

# Анализ и проектирование на UML

**Новиков Федор Александрович**

[fedornovikov@rambler.ru](mailto:fedornovikov@rambler.ru)

Курс подготовлен по заказу  
ООО Сан Майкросистемс СПб

## Часть 2

Курс подготовлен при поддержке Sun Microsystems  
Правила использования материалов опубликованы на [www.sun.ru](http://www.sun.ru)

# План лекций

- Введение в UML
- ✓ Обзор языка
- Моделирование использования
- Моделирование структуры
- Моделирование поведения
- Управление моделями
- Тенденции развития языка
- UML и процесс разработки

# **2. Обзор языка**

- **2.1. Модель UML (UML Model)**
  - **Элементы модели (Model Elements)**
  - **Сущности (Entities)**
  - **Отношения (Relations)**
- **2.2. Диаграммы (Diagrams)**
- **2.3. Представления (Views)**
- **2.4. Общие механизмы (General Mechanisms)**
- **2.5. Профили (Profiles)**
- **2.6. Свойства модели (Model Properties)**

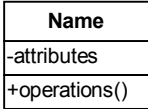


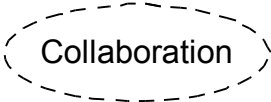
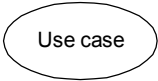
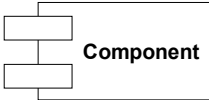
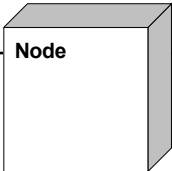
## 2.1. Модель UML

- **Граф  $G(V,E)$  = непустое множество (вершины) + множество неупорядоченных пар вершин (ребра)**
- **Орграф (упорядоченные ребра = дуги), мультиграф (кратные ребра), псевдограф (петли), гиперграф (ребра с 1, 2, 3, ... концами)**
- **Орграф = бинарное отношение на множестве**
- **Нагруженный граф:  $f : V \rightarrow M, g: E \rightarrow M$**
- **Модель UML = нагруженный мульти- псевдо- гипер- орграф**
- **Вершины = сущности, ребра = отношения**

# Основные сущности UML 1.x & 2.0

- **Структурные**
  - Класс (class), Объект (Object), Интерфейс (interface), Компонент (component), Действующее лицо (actor), Вариант использования (use case), Кооперация (collaboration), Узел (node)
- **Поведенческие**
  - Состояние (state)
  - Деятельность (activity)
  - Действие (action)
- **Группирующие**
  - Пакет (Package)
- **Аннотационные**
  - Примечание (Note)

# Нотация для структурных сущностей UML 1.x

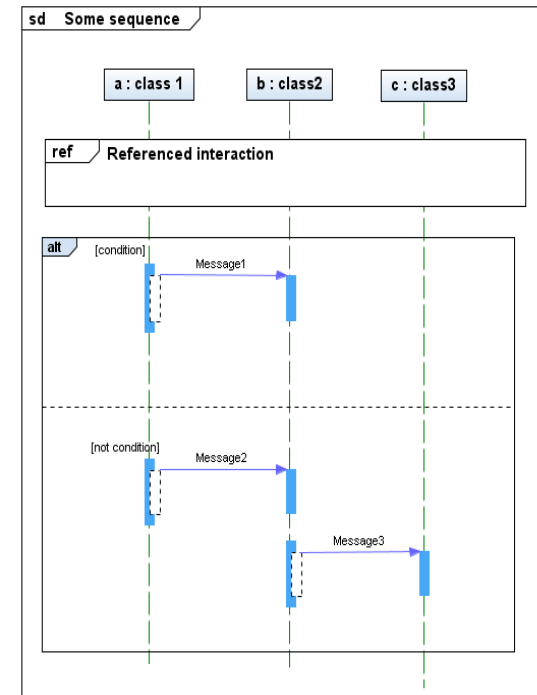
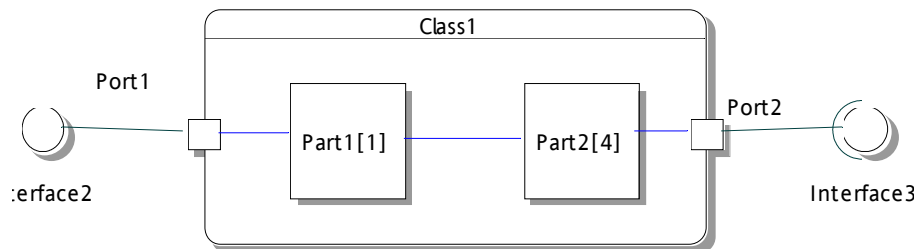
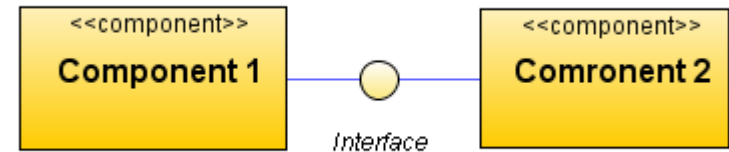
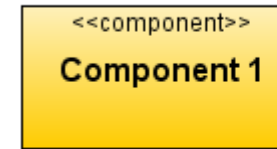
Класс	
Действующее лицо	 Actor
Интерфейс	Interface 
Кооперация	 Collaboration
Вариант использования	 Use case
Компонент	 Component
Узел	 Node

# Нотация для других сущностей UML 1.x

Состояние	
Деятельность	
Развилка / слияние	
Ветвление / соединение	
Пакет	
Примечание	

# Изменения нотации в UML 2.0

- Прямоугольник со значком для структурных сущностей вместо специальных фигур
- Предоставляемые и требуемые интерфейсы
- Рамки диаграмм и рамки шагов на диаграммах последовательности
- Вложенность диаграмм в фигуры (структурированные классификаторы и составные состояния)
- Порты на границах составных фигур





# Отношения

- **Зависимость (dependency)**
  - Зависимое -----> Независимое
- **Ассоциация (association)** —
  - **Агрегирование (aggregation)** —◇
  - **Композиция (composition)** —◆
- **Обобщение (generalization)**
  - Потомок —> Предок
- **Реализация (implementation)**
  - Implementation -----▷ Interface

## 2.2. Диаграммы

- Сущности и отношения – лексемы
- Лексемы можно комбинировать произвольным (допустимым) образом
- Диаграмма – изображение *части* графа модели
- Модель – *объединение* диаграмм
- Для удобства рекомендуется использовать *канонические* диаграммы

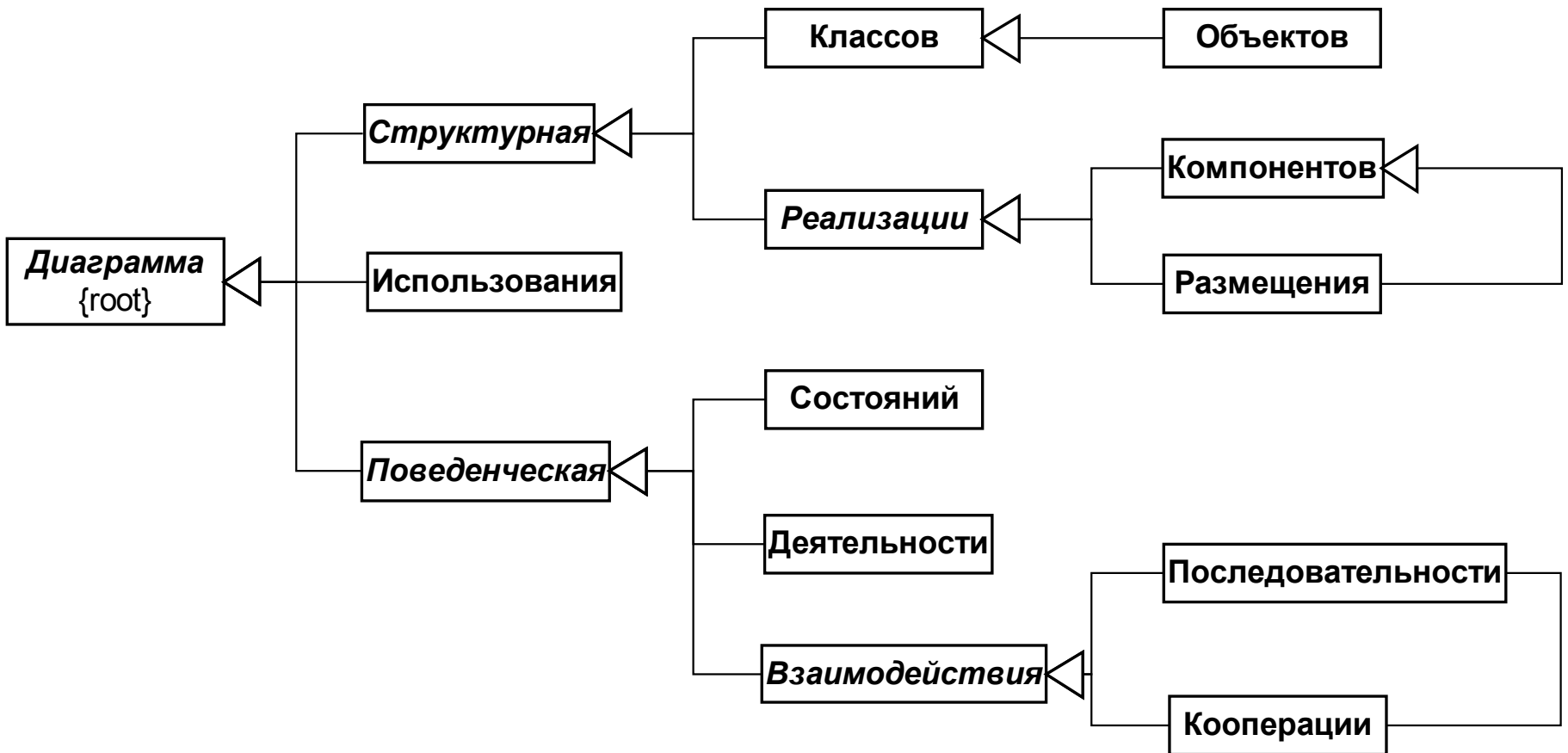
# Канонические диаграммы UML 1.x

- Классов (class diagram)
- Объектов (object diagram)
- Использования (use case diagram)
- Последовательности (sequence diagram)
- Кооперации (collaboration diagram)
- Состояний (state chart diagram)
- Деятельности (activity diagram)
- Компонентов (component diagram)
- Размещения (deployment diagram)

# Классификация диаграмм

- Структурные (static structure diagrams)
  - Диаграмма классов
    - Диаграмма объектов
  - Реализации (implementation diagrams)
    - Диаграмма компонентов
    - Диаграмма размещения
- Диаграмма использования
- Поведенческие (behavioral diagrams)
  - Диаграмма состояний
  - Диаграмма действий
  - Взаимодействия (interaction diagrams)
    - Диаграммы кооперации
    - Диаграммы последовательности

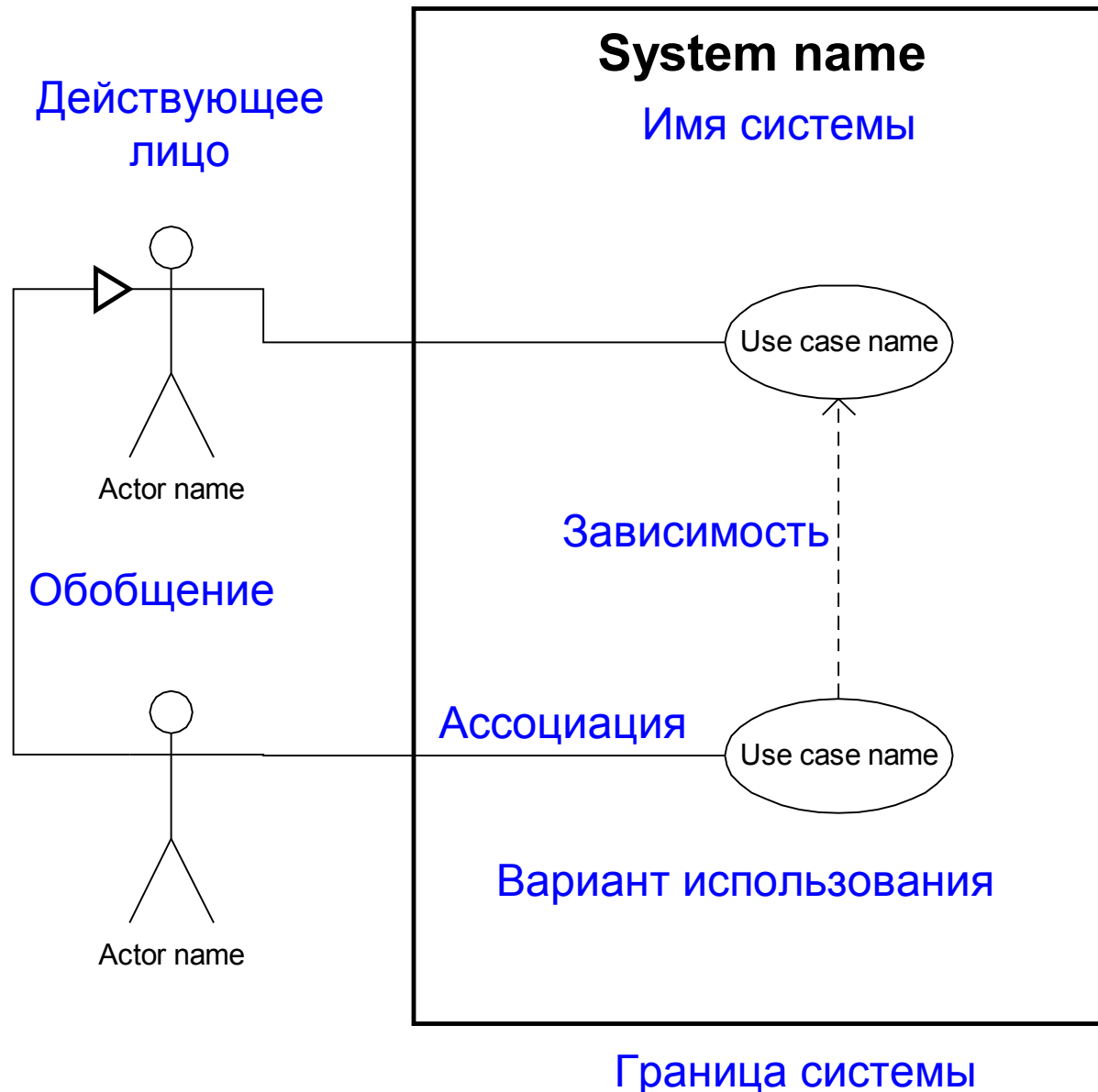
# Иерархия типов диаграмм UML 1.x



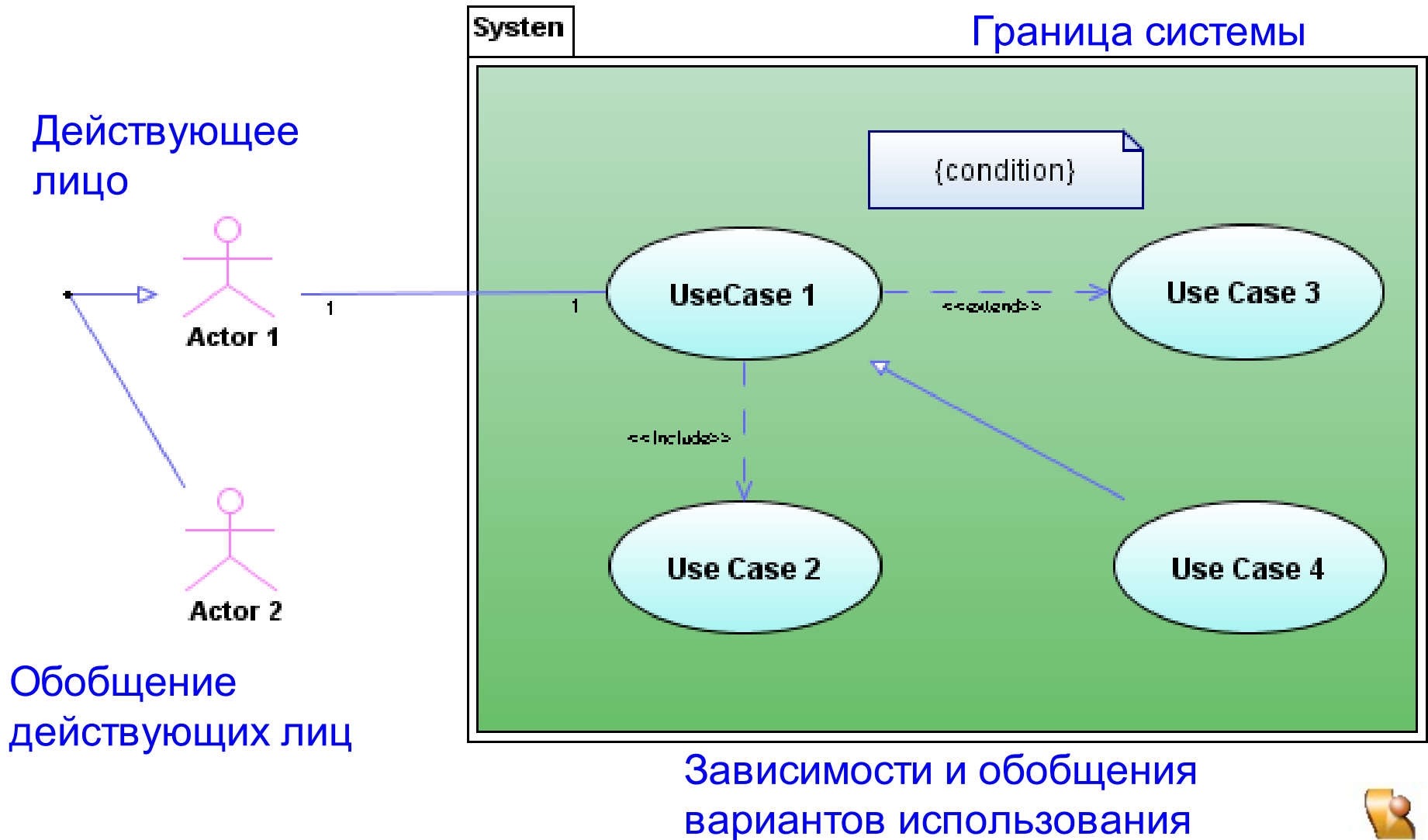
# Канонические диаграммы UML 2.0

- **Диаграмма использования =**
- **Диаграмма классов**
  - + внутренняя структура
  - + диаграмма пакетов
  - – диаграмма объектов
- **Диаграммы компонентов и размещения → развертывания**
  - + предоставляемые и требуемые интерфейсы
- **Диаграмма состояний**
  - + точки входа/выхода составных состояний
- **Диаграмма деятельности**
  - + контакты действий
  - + области
- **Диаграмма последовательности**
  - + составные шаги + обзорная диаграмма + синхронизации
- **Диаграмма кооперации → Диаграмма коммуникации**

# Диаграмма использования (1.x)

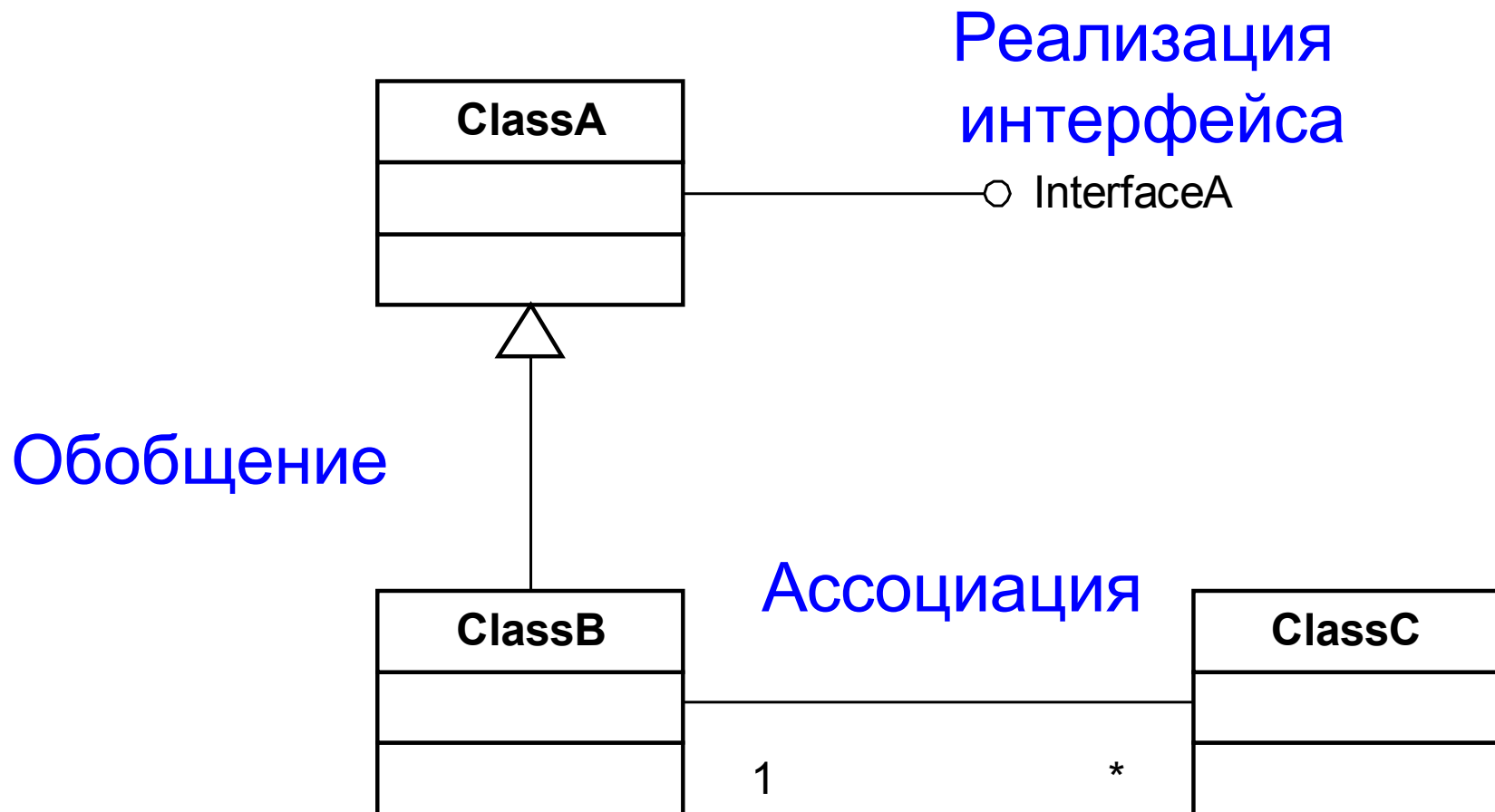


## Диаграмма использования (2.0)

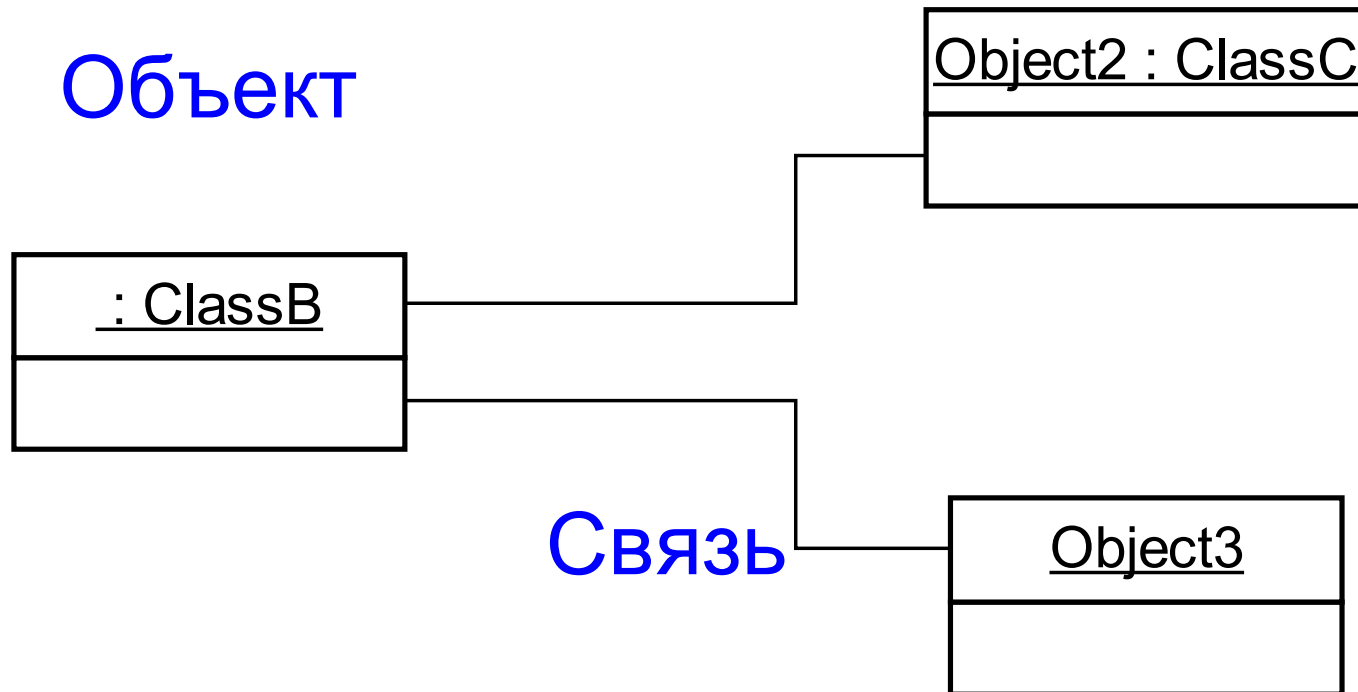




# Диаграмма классов (1.x)

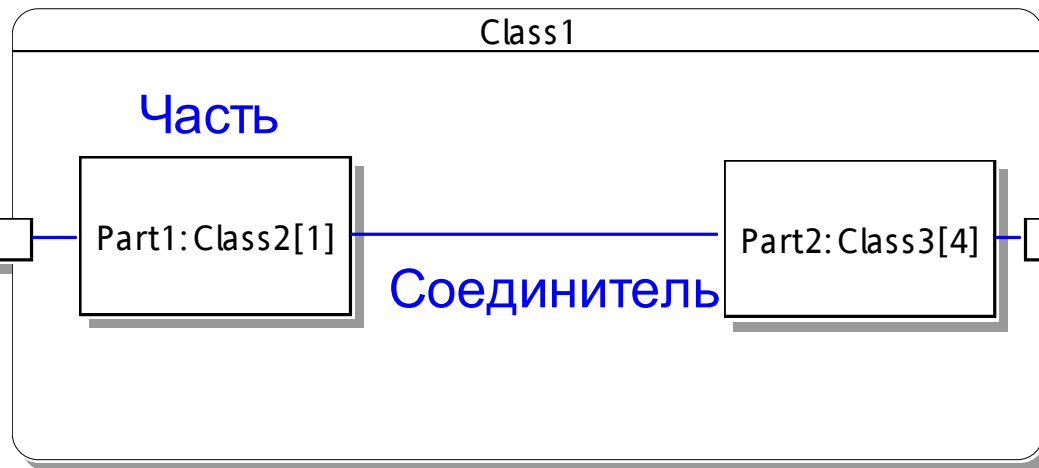
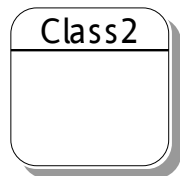
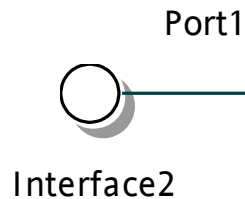


# Диаграмма объектов

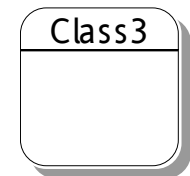
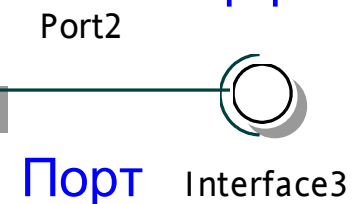


# Диаграмма внутренней структуры (Structured Classifier) в UML 2.0

Предостав-  
ляемый  
интерфейс



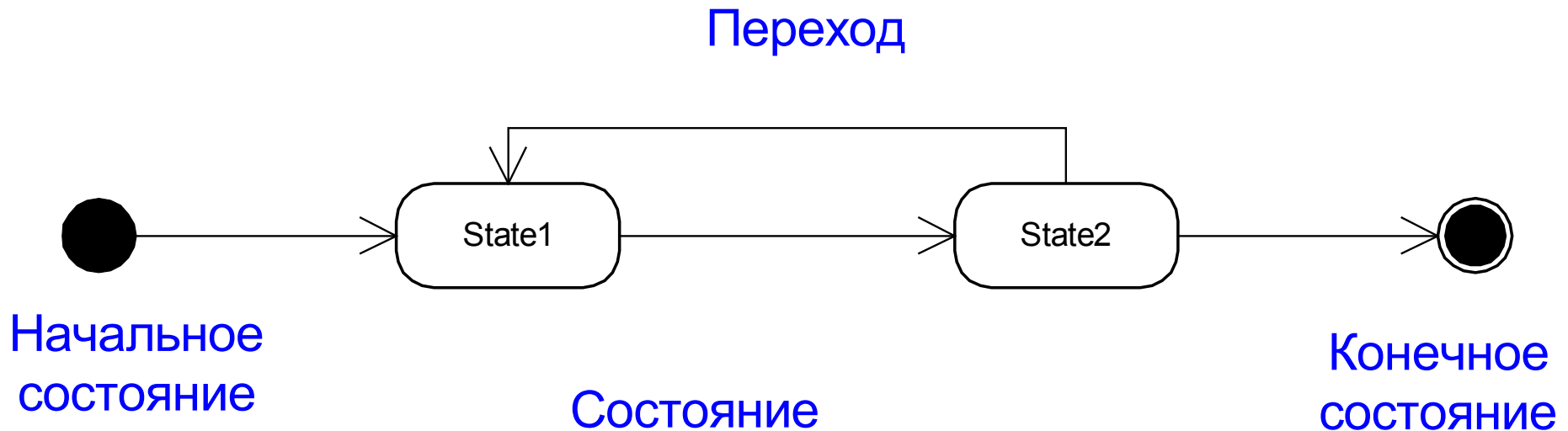
Требуемый  
интерфейс



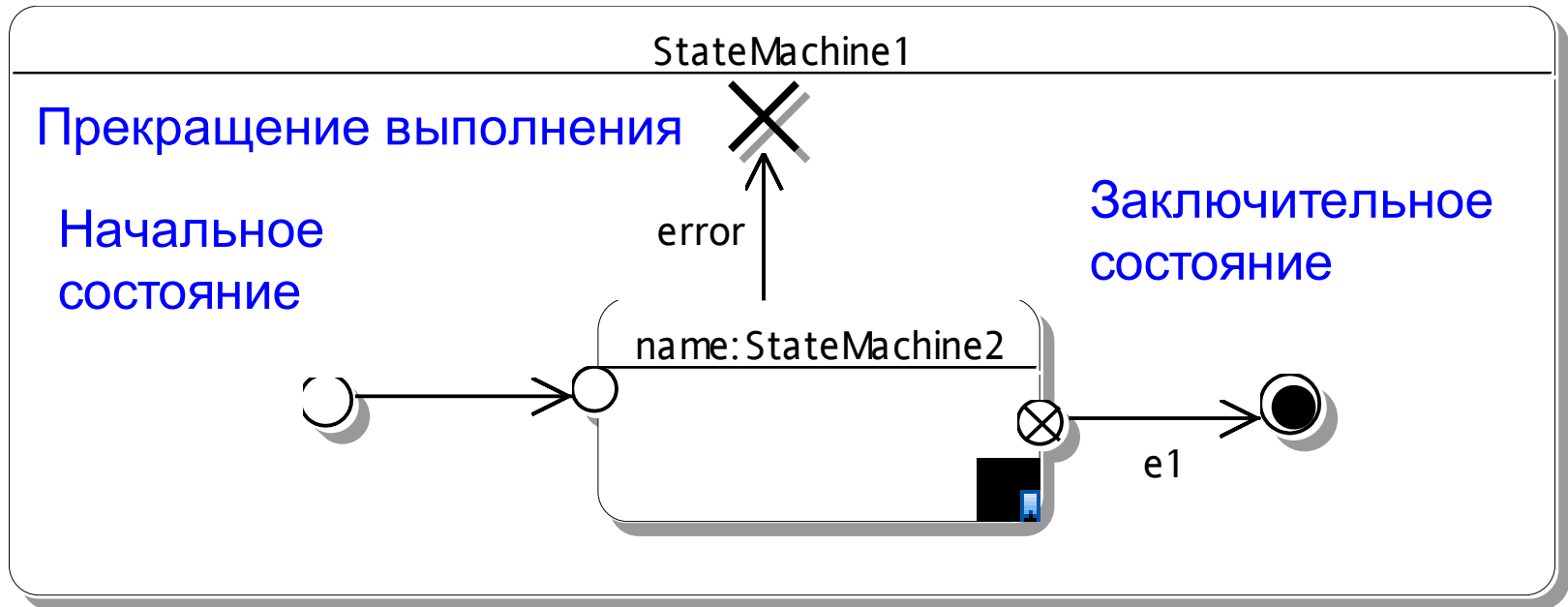
Together



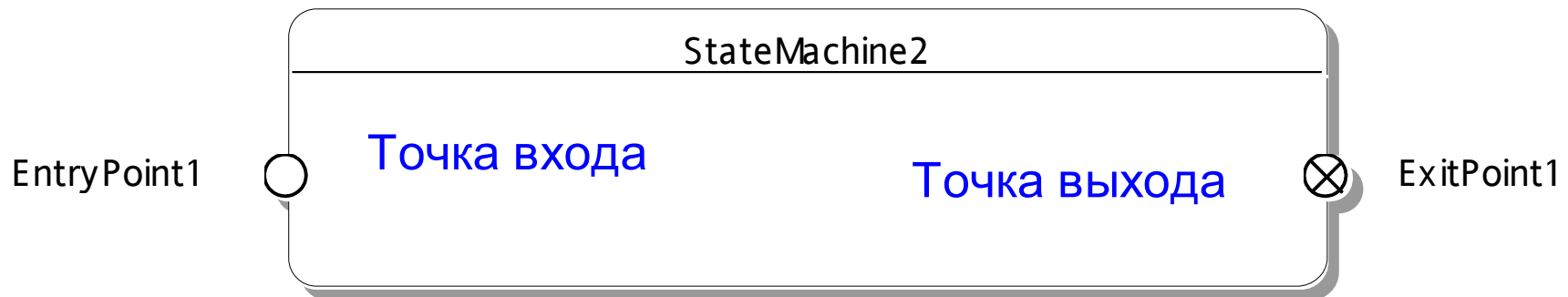
# Диаграмма состояний (1.x)



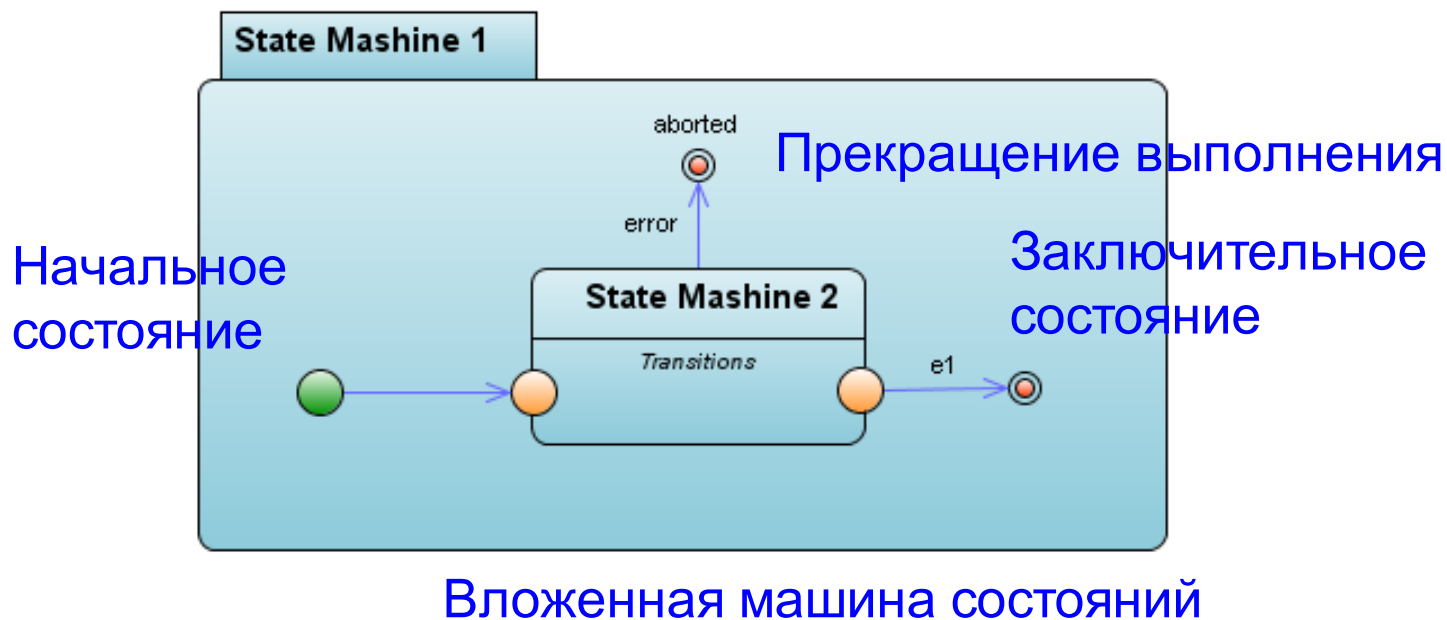
# Диаграмма состояний (2.0)



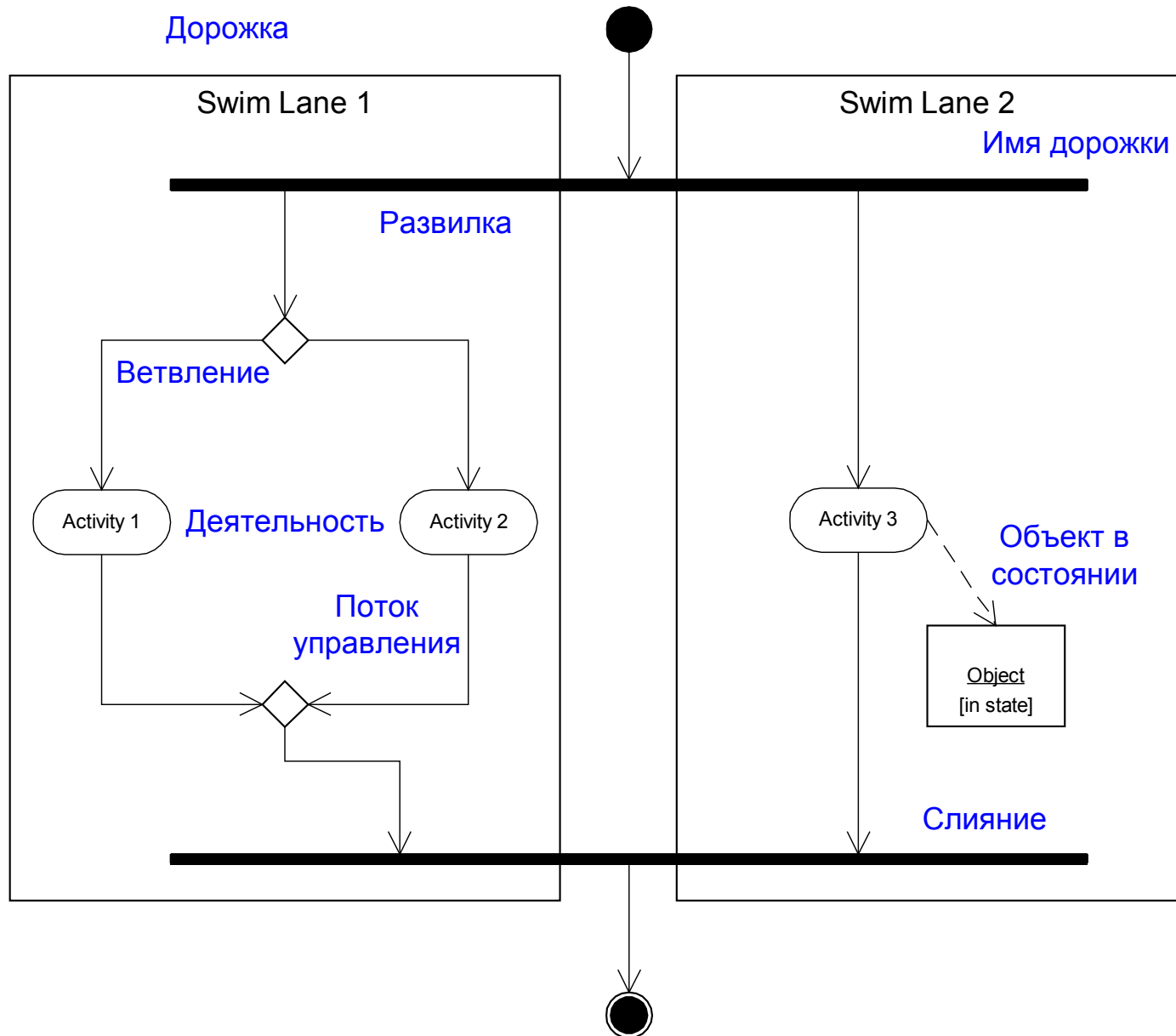
Вложенная машина состояний



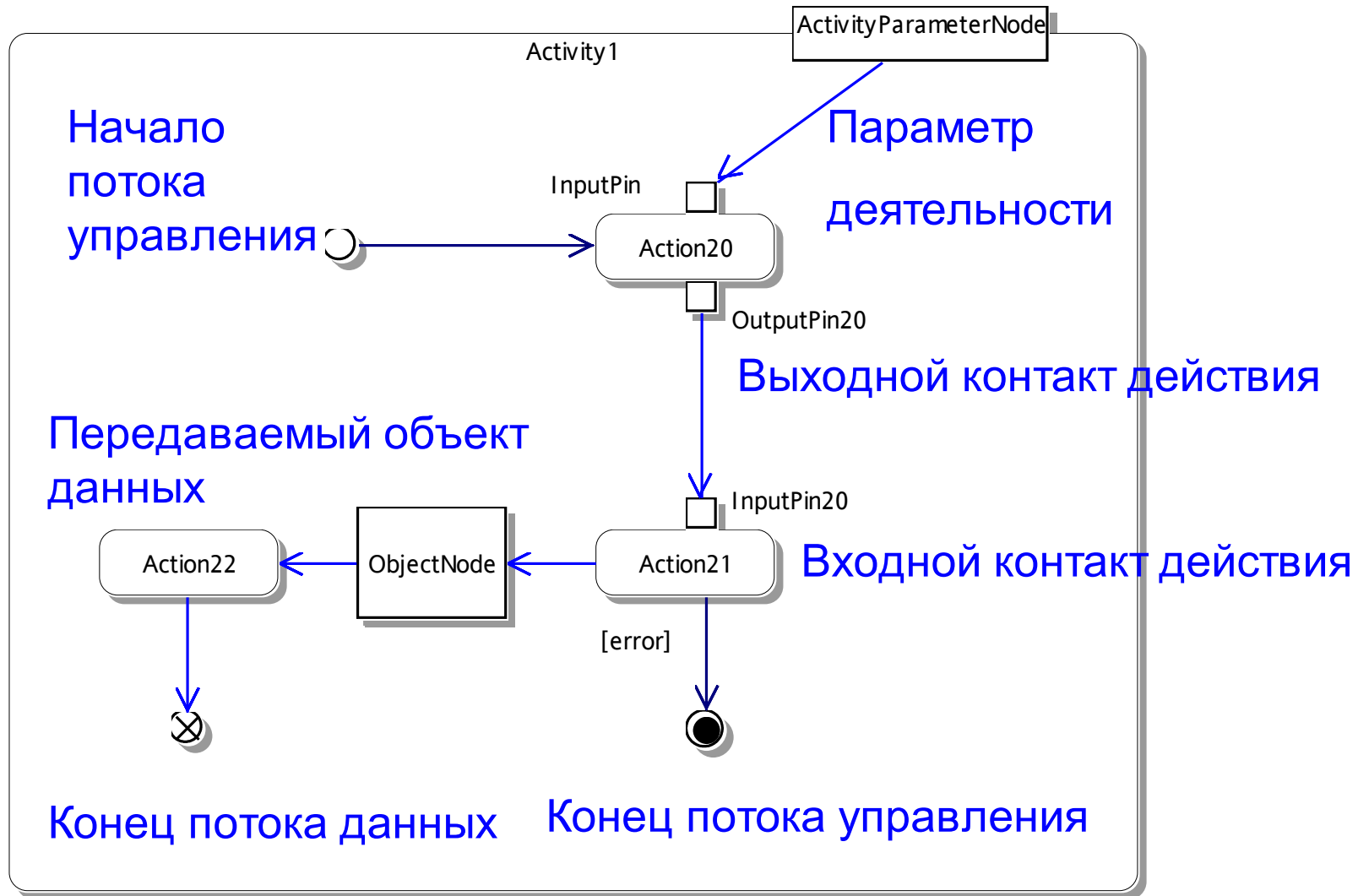
# Диаграмма состояний (2.0)



# Диаграмма деятельности (1.x)

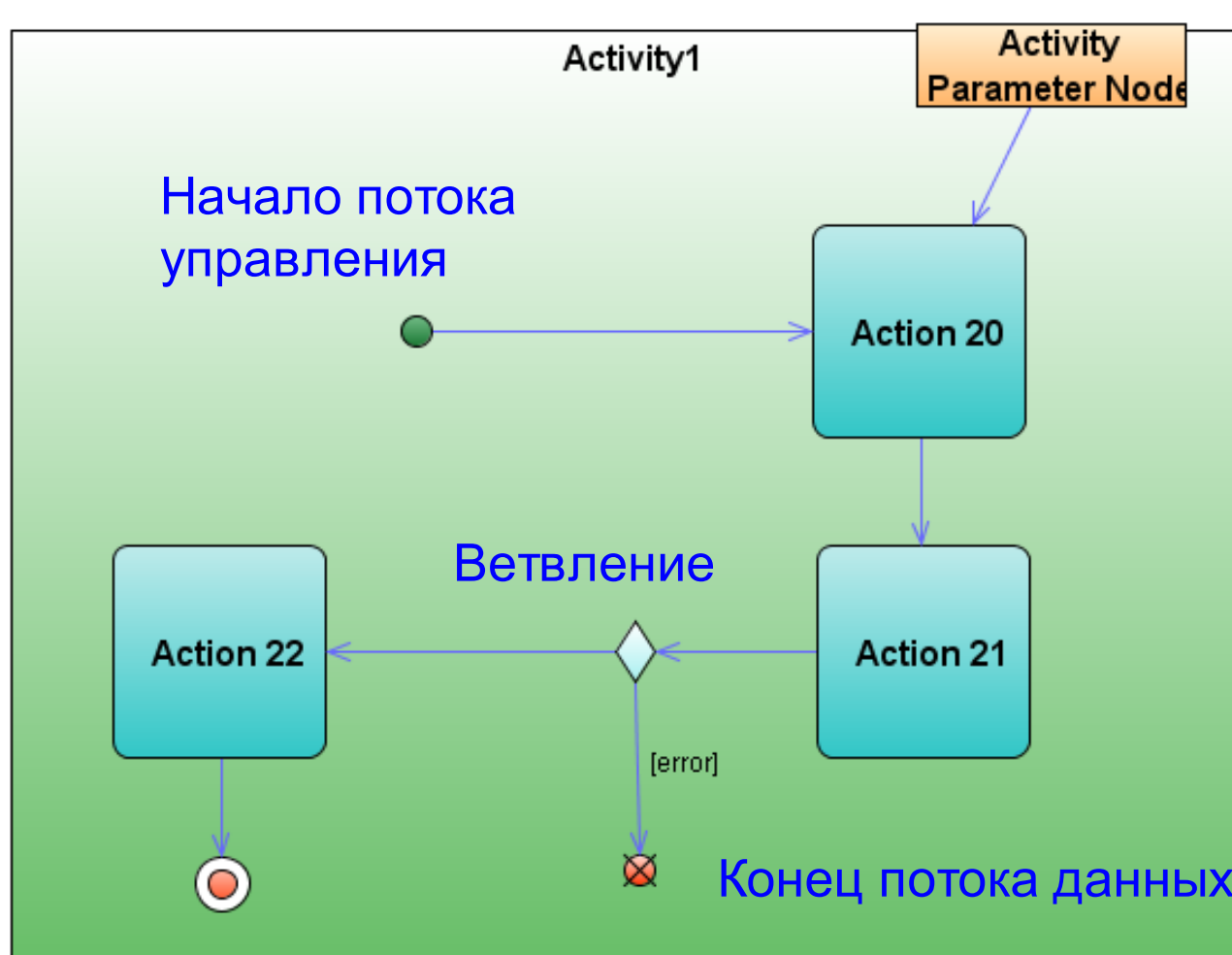


# Диаграмма деятельности (2.0)





# Диаграмма деятельности (2.0)



Параметр  
деятельности

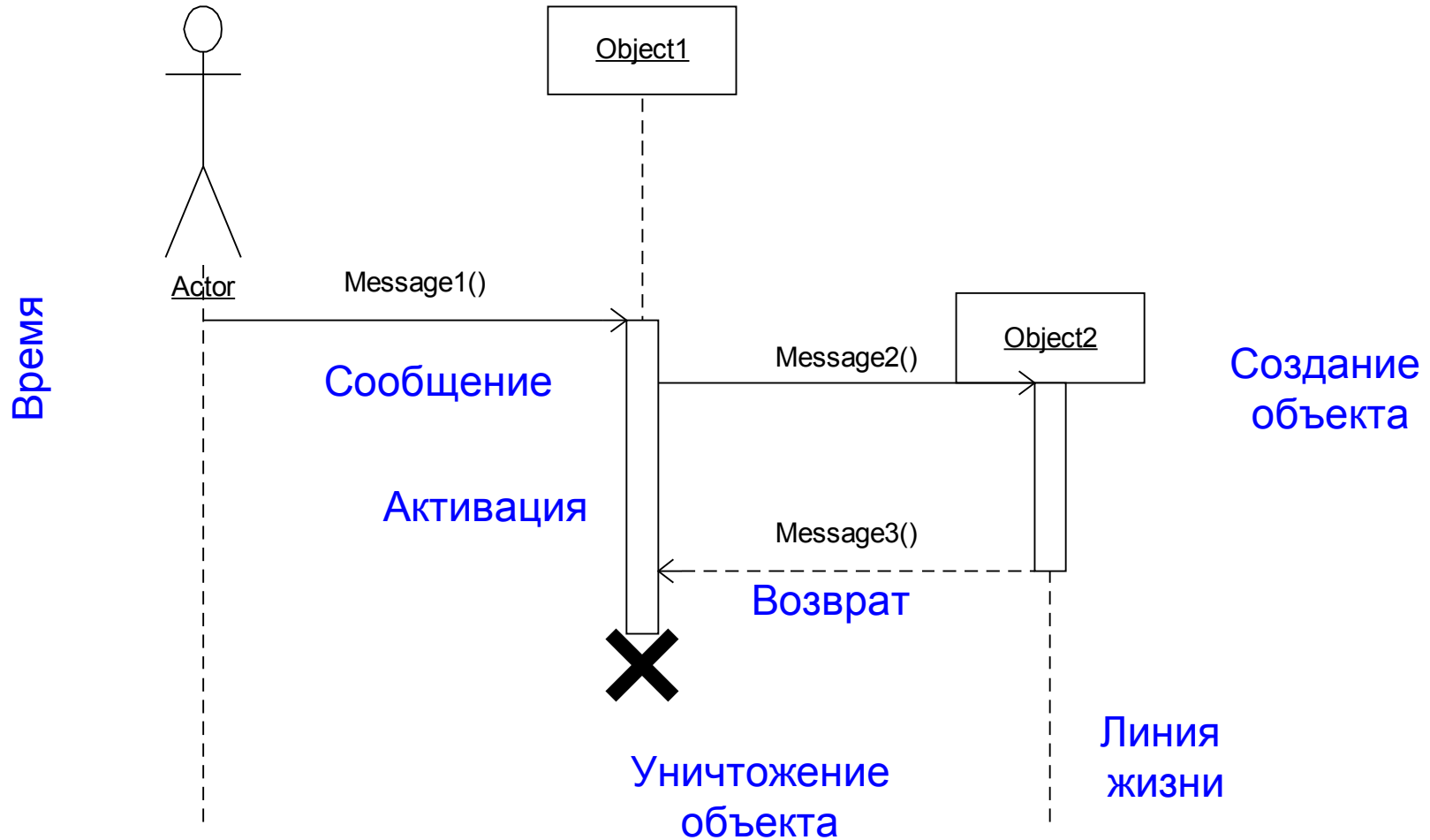
Конец потока управления

Конец потока данных



# Диаграмма последовательности (1.x)

Объект (экземпляр класса)

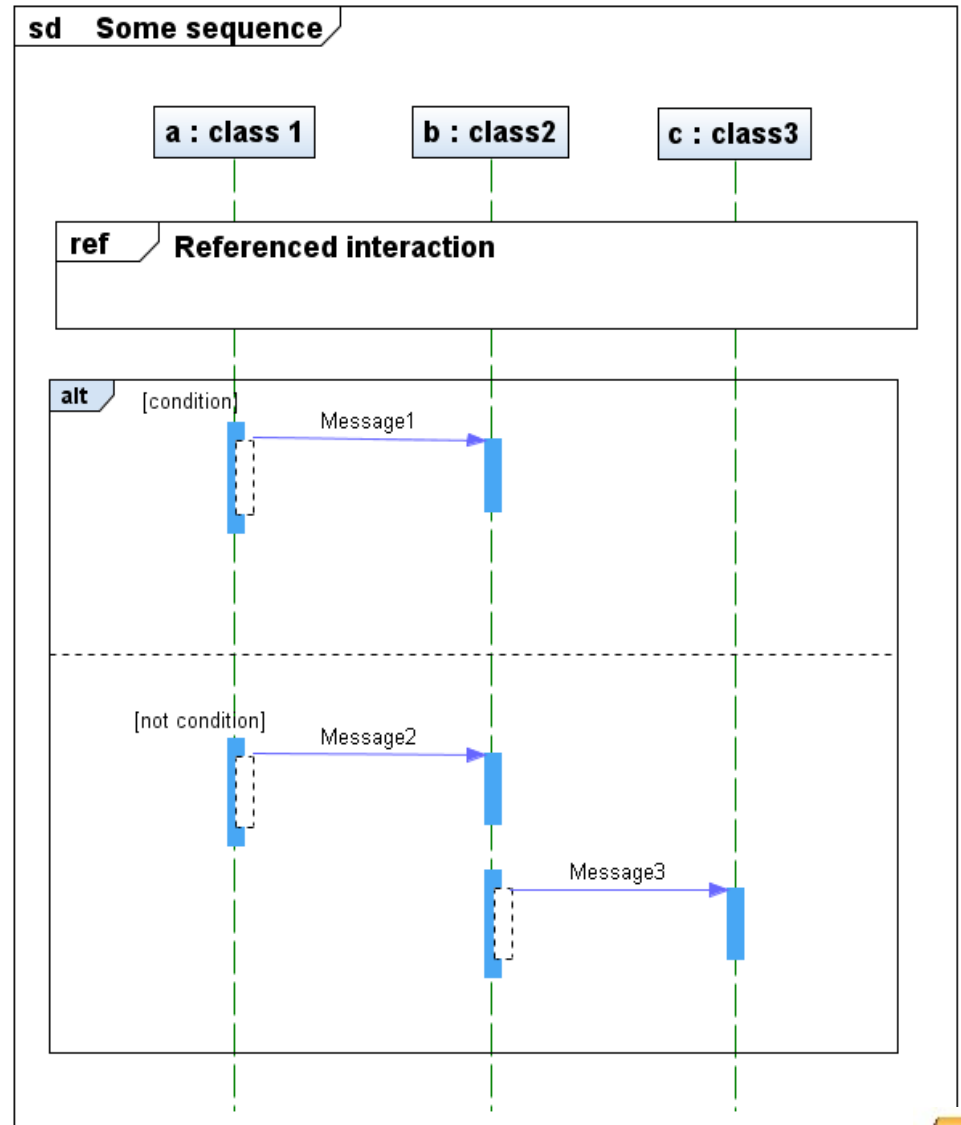


# Диаграмма последовательности (2.0)

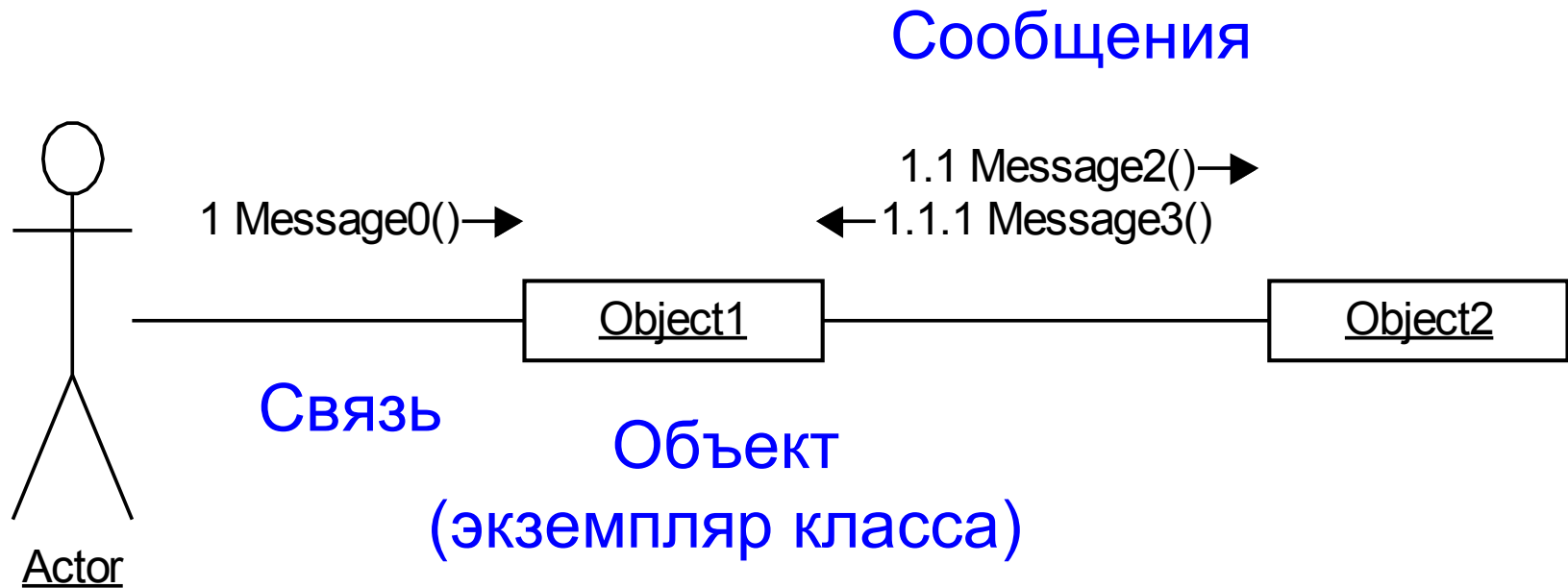
Рамка диаграммы с тегом и именем диаграммы

Ссылка на другую диаграмму

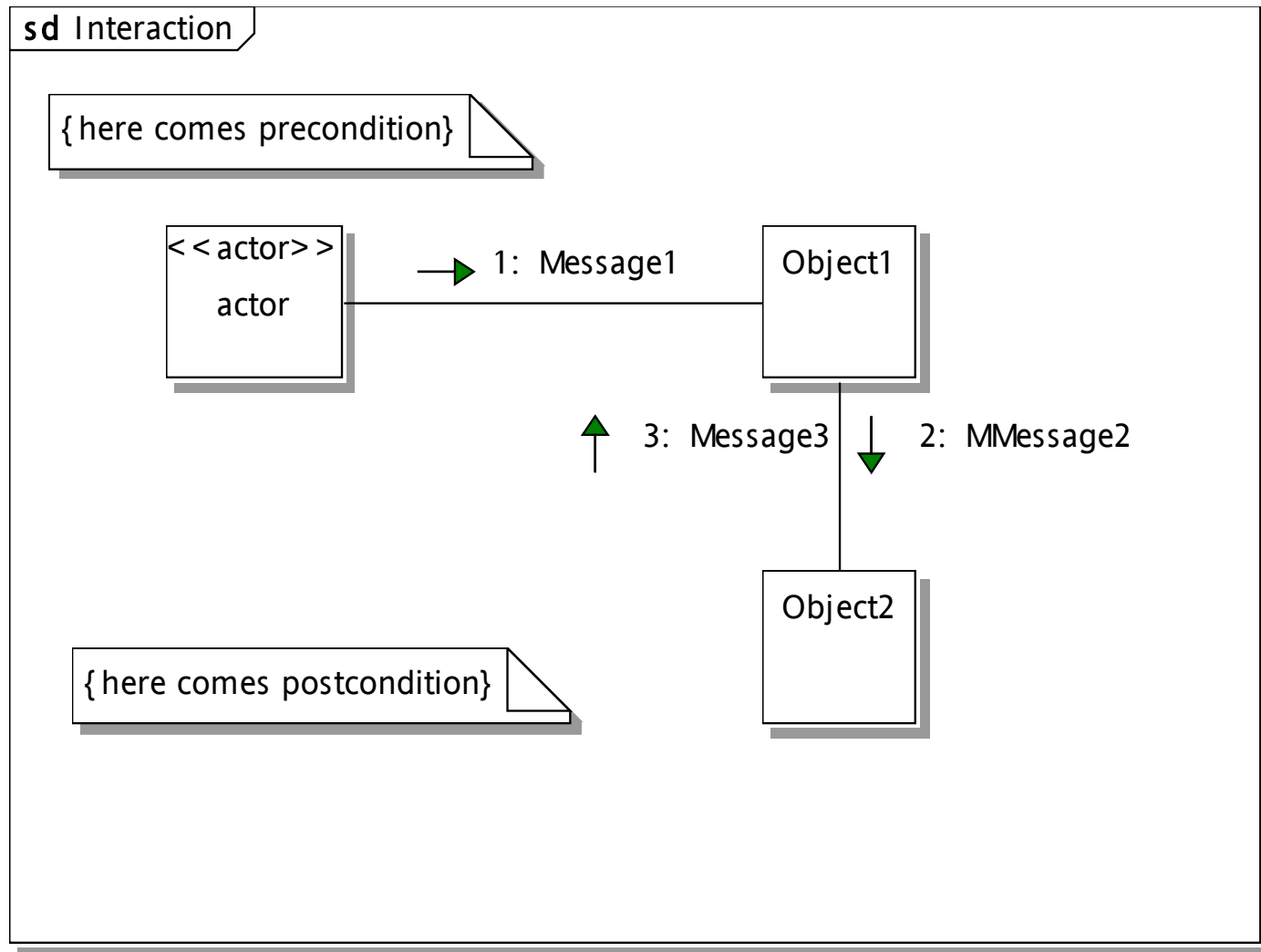
Составной шаг (альтернатива) с рамкой и тегом шага



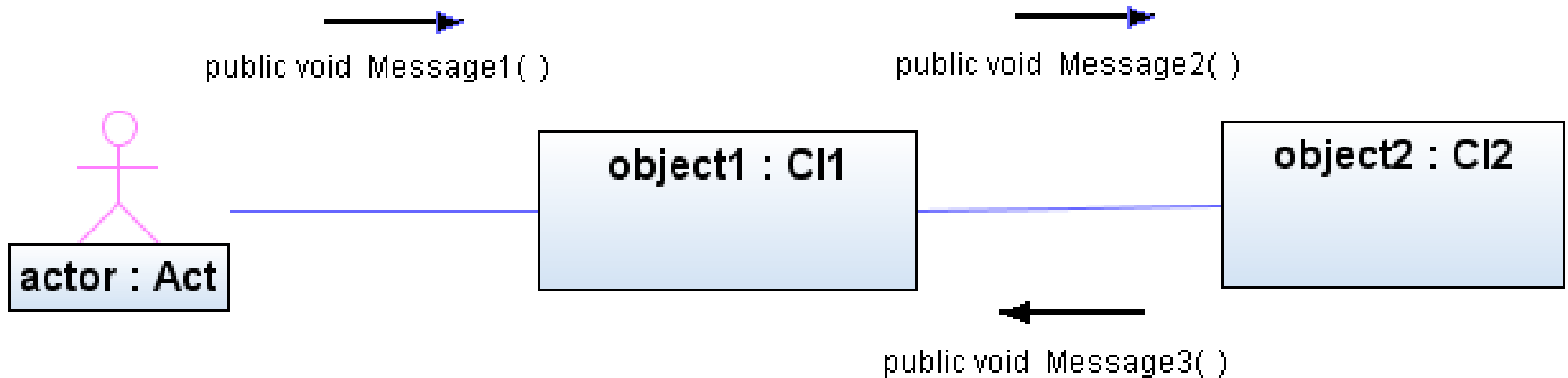
# Диаграмма кооперации (1.x)



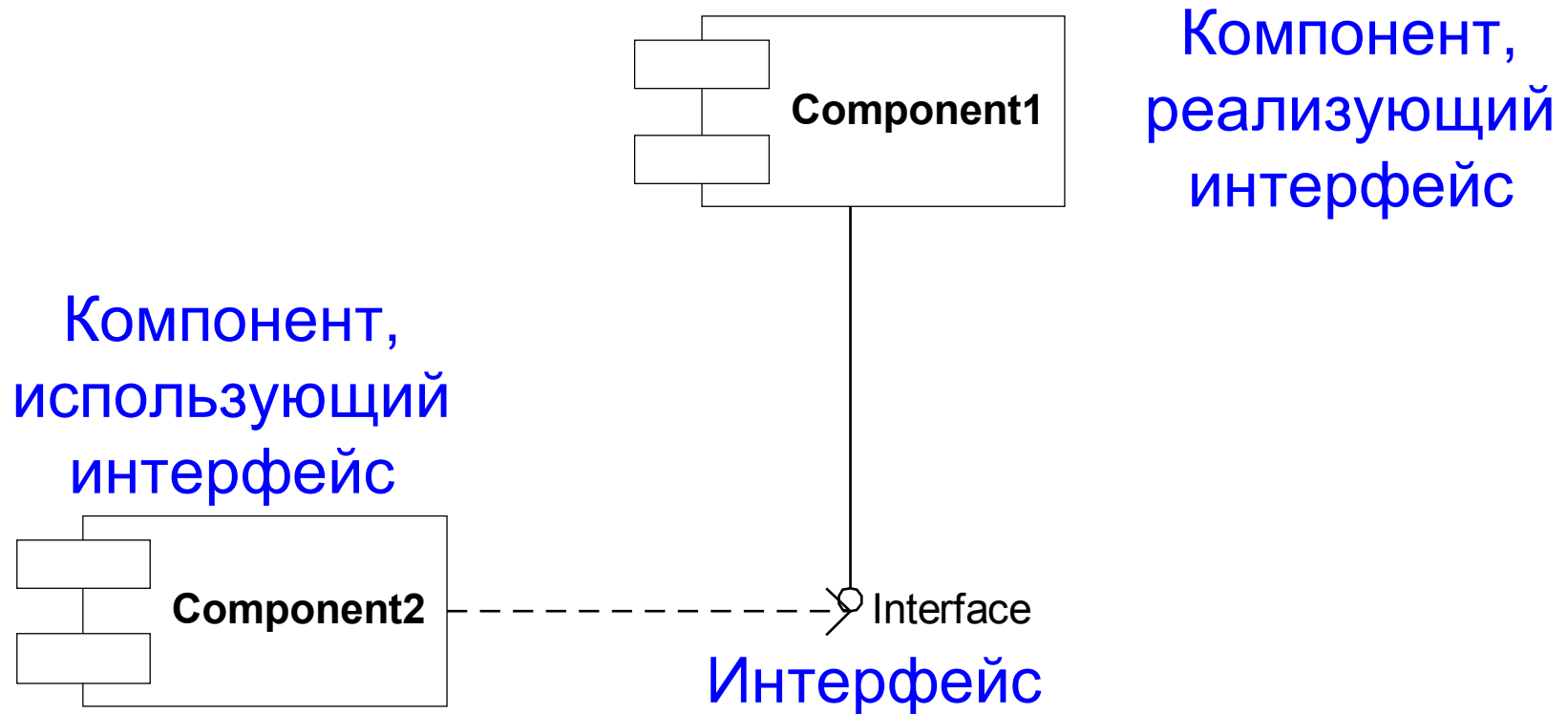
# Диаграмма коммуникации (2.0)



# Диаграмма коммуникации (2.0)

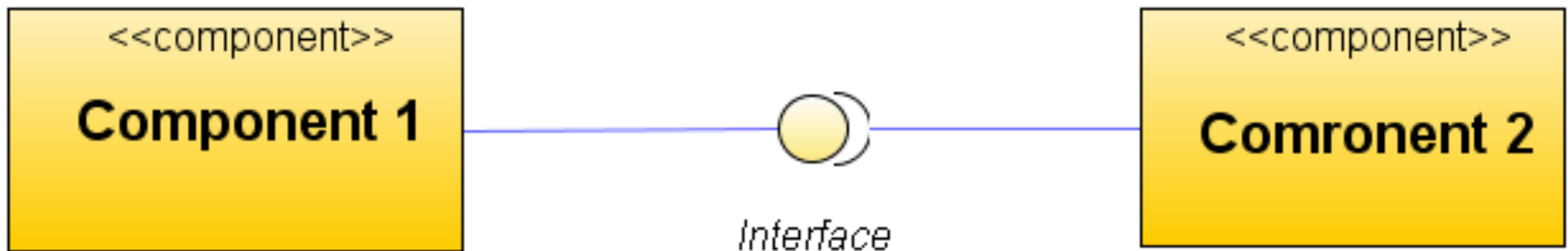


# Диаграмма компонентов (1.x)



# Диаграмма компонентов (2.0)

Предоставляемый  
интерфейс

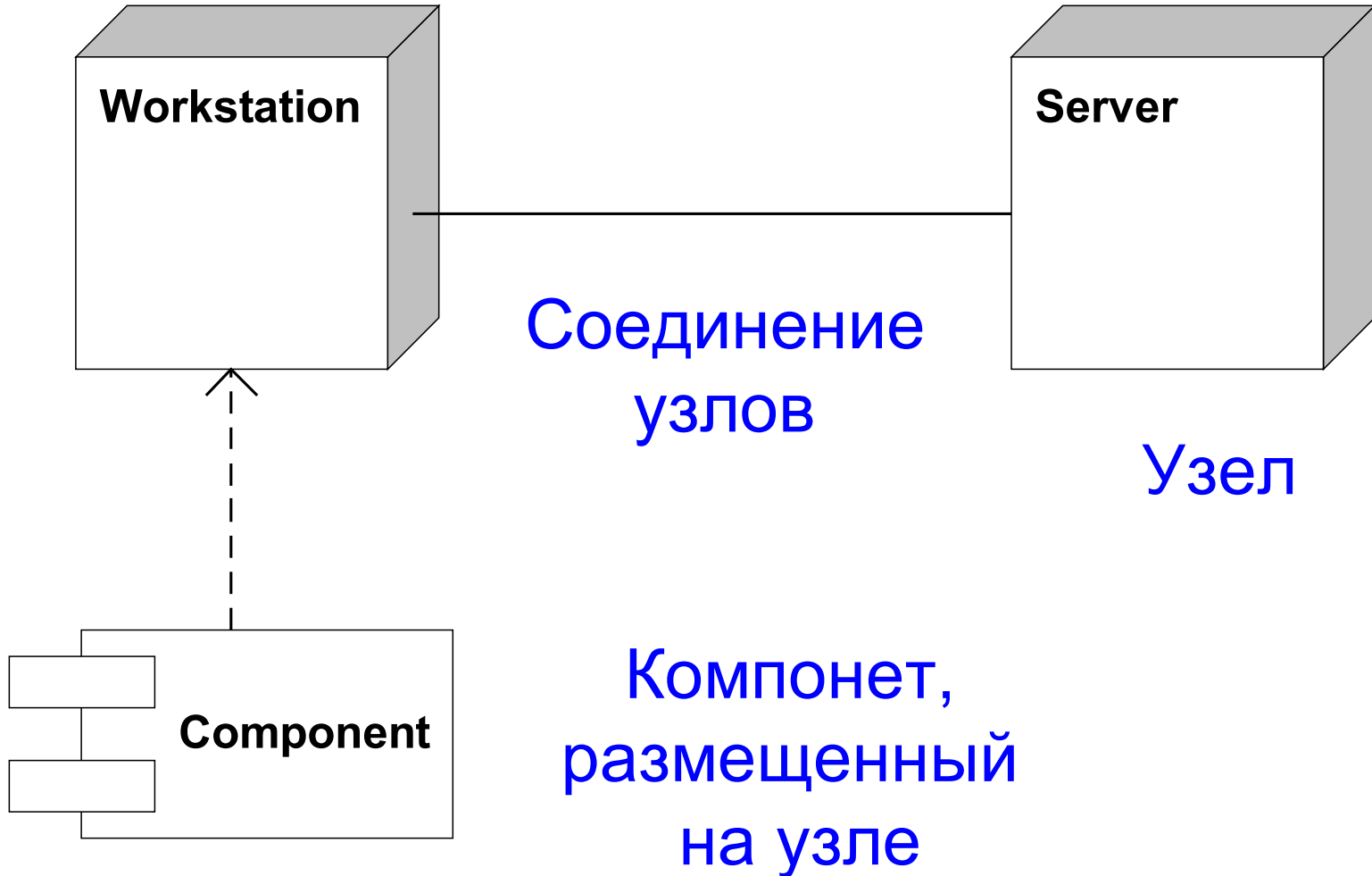


Используемый  
интерфейс



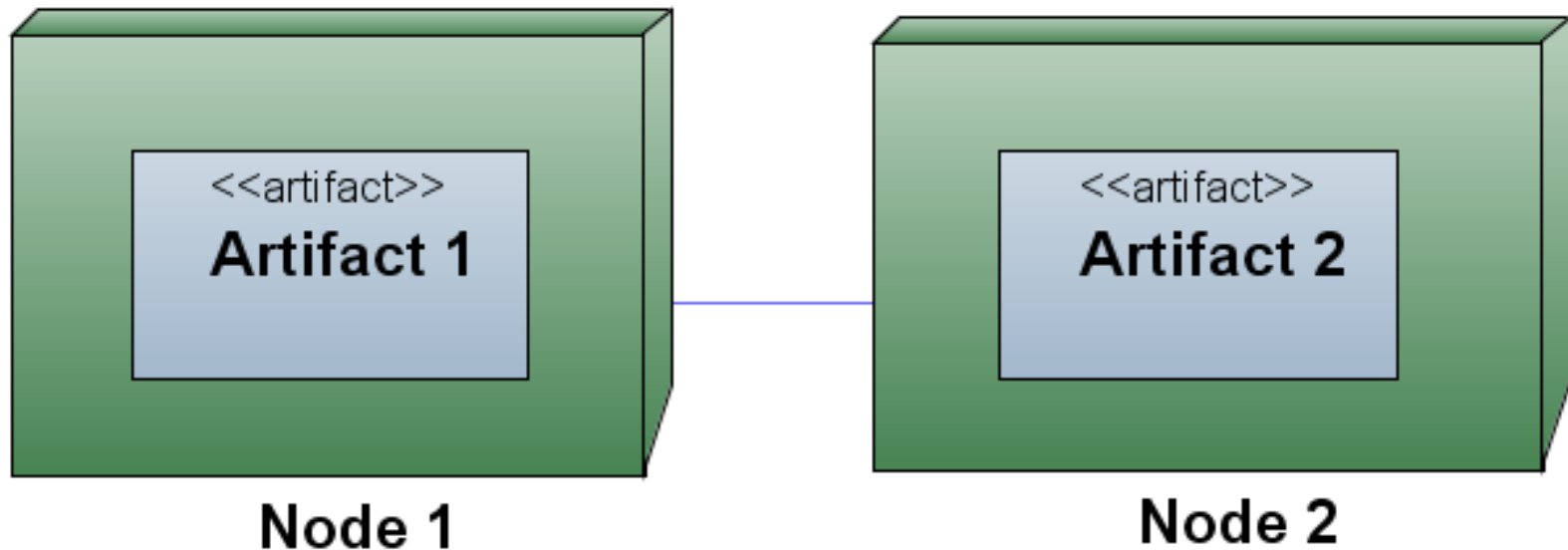


# Диаграмма размещения (1.x)



# Диаграмма размещения (2.0)

Соединение узлов



Компонент,  
размещенный на узле



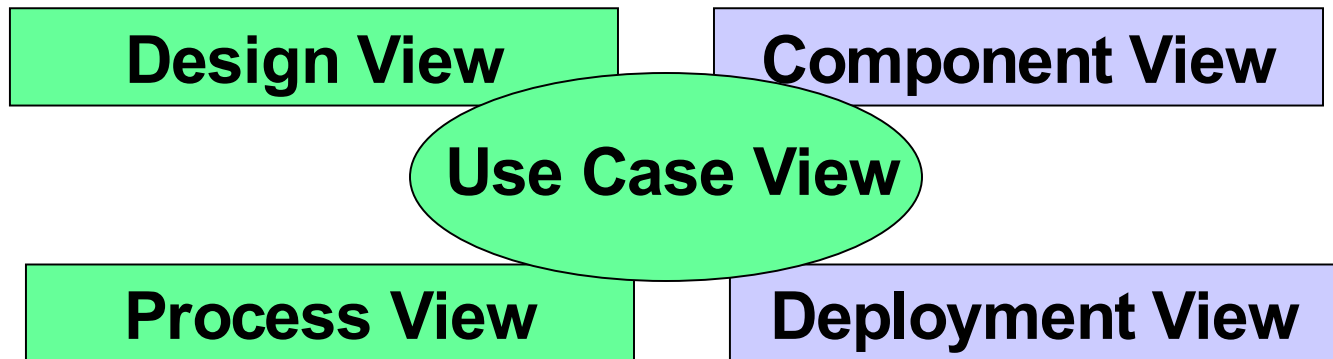
## 2.3. Представления (views)

- Реальные модели сложны
- Сложную модель невозможно обозреть с одной точки зрения
- Представление = проекция (фильтрация) единого графа модели
- 5 (... 8) представлений
- Представление использования (Use Case View)
- Логическое представление (Design View)
- Представление процессов (Process View)
- Представление компонентов (Component View)
- Представление размещения (Deployment View)

# Пять представлений

Словарь предметной  
области,  
функциональность

Сборка системы,  
управление  
конфигурацией



Производительность,  
масштабируемость,  
пропускная способность

Топология системы,  
распределение,  
установка

# Представление и использования

- **Описание поведения с точки зрения внешних агентов**
- **Структурные аспекты**  
– **диаграммы использования**
- **Поведенческие аспекты**  
– **диаграммы взаимодействия, состояний и деятельности**

# Логическое представление

- **Классы**
  - словарь предметной области
- **Структурные аспекты**
  - диаграммы классов и объектов
- **Поведенческие аспекты**
  - диаграммы взаимодействия, состояний и действий

# Представление процессов

- **Параллелизм и синхронизация**
  - производительность, масштабируемость, пропускная способность
- **Структурные аспекты**
  - активные классы: процессы и потоки
- **Поведенческие аспекты**
  - диаграммы взаимодействия, состояний и действий

# Представление компонентов

- **Компоненты (файлы)**
  - управление конфигурацией
- **Структурные аспекты**
  - диаграммы компонентов
- **Поведенческие аспекты**
  - диаграммы взаимодействия, состояний и действий



# Представление размещения

- **Топология аппаратных средств и размещение компонентов**
- **Структурные аспекты**
  - **диаграммы размещения**
- **Поведенческие аспекты**
  - **диаграммы взаимодействия, состояний и действий**

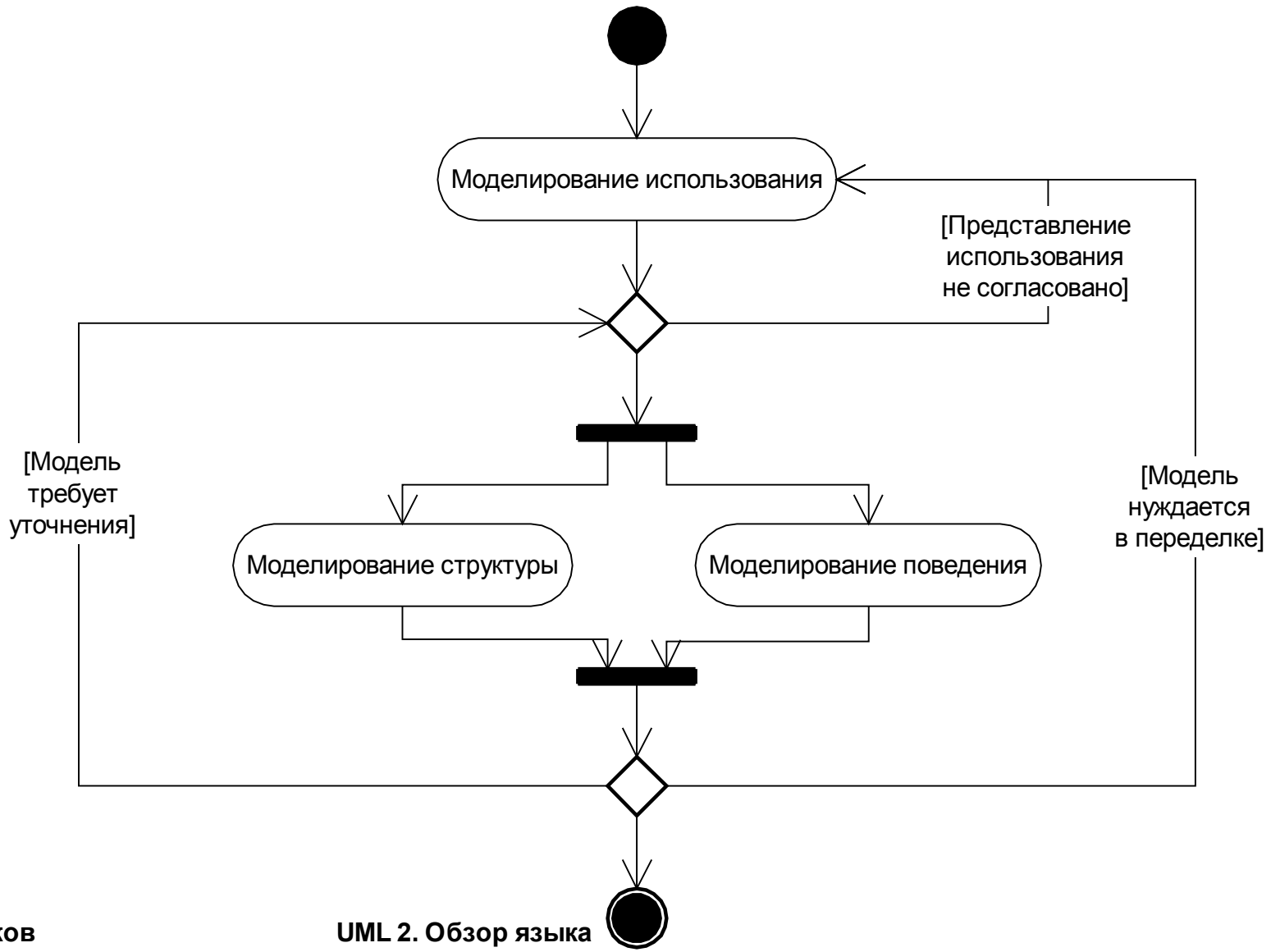
# Восемь представлений

<i>Область</i>	<i>Представления</i>	<i>Диаграммы</i>
Структурная	Статическое	Классов
	Проектирования	Внутренняя структура Кооперации Компонентов
	Использования	Использования
Динамическая	Конечных автоматов	Состояний
	Деятельности	Деятельности
	Взаимодействия	Последовательности Коммуникации
Физическая	Размещения	Размещения
Управления моделью	Управления моделью	Пакетов

# Три представления

- **Представление использования**
  - ЧТО делает система
  - Диаграммы использования
- **Представление структуры**
  - ИЗ ЧЕГО состоит система
  - Диаграммы классов, компонентов и размещения
- **Представление поведения**
  - КАК работает система
  - Диаграммы состояний, деятельности и взаимодействия (последовательности / коммуникации)

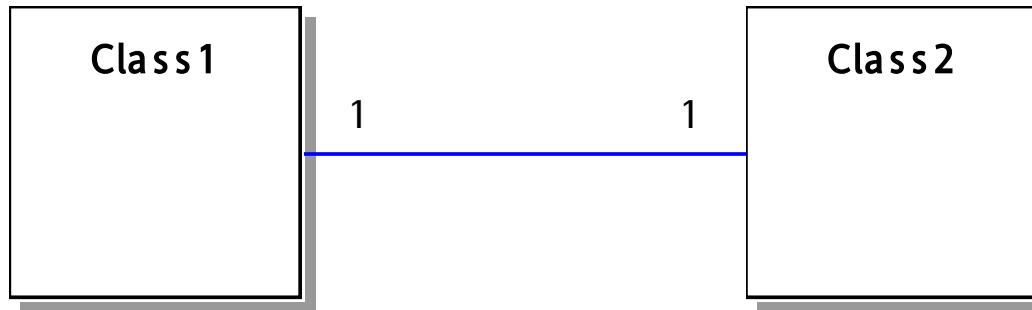
# Итеративный процесс моделирования



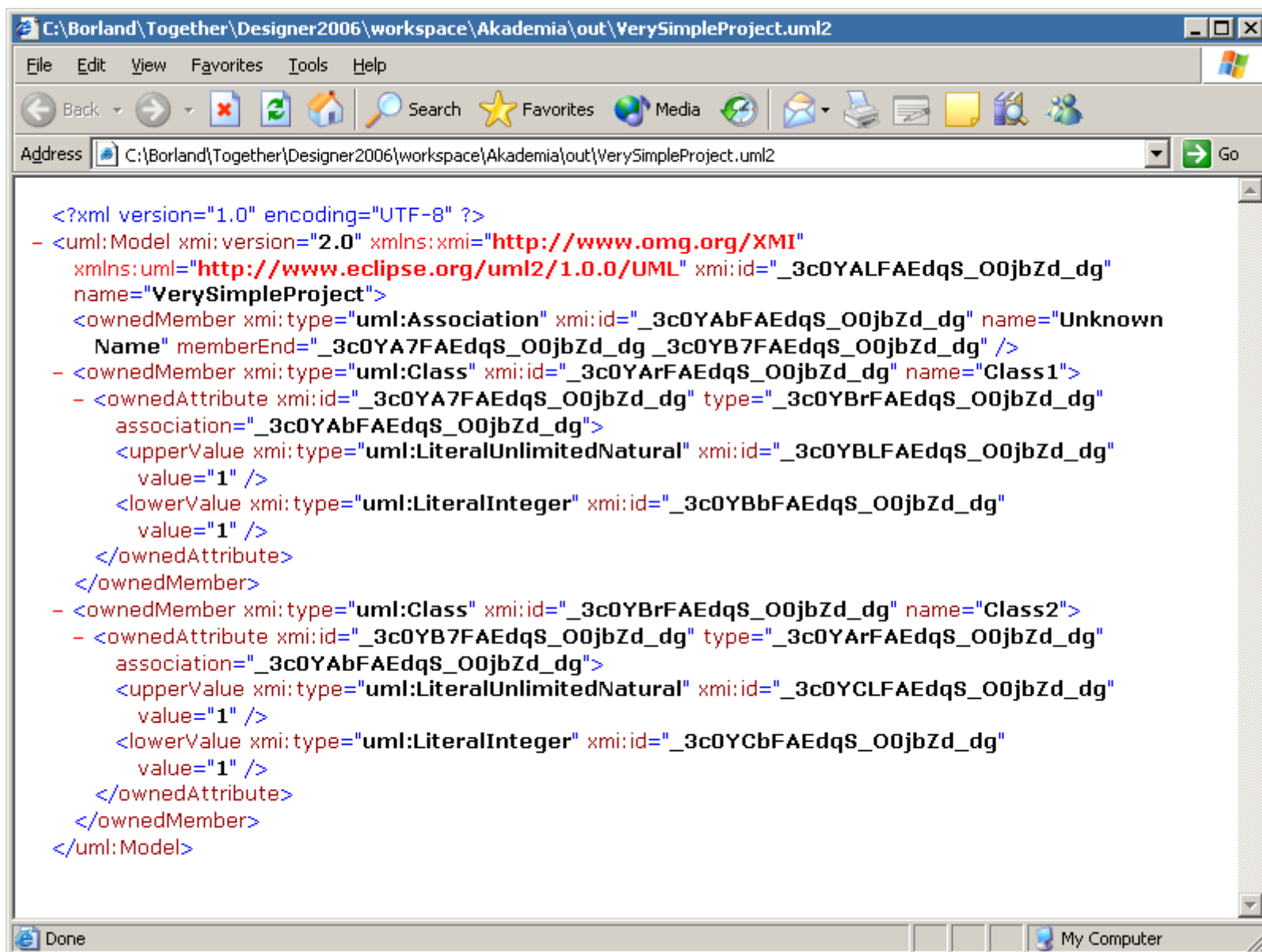
## 2.4. Общие механизмы

- Внутреннее представление = Спецификации (specifications)
  - Обратная сторона картинки
  - XMI
- Дополнения (adornments) → Украшения (decorations)
  - Доп. разделы, спец. значки
- Подразделения (common divisions)
  - Класс – объект
  - *Интерфейс* - реализация
- Механизмы расширения (extension mechanisms)
  - Стереотипы (stereotypes)
  - Именованные значения (tagged values)
  - Ограничения (constraints)

# Внутреннее представление и XMI



# Внутреннее представление и XMI



The screenshot shows a text editor window titled "C:\Borland\Together\Designer2006\workspace\Akademia\out\VerySimpleProject.uml2". The window contains XMI (XML Metadata Interchange) code for a UML model. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <uml:Model xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
  xmlns:uml="http://www.eclipse.org/uml2/1.0.0/UML" xmi:id="_3c0YALFAEdqS_00jbZd_dg"
  name="VerySimpleProject">
  <ownedMember xmi:type="uml:Association" xmi:id="_3c0YAbFAEdqS_00jbZd_dg" name="Unknown
    Name" memberEnd="_3c0YA7FAEdqS_00jbZd_dg _3c0YB7FAEdqS_00jbZd_dg" />
- <ownedMember xmi:type="uml:Class" xmi:id="_3c0YArFAEdqS_00jbZd_dg" name="Class1">
  - <ownedAttribute xmi:id="_3c0YA7FAEdqS_00jbZd_dg" type="_3c0YBrFAEdqS_00jbZd_dg"
    association="_3c0YAbFAEdqS_00jbZd_dg">
    <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_3c0YBLFAEdqS_00jbZd_dg"
      value="1" />
    <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_3c0YBbFAEdqS_00jbZd_dg"
      value="1" />
    </ownedAttribute>
  </ownedMember>
- <ownedMember xmi:type="uml:Class" xmi:id="_3c0YBrFAEdqS_00jbZd_dg" name="Class2">
  - <ownedAttribute xmi:id="_3c0YB7FAEdqS_00jbZd_dg" type="_3c0YArFAEdqS_00jbZd_dg"
    association="_3c0YAbFAEdqS_00jbZd_dg">
    <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_3c0YCLFAEdqS_00jbZd_dg"
      value="1" />
    <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_3c0YCbFAEdqS_00jbZd_dg"
      value="1" />
    </ownedAttribute>
  </ownedMember>
</uml:Model>
```

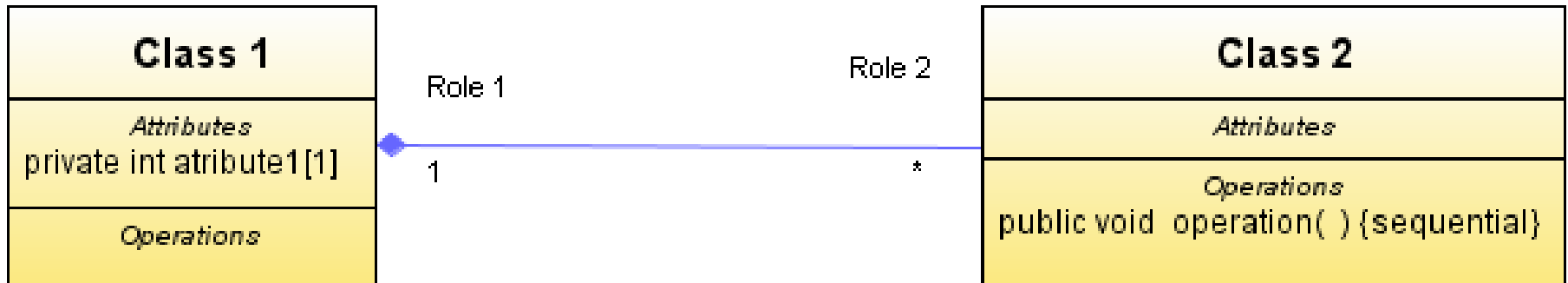
# Украшения

Класс

Ассоциация



Класс с  
дополнениями



Ассоциация с дополнениями

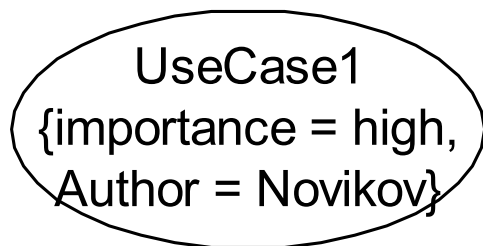




# Подразделения

- Если в одном и том же контексте нужно различать множества (классы) и элементы (объекты), то имена объектов подчеркиваются, а множеств – нет
- *Абстрактный классификатор* = не может иметь непосредственных экземпляров (потомки могут)
- Конкретный классификатор = может иметь непосредственные экземпляры
- Если в одном и том же контексте нужно различать абстрактные и конкретные классификаторы, то имена абстрактных записываются *курсивом*, а конкретных – прямым шрифтом

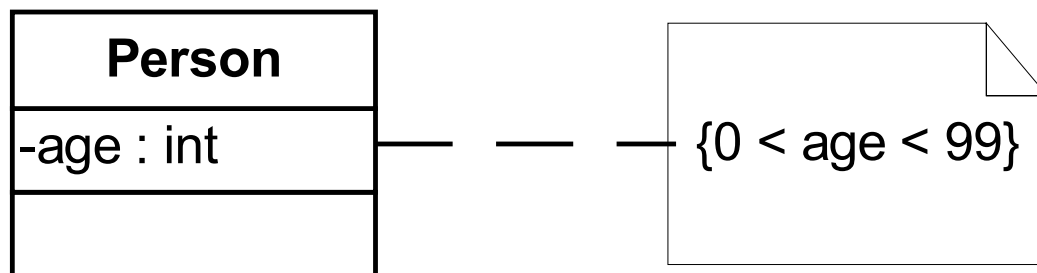
# Именованные значения и ограничения



Вариант использования  
с помеченными значениями

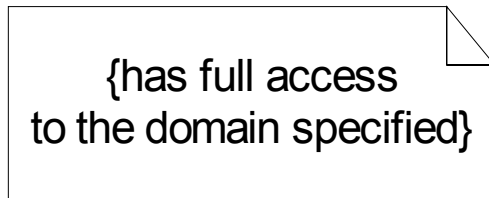
Класс с  
атрибутом

Ограничение  
на значение  
атрибута

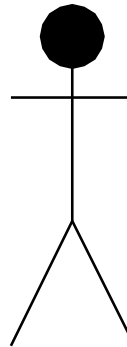


# Стереотипы

Ограничение  
стереотипа



Измененный  
значок  
стереотипа



«PowerUser»  
Administrator  
{domain = LAN}

Имя стереотипа

Помеченное  
значение  
стереотипа



## 2.5. Профили

- **Проблемно ориентированные языки**
  - Domain (problem) specific (oriented) languages
- **Тяжеловесное «расширение» – изменение метамодели (мета моделирование)**
  - + Нет ограничений
  - – Потеря совместимости (interoperability)
- **Легковесное «расширение» – добавление стереотипов (профилирование)**
  - + Полная совместимость
  - – Нельзя изменить сам язык

## 2.6. Общие свойства модели

- **Правильность (correctness)**
  - Проверяет инструмент
- **Непротиворечивость (well formed)**
  - Проверяем мы
- **Полнота (completeness)**
  - Недостижима и не нужна
- **Семантические вариации (semantic variation points)**
  - Статическая или динамическая классификация? Зарыто глубоко

# Выводы

- **Модель UML состоит из описания сущностей и отношений между ними**
- **Элементы модели группируются в диаграммы и представления для наилучшего описания моделируемой системы с различных точек зрения**
- **В случае необходимости элементы UML могут быть расширены и переопределены средствами самого языка**