



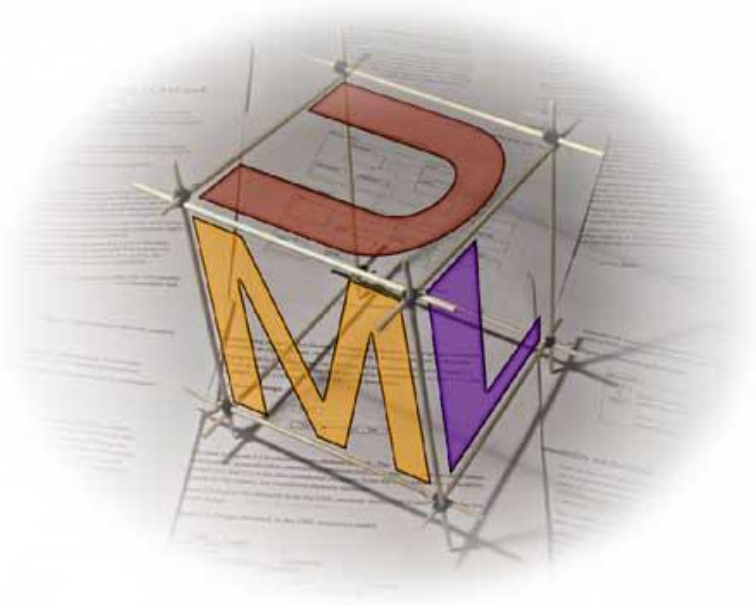
## Модуль #3

### Введение в UML

# Модуль 3 Введение в UML

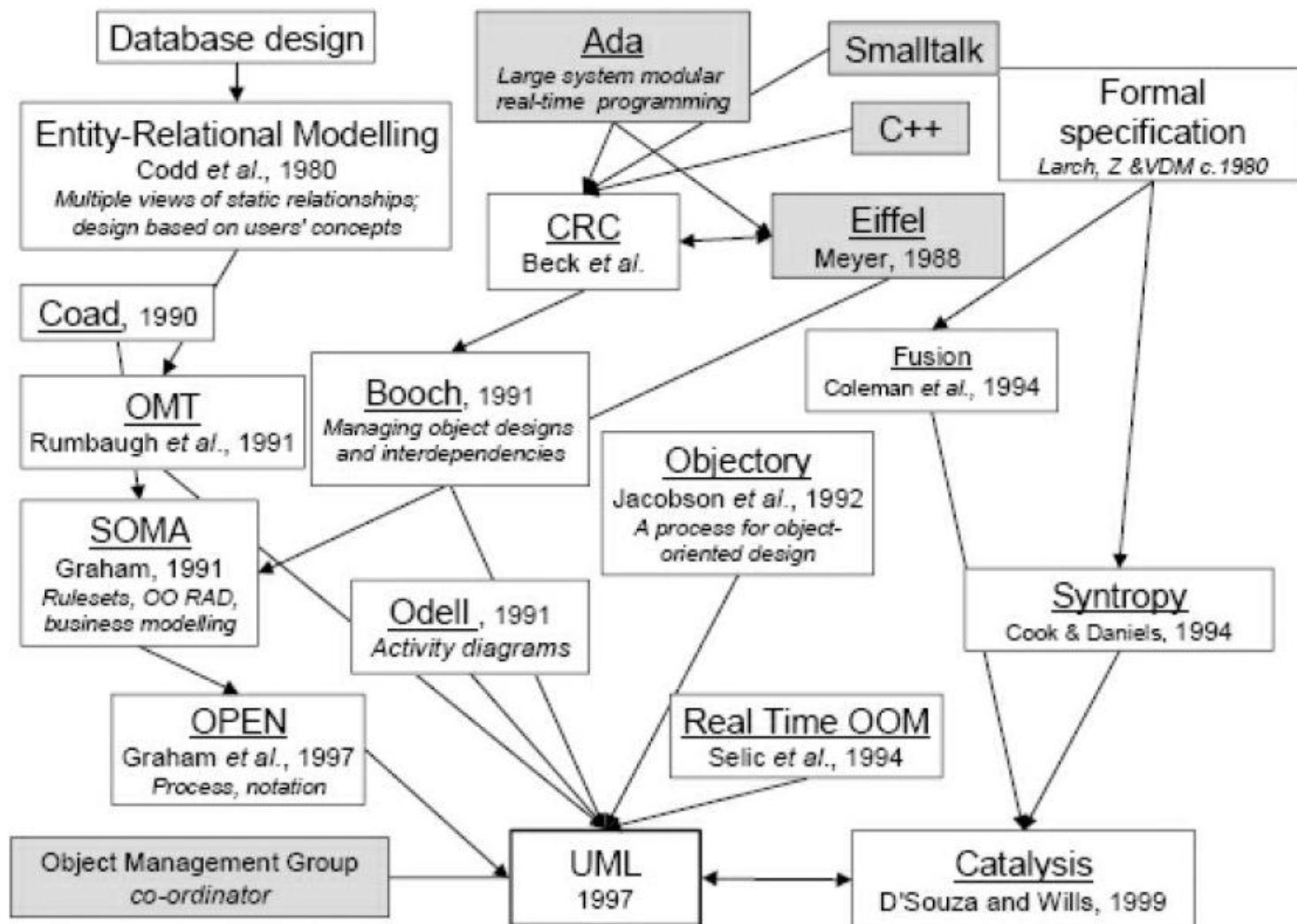
- **Введение**
- Проблемы проектирования
- Нотация
- Средства проектирования
- Виды диаграмм
- Диаграмма классов
- Диаграмма последовательности
- Обзор прочих диаграмм

# Введение в UML



- UML – унифицированный язык моделирования.
- “Унифицированный”, т.к. вобрал в себя черты нотаций Грейди Буча, Джима Румбаха, Айвара Якобсона и многих других.
- Является стандартом в ООП.
- UML принадлежит OMG.
- Профайлы UML.

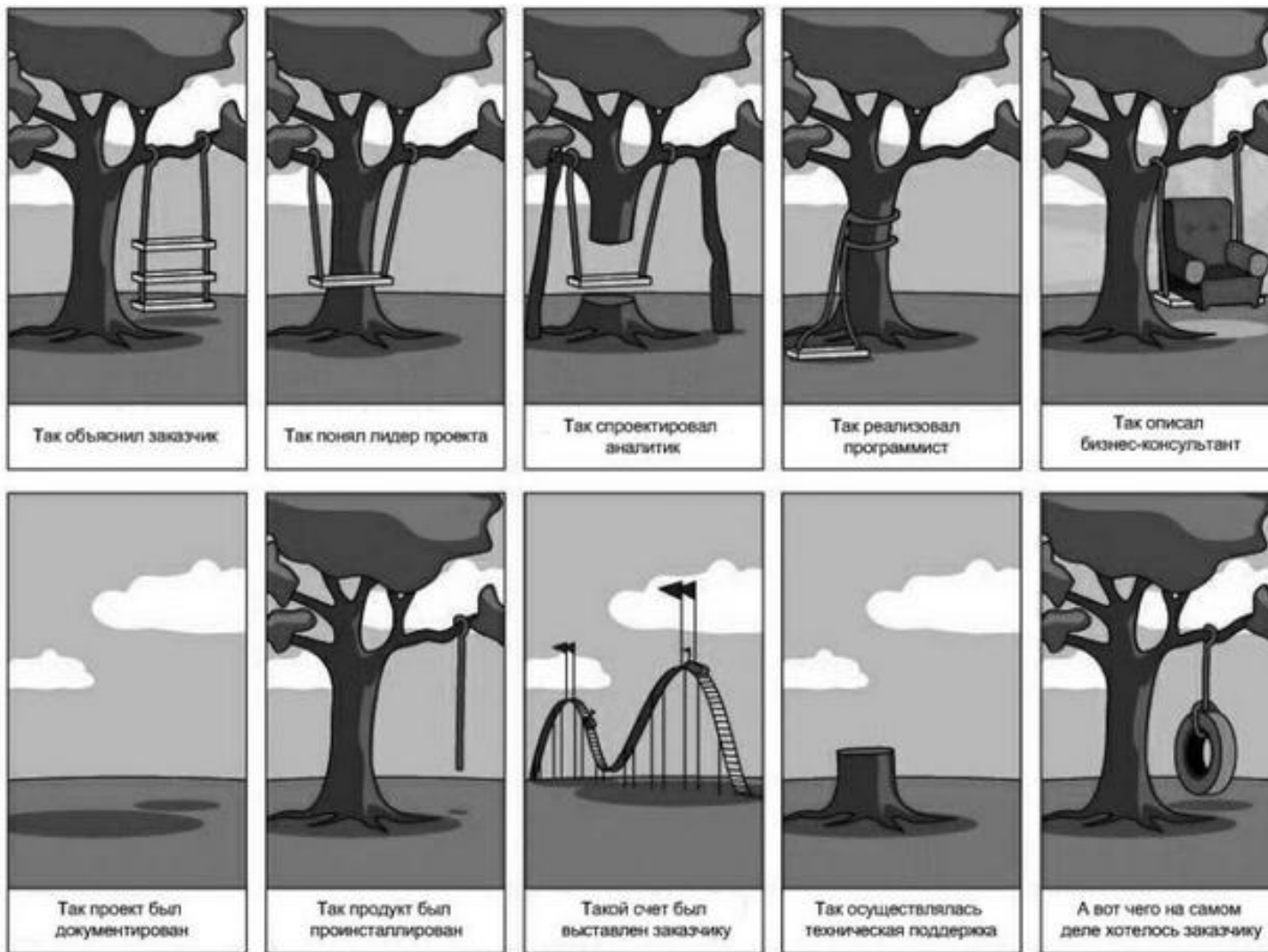
# Введение в UML



# Модуль 3 Введение в UML

- Введение
- **Проблемы проектирования**
- Нотация
- Средства проектирования
- Виды диаграмм
- Диаграмма классов
- Диаграмма последовательности
- Обзор прочих диаграмм

# Проблемы проектирования



# Проблемы проектирования

- Проблемы с коммуникацией и пониманием, вызванные отсутствием четкой спецификации создаваемого продукта.
- Объектно-ориентированные языки моделирования предназначены для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых в ходе разработки.

# Модуль 3 Введение в UML

- Введение
- Проблемы проектирования
- **Нотация**
- **Средства проектирования**
- Виды диаграмм
- Диаграмма классов
- Диаграмма последовательности
- Обзор прочих диаграмм



# Нотация

- Нотация (или “синтаксис” в других языках) – подчеркивает, что UML – язык графический и модели не «записывают», а рисуют.
- В UML используется 4 вида элементов нотации:
  - ◆ фигуры
  - ◆ линии
  - ◆ значки
  - ◆ надписи

# Специальное ПО

- Существует ПО, позволяющее осуществлять проектирование и “рисование” UML диаграмм “на компьютере” (CASE средство):
  - ◆ IBM Rational Rose
  - ◆ Borland Together
  - ◆ Gentleware Poseidon
  - ◆ Microsoft Visio
  - ◆ Telelogic TAU G2

# Модуль 3 Введение в UML

- Введение
- Проблемы проектирования
- Нотация
- Средства проектирования
- **Виды диаграмм**
- Диаграмма классов
- Диаграмма последовательности
- Обзор прочих диаграмм

# Виды диаграмм

- В UML определяется 13 типов диаграмм, разделенных на 3 группы:
  - ♦ **7** типов диаграмм представляют статическую структуру приложения (статические диаграммы).
  - ♦ **3** представляют поведенческие аспекты системы (поведенческие диаграммы).
  - ♦ **3** представляют физические аспекты функционирования системы (диаграммы взаимодействия).

# Виды диаграмм

- ◆ Диаграмма классов
- ◆ Диаграмма объектов
- ◆ Диаграмма прецедентов
- ◆ Диаграмма последовательностей
- ◆ Диаграмма состояний
- ◆ Диаграмма развертывания
- ◆ Диаграмма активности

# Модуль 3 Введение в UML

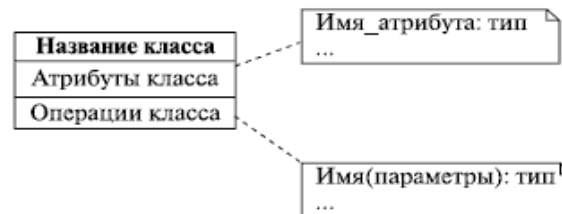
- Введение
- Проблемы проектирования
- Нотация
- Средства проектирования
- Виды диаграмм
- **Диаграмма классов**
- Диаграмма последовательности
- Обзор прочих диаграмм

# Диаграмма классов

- **Классы** – это строительные блоки любой объектно-ориентированной системы.
- Информация с диаграммы классов напрямую отображается в исходный код приложения.

# Диаграмма классов

- Класс на диаграмме отображается в виде прямоугольника, разделенного горизонтальными линиями на три части.
  - ◆ В первой части указывается название класса
  - ◆ Вторая часть содержит перечень атрибутов класса, которые характеризуют тот или иной объект класса.
  - ◆ Третья часть содержит перечень операций, отражающих его поведение в модели предметной области.





# Диаграмма классов

- При проектировании класса, он выступает в роли “черного ящика”. В UML не специфицируется тело методов класса.
- Требование инкапсуляции объектов класса диктует необходимость указания модификаторов видимости.
  - ◆ + (**public**) – открытый доступ.
  - ◆ - (**private**) – только из операции того же класса
  - ◆ # (**protected**) – только из операций этого же класса и классов, создаваемых на его основе.

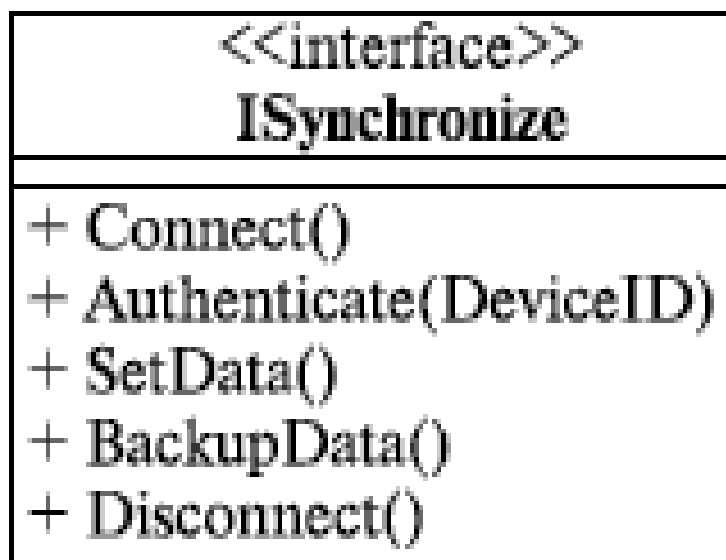
# Диаграмма классов

Телевизор
+ Язык экранного меню — Частота каналов + Порядок и именование каналов + ...
— Самодиагностики() + Включить() + Выключить() + Поиск каналов() — Декодирование сигнала() + Переключение каналов() + ...()

# Диаграмма классов

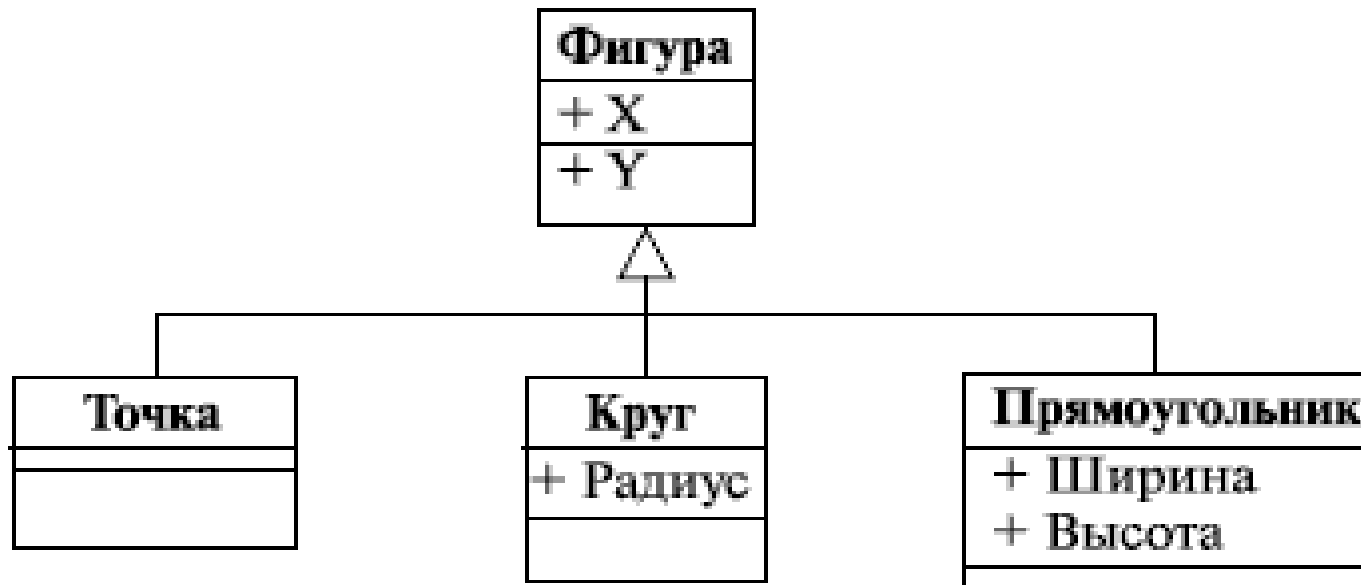
- **Интерфейс** – это логическая группа открытых (public) операций объекта.
- Один и тот же объект может иметь несколько интерфейсов.
- Интерфейс в диаграмме классов – класс со стереотипом `<<interface>>`.

# Диаграмма классов



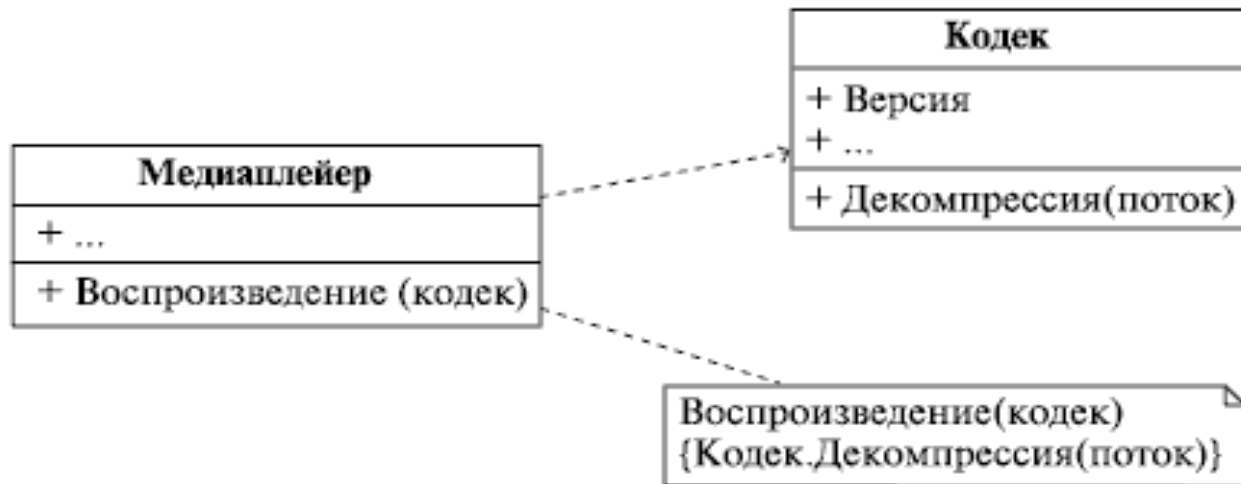
# Диаграмма классов

- **Обобщение** – это отношение между более общей сущностью, называемой суперклассом, и ее конкретным воплощением, называемым подклассом.



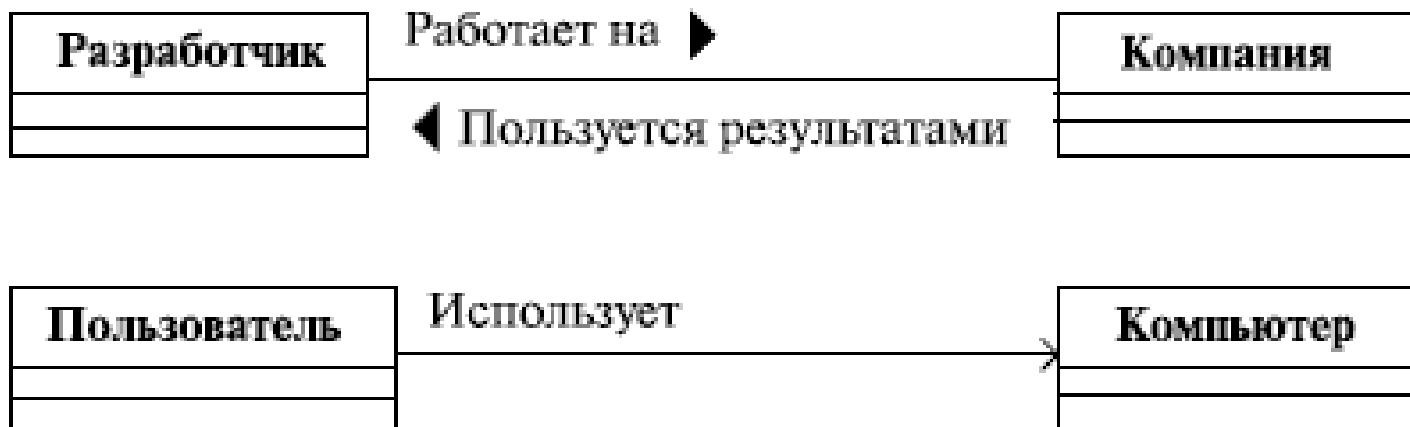
# Диаграмма классов

- Зависимость – тип отношений между объектами класса, когда реализация класса одного объекта зависит от спецификации операций класса другого объекта.



# Диаграмма классов

- **Ассоциация** – связь между объектами, по которой можно между ними перемещаться.



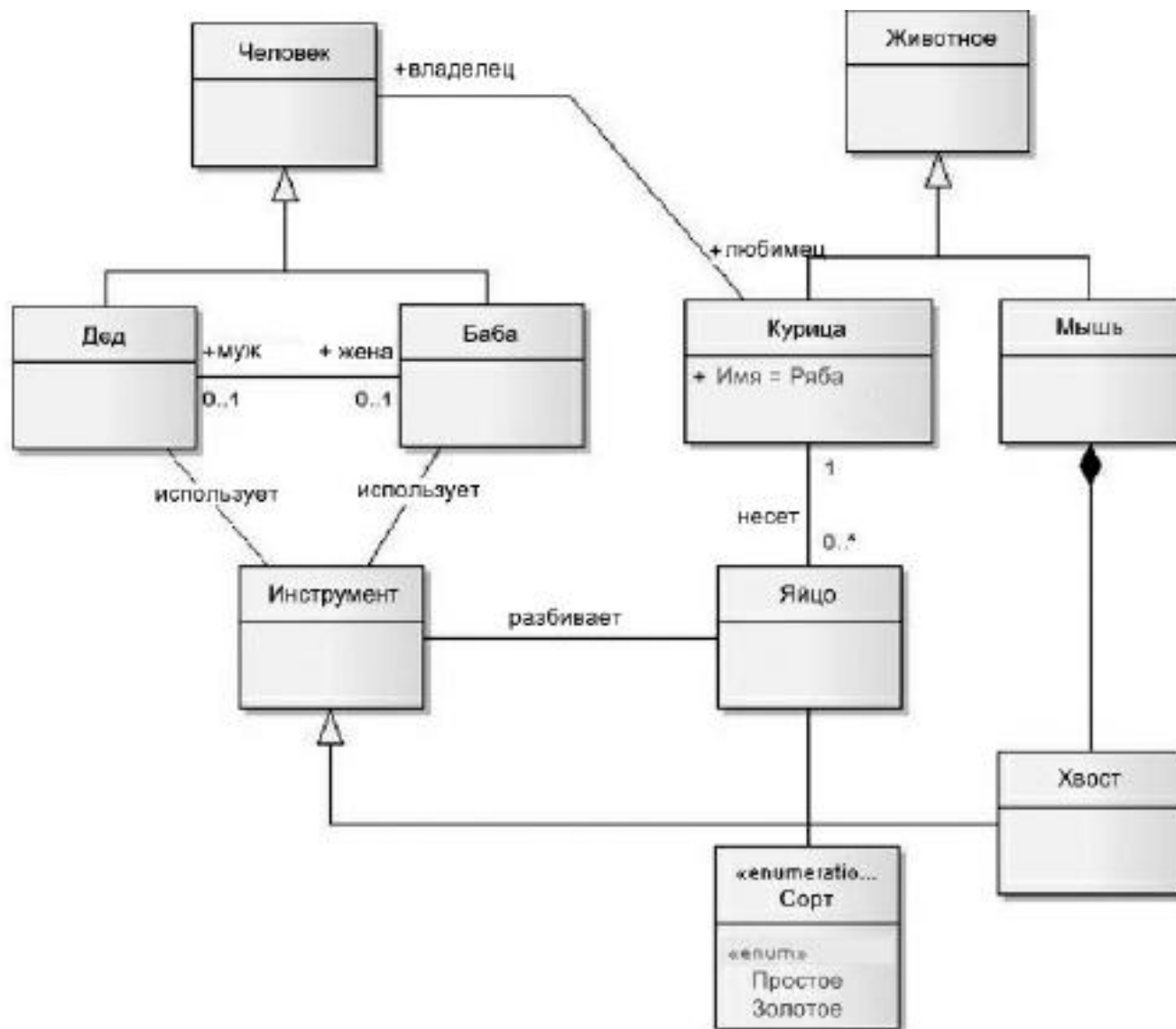
# Диаграмма классов

- Связь вида “часть-целое” называется ассоциацией с агрегированием.
- В этом случае один класс имеет более высокий статус и состоит из низших по статусу классов.





# Диаграмма классов



# Модуль 3 Введение в UML

- Введение
- Проблемы проектирования
- Нотация
- Средства проектирования
- Виды диаграмм
- Диаграмма классов
- **Диаграмма последовательности**
- Обзор прочих диаграмм

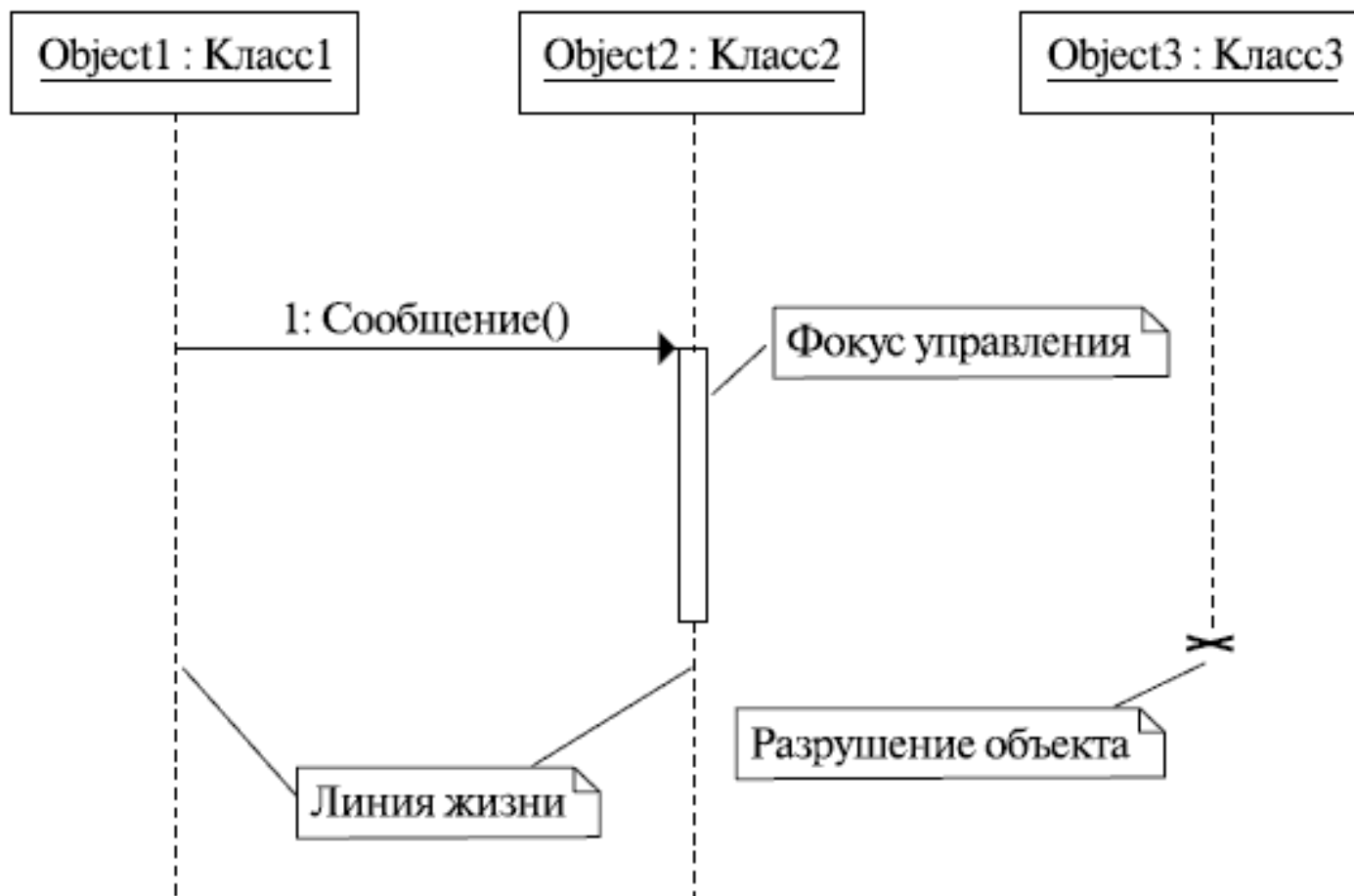
# Диаграмма последовательности

- Диаграмма последовательностей показывает последовательность, в которой объекты в процессе взаимодействия обмениваются сообщениями.
- Объект изображается как прямоугольник, внутри которого указаны подчеркнутые имя объекта и название класса, разделенные двоеточием.
- Объекты располагаются в верхней части диаграммы друг за другом. А вниз от каждого объекта тянется пунктирная линия.

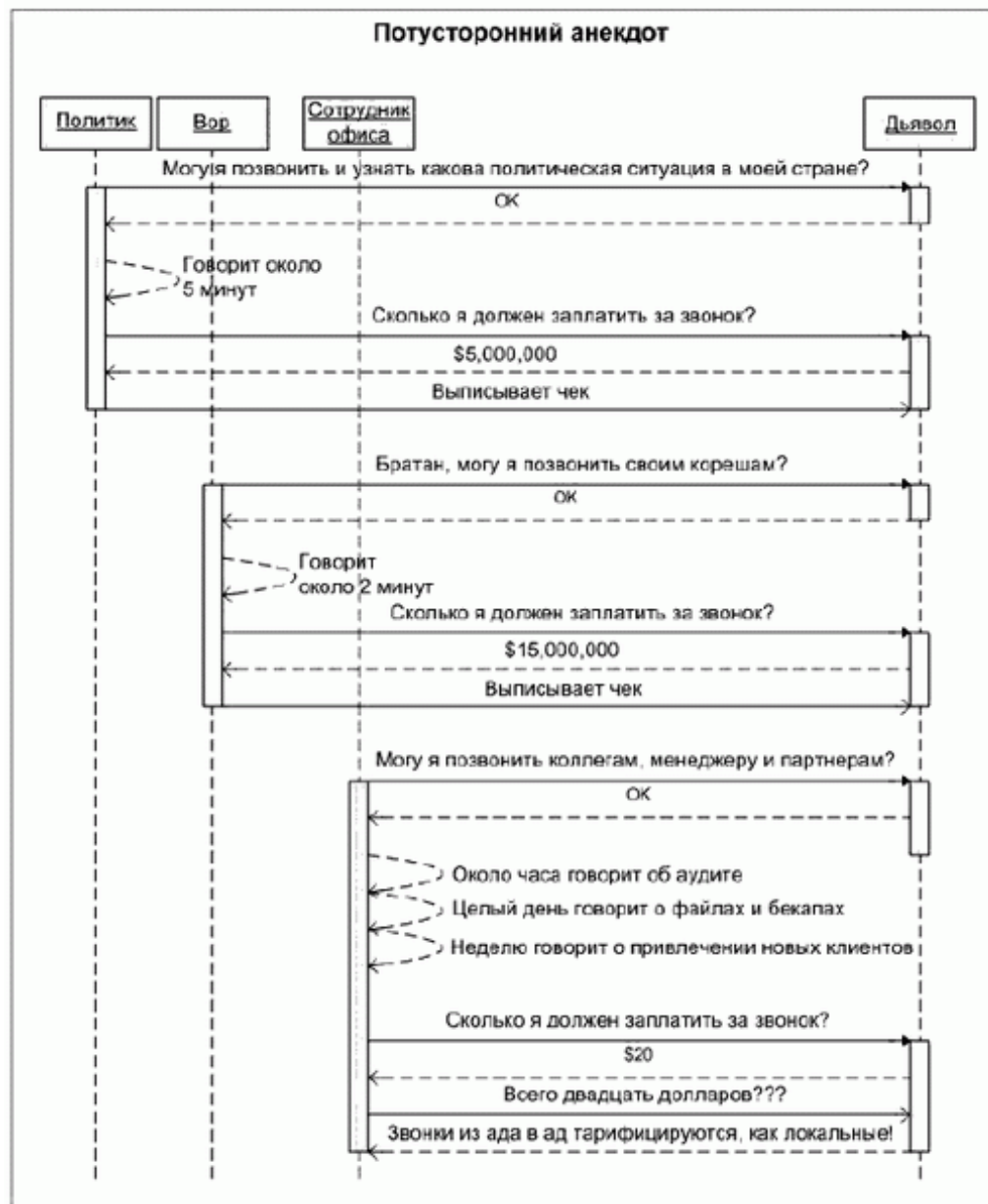
# Диаграмма последовательности

- **Линия жизни объекта** – это линия, которая изображает существование объекта на протяжении некоторого промежутка времени.
- Сообщения, которыми обмениваются объекты, изображаются в виде стрелок, направленных от линии жизни одного объекта к линии жизни другого.
- Последовательность сообщений читается сверху вниз.

# Диаграмма последовательности



# Диаграмма последовательности



# Упражнение 4

Проектирование приложения на UML.

# Модуль 3 Введение в UML

- Введение
- Проблемы проектирования
- Нотация
- Средства проектирования
- Виды диаграмм
- Диаграмма классов
- Диаграмма последовательности
- **Обзор прочих диаграмм**



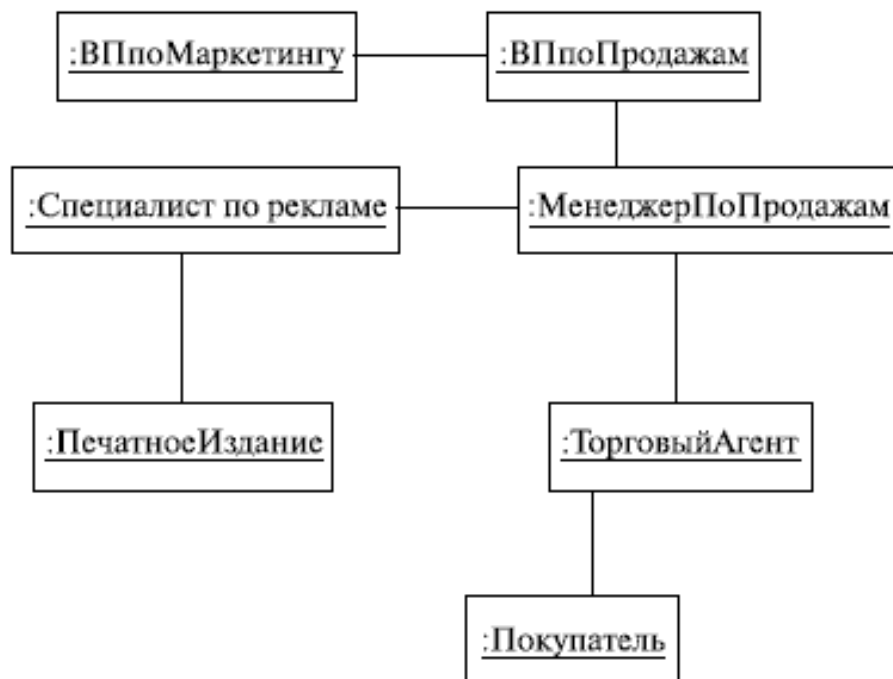
# Диаграмма прецедентов

- Сущности, с которыми взаимодействует система в процессе своей работы, называются **actor**.
- **Прецедент** (use-case) – описание отдельного аспекта поведения системы с точки зрения пользователя.



# Диаграмма объектов

- Диаграммы объектов представляют статический вид системы с точки зрения проектирования и процессов.



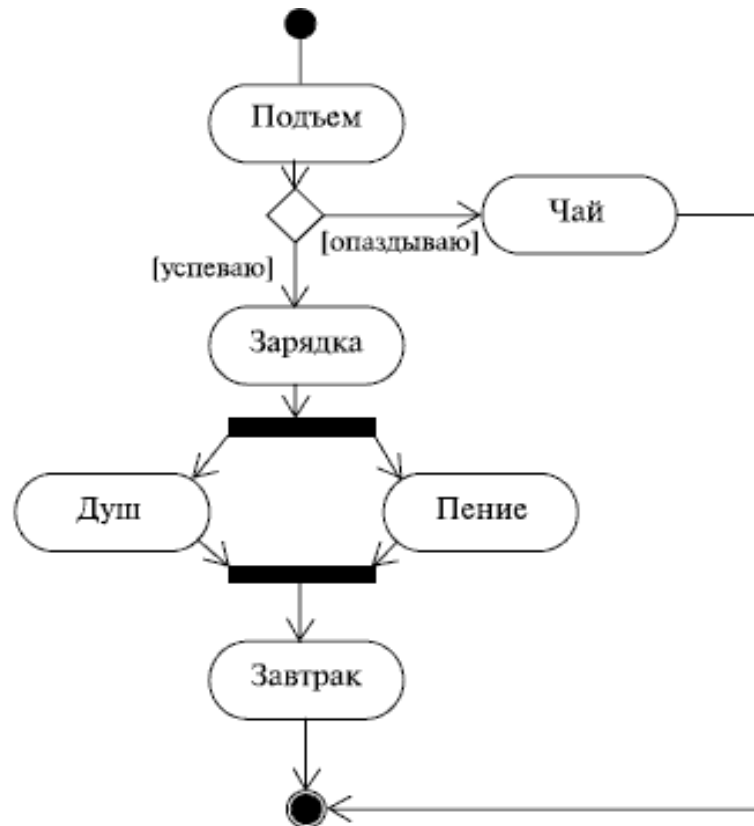
# Диаграмма состояний

- Объекты характеризуются поведением и состоянием, в котором находятся.
- Диаграмма состояний применяется для того чтобы показать каким образом работают сложные объекты.



# Диаграмма активностей

- Диаграммы активности применяют для визуализации алгоритмов, по которым работают операции классов.



# Диаграмма развертывания

- В случаях сложной архитектуры (база данных, веб-сервисы и т. д.), полезно иметь графическое представление этой инфраструктуры.



# Модуль 3 Введение в UML

- Введение
- Проблемы проектирования
- Нотация
- Средства проектирования
- Виды диаграмм
- Диаграмма классов
- Диаграмма последовательности
- Обзор прочих диаграмм