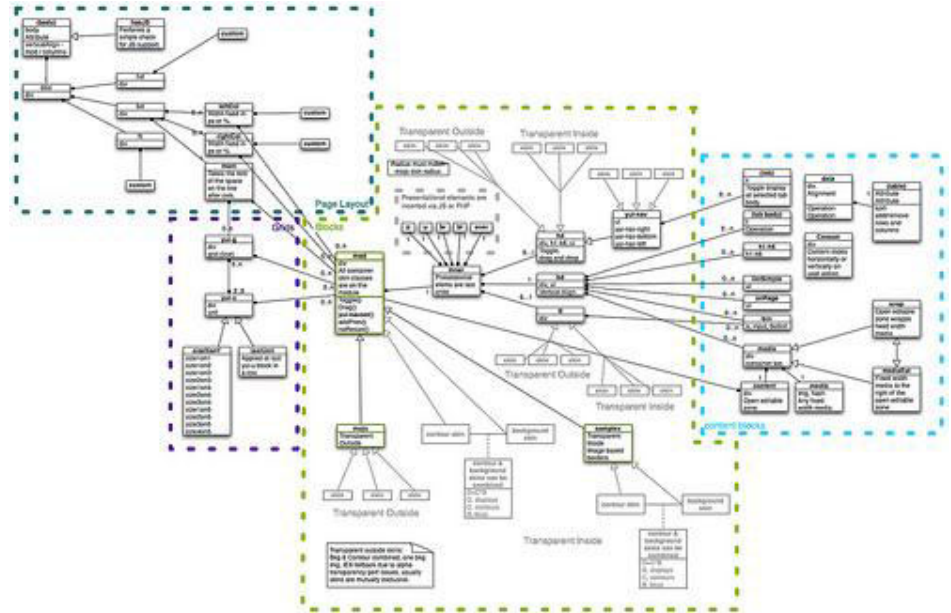
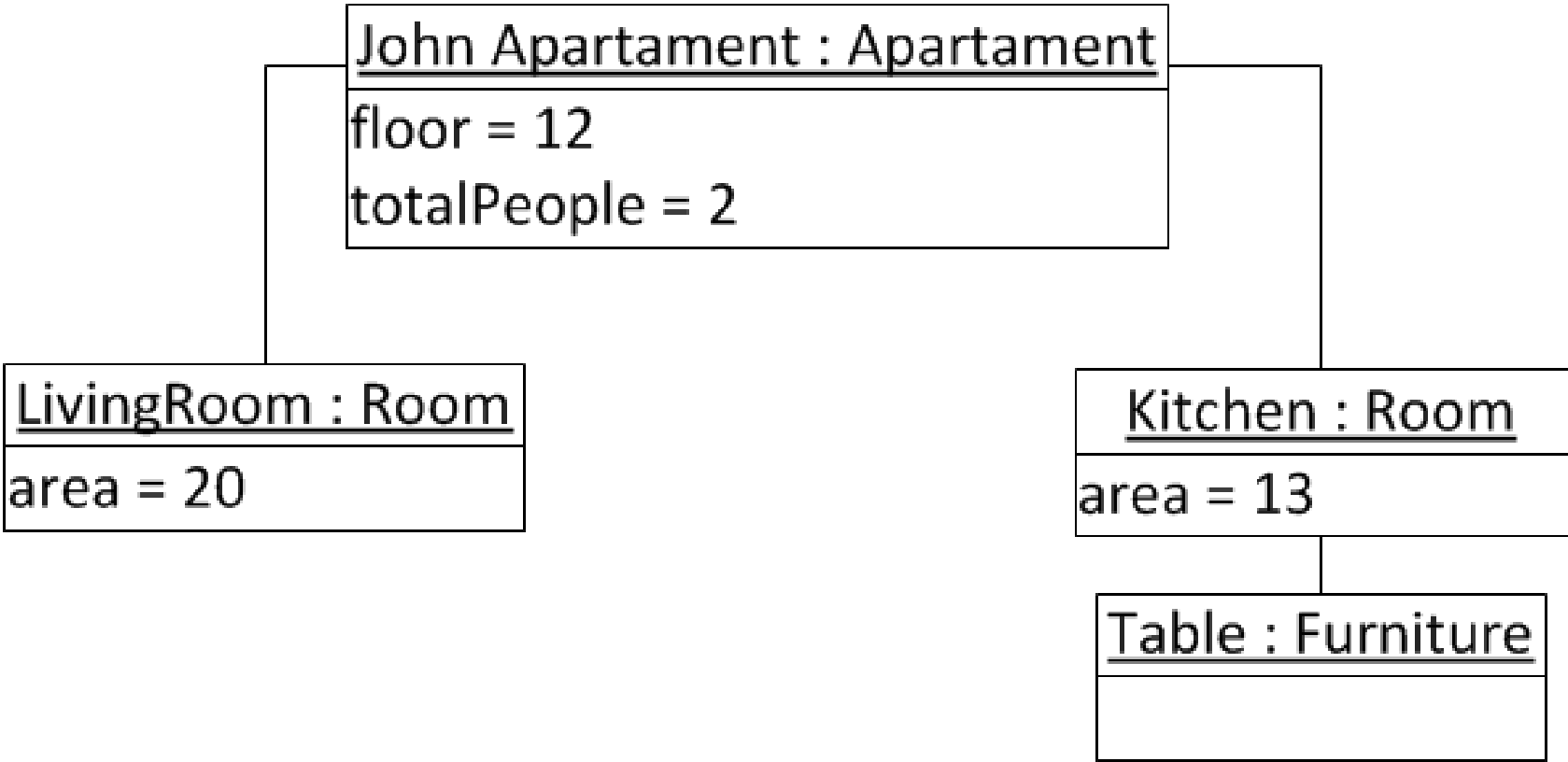


Диаграмма объектов



Упражнение

- **Нарисовать метамодель диаграммы классов**

Диаграммы состояний

- Описывают поведение систем, управляемых внешними событиями
- Основные элементы
 - Состояние
 - Переход
 - Событие
 - Сигнал, вызов, таймер, изменение состояния
 - Сторожевое условие
 - Действие

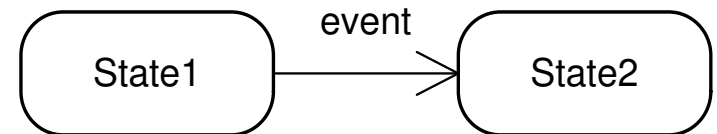
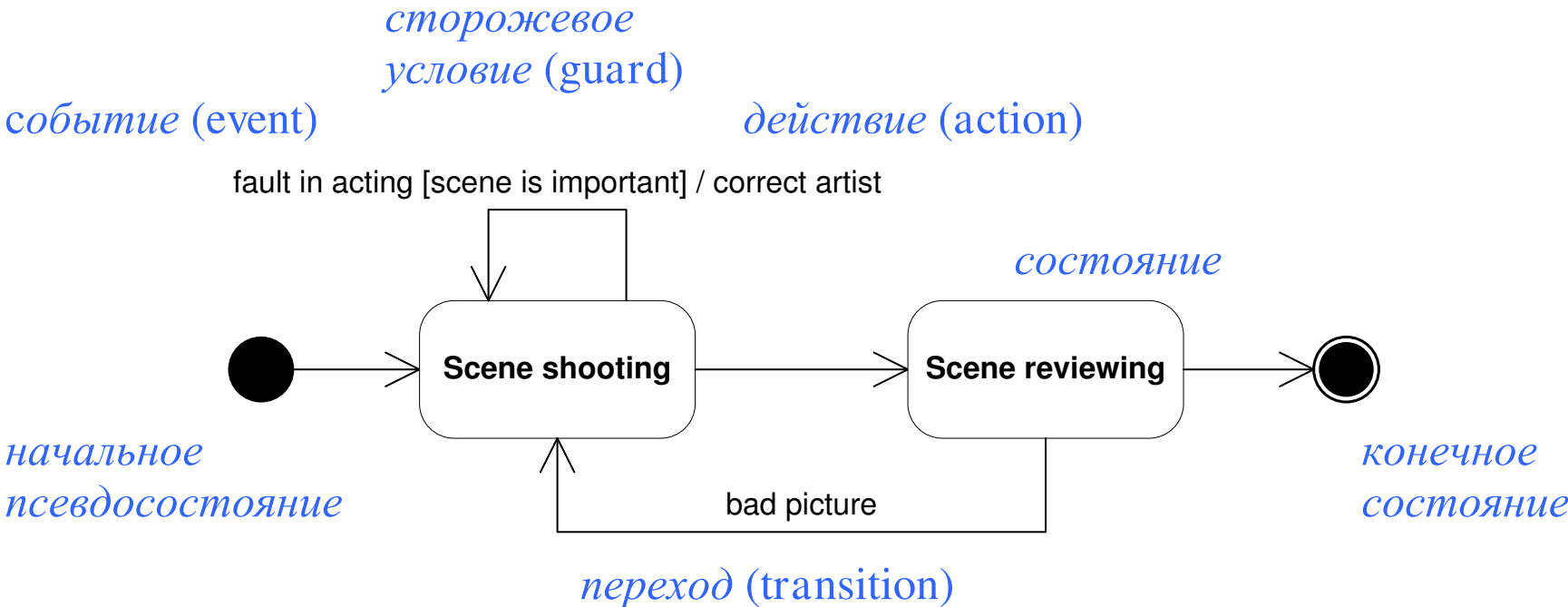
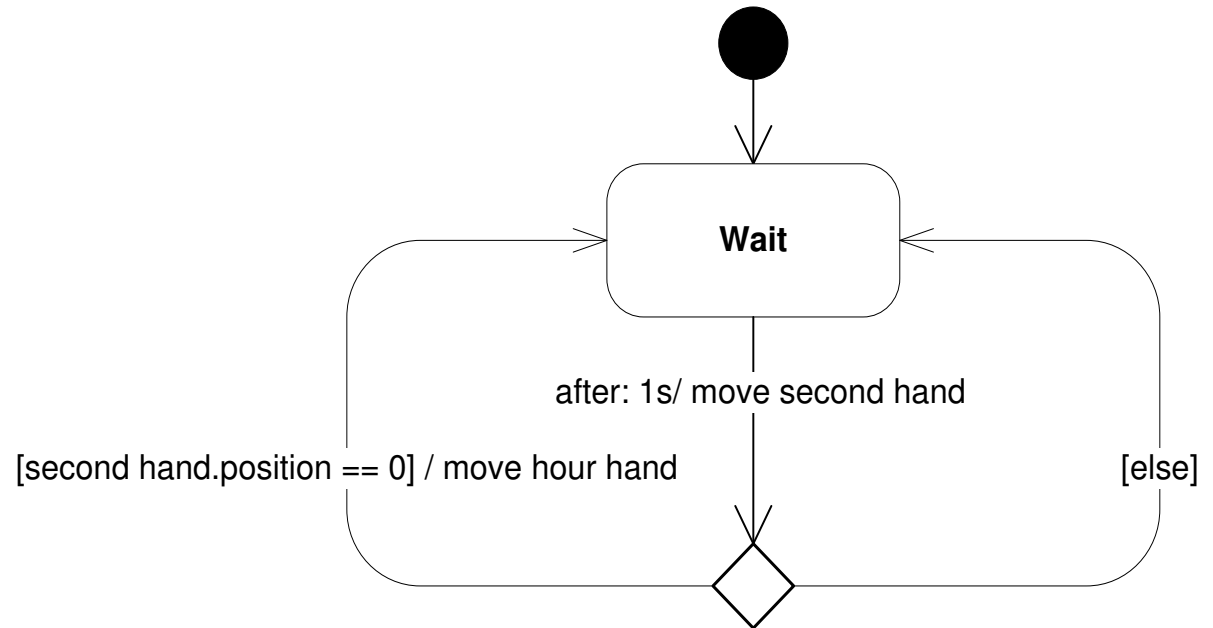


Диаграмма состояний



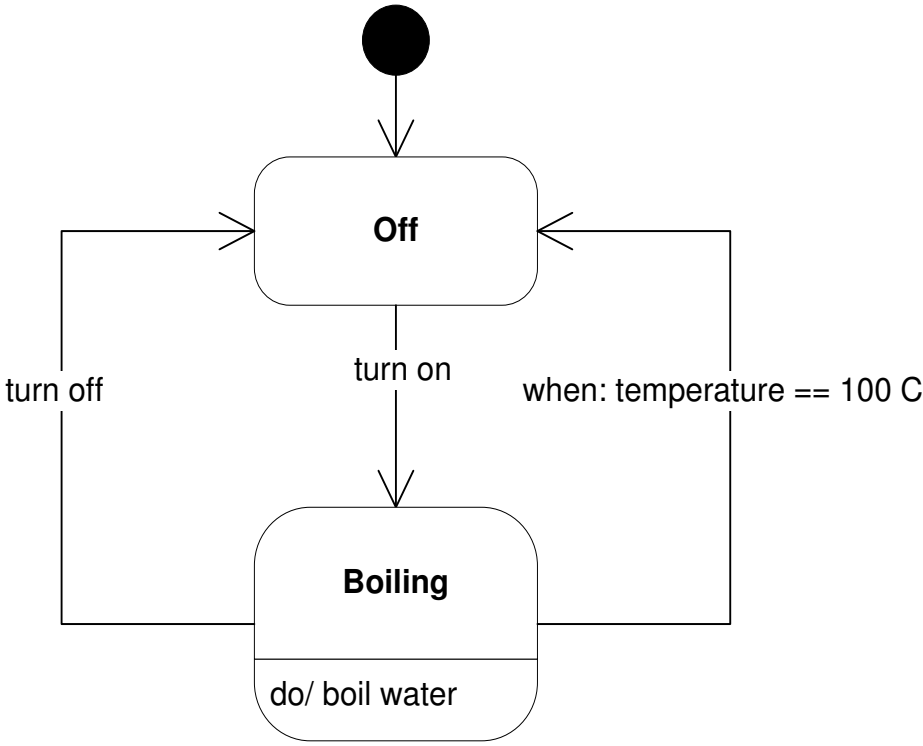
События таймера

- Пример: часы со стрелками



Состояния активности

- Пример: чайник



Выводы

- Удобны для моделирования «пассивного» поведения
- Хорошо описывают
 - *Протоколы*
 - *Пользовательский интерфейс*
 - *Жизненный цикл объекта/устройства*
 - *...*

Упражнение

- Нарисуйте диаграмму состояний для светофора

Спасибо за внимание
