

# **Лекция 2. Унифицированный язык визуального моделирования UML**

## *Учебные вопросы:*

- 1. Понятие UML.**
- 2. Предметы UML.**
- 3. Отношения UML.**
- 4. Диаграммы UML.**
- 5. Механизмы расширения UML.**

**Литература: [4], [6], [10], [16].**

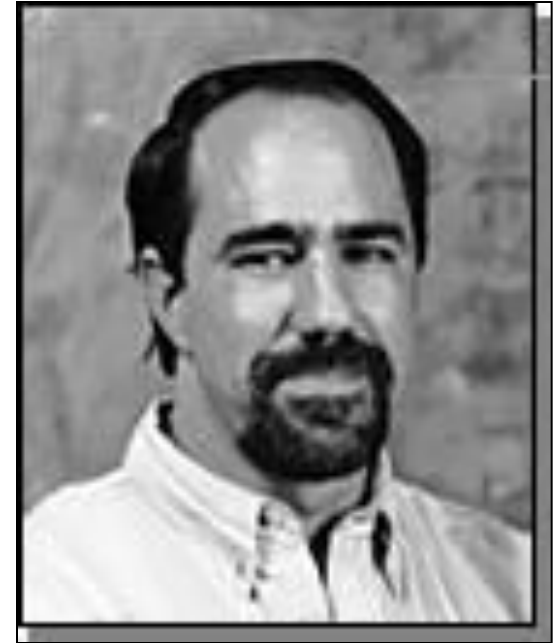
# Создатели UML



**Ивар Джакобсон**



**Джим Рэмбо**



**Грэди Буч**

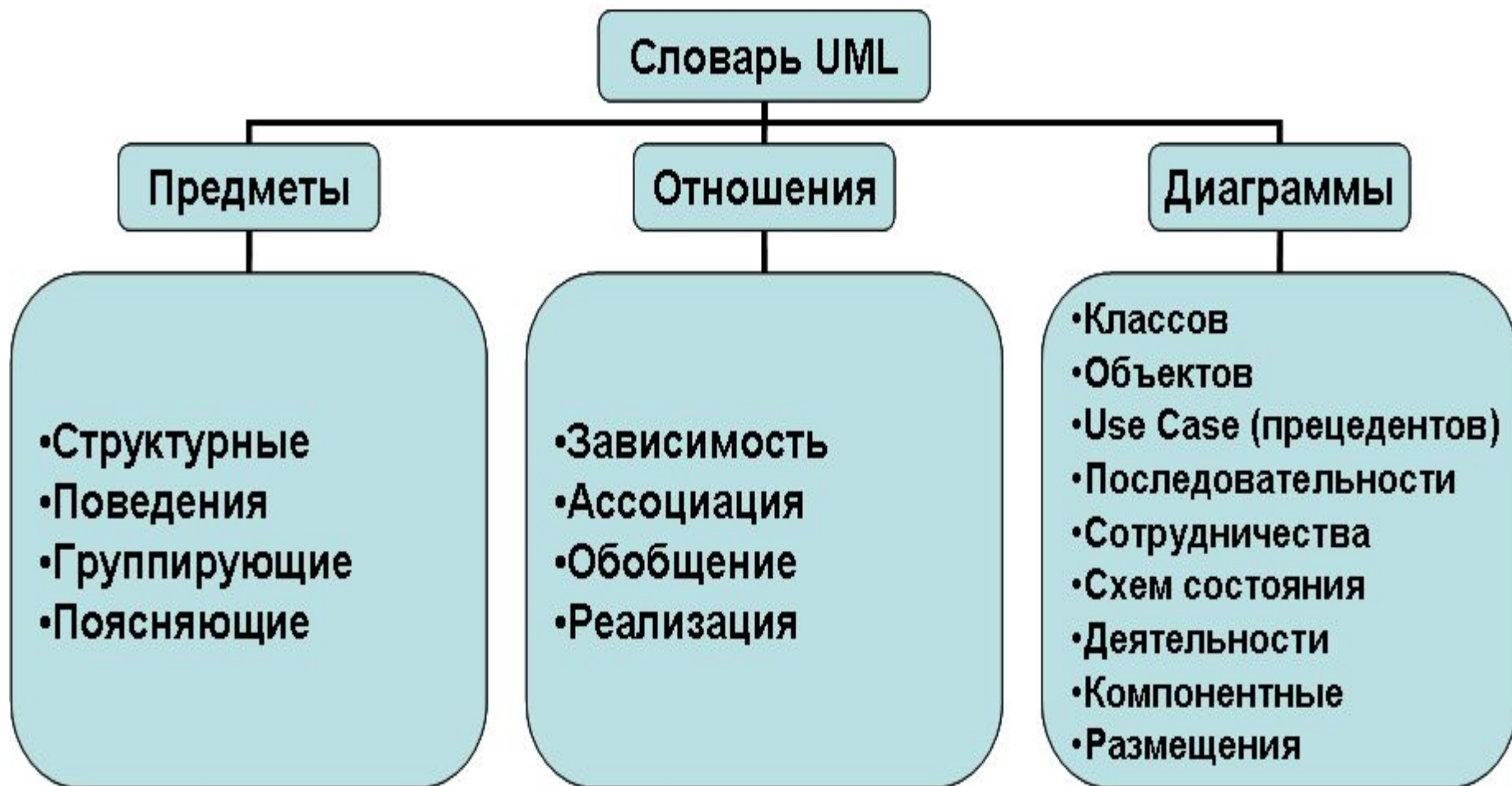
# Понятие UML

UML – это язык для определения, визуализации, конструирования и документирования артефактов программных систем, а также для моделирования экономических процессов и других не программных систем.

UML обладает следующими основными характеристиками:

- является языком визуального моделирования, который обеспечивает разработку репрезентативных моделей для организации взаимодействия заказчика и разработчика ИС, различных групп разработчиков;
- содержит механизмы расширения и специализации базовых концепций языка.

# Словарь UML

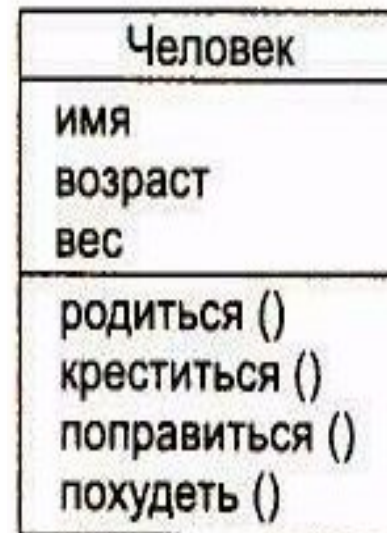


# Структурные предметы

- **Класс**
- **Интерфейс**
- **Кооперация**
- **Актер**
- **Элемент Use Case (Прецедент)**
- **Активный класс**
- **Компонент**
- **Узел**

**Класс реализует один или несколько интерфейсов**

**Графически класс отображается в виде прямоугольника, обычно включающего секции с именем, свойствами (атрибутами) и операциями**



# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актер
- Элемент Use Case (Прецедент)
- Активный класс
- Компонент
- Узел

Интерфейс описывает поведение элемента, видимое извне

Интерфейс может представлять полные услуги класса или компонента или часть таких услуг

Графически интерфейс изображается в виде кружка с именем

Имя интерфейса обычно начинается с буквы «I»



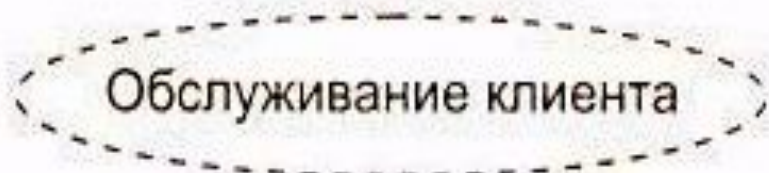
# Структурные предметы

- Класс
- Интерфейс
- *Кооперация*
- Актер
- Элемент Use Case (Прецедент)
- Активный класс
- Компонент
- Узел

Кооперации имеют как структурное, так и поведенческое измерения

Конкретный класс может участвовать в нескольких кооперациях

Графически кооперация изображается как пунктирный эллипс, в который вписывается ее имя



Обслуживание клиента

# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актёр
- Элемент Use Case (Прецедент)
- Активный класс
- Компонент
- Узел

Каждая роль требует от системы определенного поведения

Изображается как проволочный человечек с именем





# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актер
- Элемент Use Case  
(Прецедент)
- Активный класс
- Компонент
- Узел

В модели элемент Use Case применяется для структурирования предметов поведения

Элемент Use Case реализуется кооперацией

Изображается как эллипс, в который вписывается его имя

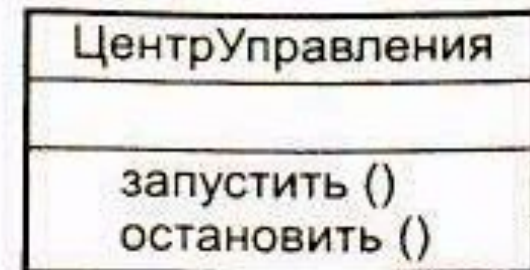


# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актер
- Элемент Use Case (Прецедент)
- Активный класс
- Компонент
- Узел

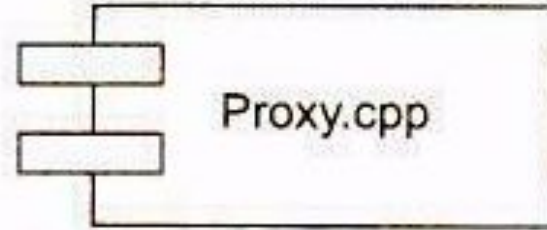
Похож на обычный класс за исключением того, что его объекты действуют одновременно с объектами других классов

Изображается как активный прямоугольник, обычно включающий имя, свойства (атрибуты) и операции



# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актер
- Элемент Use Case (Прецедент)
- Активный класс
- **Компонент**
- Узел



Обычно компонент – это физическая упаковка различных логических элементов (классов, интерфейсов и содружеств)

Изображается как прямоугольник с вкладками, обычно включающий имя

# Структурные предметы

- Класс
- Интерфейс
- Кооперация
- Актер
- Элемент Use Case (Прецедент)
- Активный класс
- Компонент
- Узел



В узле размещается набор компонентов, который может перемещаться от узла к узлу

Изображается как куб с именем

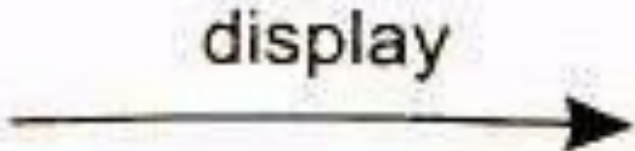
# Предметы поведения

- *Взаимодействие*
- **Конечный автомат**

**Взаимодействие может определять динамику как совокупности объектов, так и отдельной операции**

**Элементами взаимодействия являются сообщения, последовательность действий (поведение, вызываемое сообщением) и связи (соединения между объектами)**

**Сообщение изображается в виде направленной линии с именем ее операции**



display

The diagram shows a horizontal arrow pointing to the right. Above the arrow, the word "display" is written in a simple, sans-serif font.

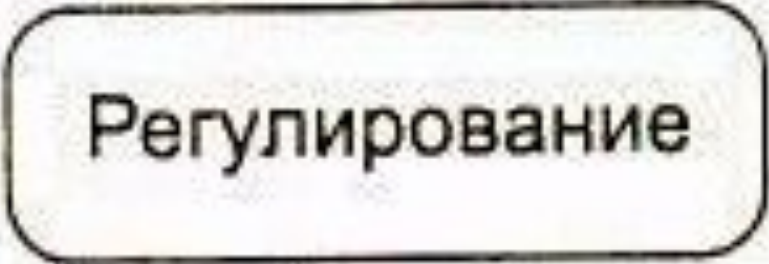
# Предметы поведения

- **Взаимодействие**
- **Конечный автомат**

С помощью конечного автомата может определяться поведение индивидуального класса или кооперации классов

Элементами конечного автомата являются состояния, переходы (от состояния к состоянию), события (предметы, вызывающие переходы) и действия (реакции на переход)

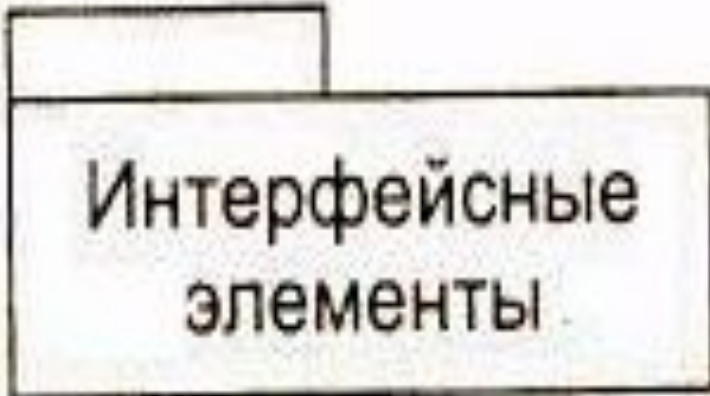
Изображается как закругленный прямоугольник, обычно включающий его имя и его подсостояния (если они есть)



Регулирование

# Группирующие предметы

- *Пакет*



**В пакет могут помещаться структурные предметы, предметы поведения и даже другие группировки предметов**

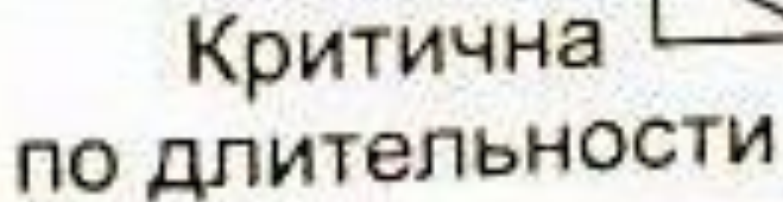
**Пакет – это чисто концептуальное понятие и существует только в период разработки**

**Изображается как папка с закладкой, на которой обозначено его имя и, иногда, его содержание**

# Поясняющие предметы

- Примечание

Изображается в виде прямоугольника с загнутым углом, в который вписывается текстовый или графический комментарий



Критична  
по длительности



# Отношения UML

- *зависимость*
- ассоциация
- обобщение
- реализация

Изображается в виде пунктирной линии, возможно направленной на независимый предмет и иногда имеющей метку



# Отношения UML

- **зависимость**
- **ассоциация**
- **обобщение**
- **реализация**



Агрегация – это специальная разновидность ассоциации, представляющая структурное отношение между целым и его частями

Изображается в виде сплошной линии, возможно направленной, иногда имеющей метку и часто включающей другие «украшения», такие как мощность и имена ролей

# Отношения UML

- **зависимость**
- **ассоциация**
- **обобщение**
- **реализация**

Потомок разделяет  
структуру и поведение  
родителя

Изображается в виде  
сплошной стрелки с полым  
наконечником,  
указывающим на родителя

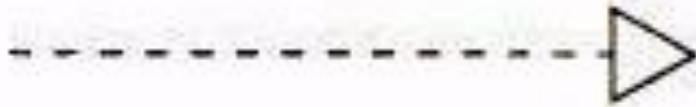


# Отношения UML

- **зависимость**
- **ассоциация**
- **обобщение**
- **реализация**

Отношения реализации применяют в двух случаях: между интерфейсами и классами (или компонентами), реализующими их; между элементами Use Case и кооперациями, которые реализуют их

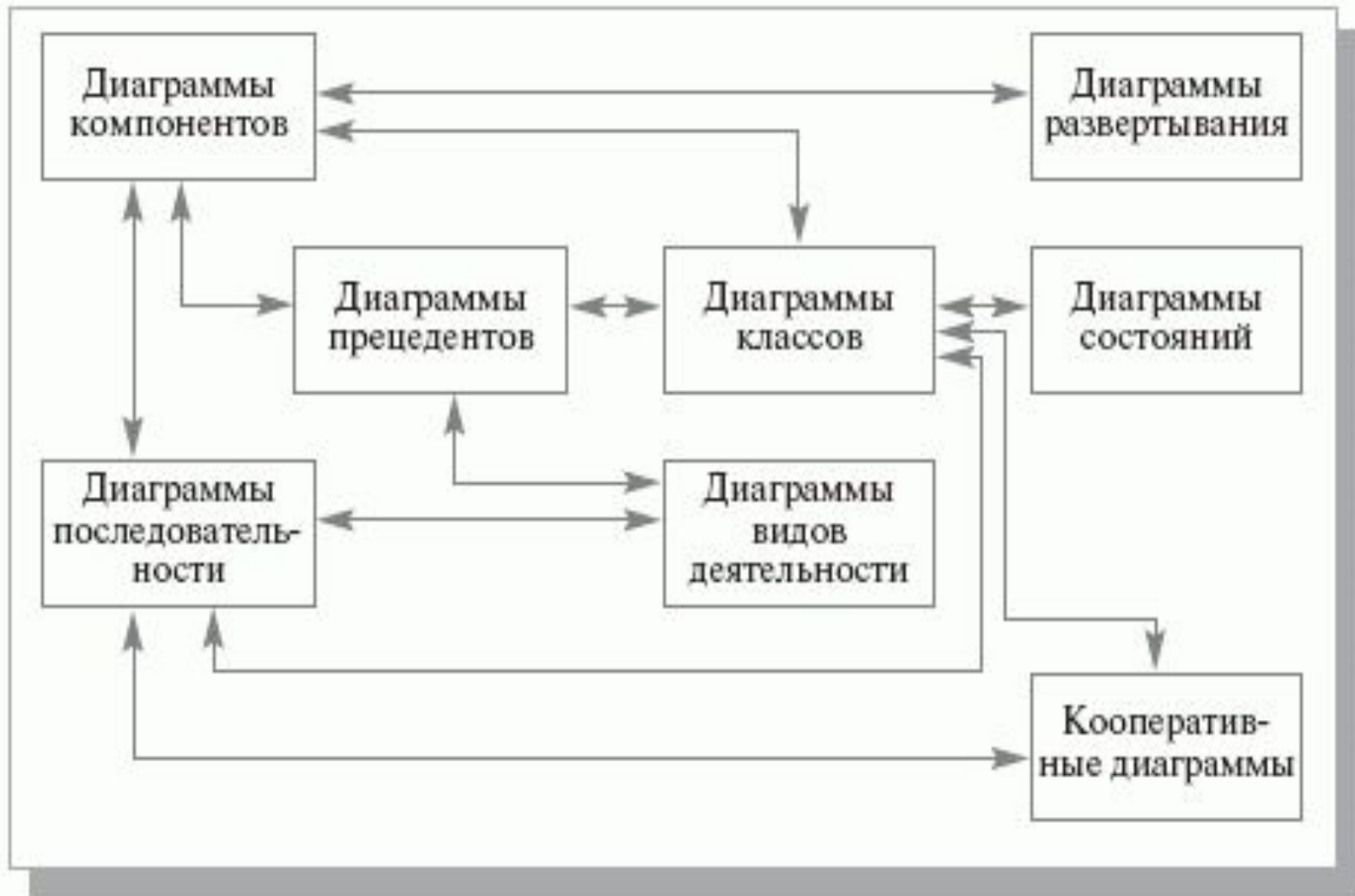
Изображается как нечто среднее между обобщением и зависимостью



# Диаграммы UML

- **Диаграмма классов** – показывает набор классов, интерфейсов, содружеств и их отношений. Обеспечивает статическое проектное представление системы.
- **Диаграмма объектов** – показывает набор объектов и их отношения. Диаграмма объектов представляет статический «моментальный снимок» с экземпляров предметов, которые находятся в диаграммах классов.
- **Диаграмма Use Case** (диаграмма прецедентов) – показывает набор элементов Use Case, актеров и их отношений. Обеспечивает статическое представление Use Case. Используется при организации и моделировании поведения системы, задании требований заказчика к системе.
- **Диаграмма взаимодействия** – показывает взаимодействие, включающее набор объектов, их отношений, а также пересылаемые между объектами сообщения. Обеспечивает динамическое представление системы.
- **Диаграмма последовательности** – это диаграмма взаимодействия, которая выделяет упорядочение сообщений по времени.
- **Диаграмма сотрудничества** (диаграмма кооперации) – это диаграмма взаимодействия, которая выделяет структурную организацию объектов, посылающих и принимающих сообщения.
- **Диаграмма схем состояний** – показывает конечный автомат, представляет состояния, переходы, события и действия. Обеспечивает динамическое представление системы.
- **Диаграмма деятельности** – специальная разновидность диаграммы схем состояний, которая показывает поток от действия к действию внутри системы. Обеспечивает динамическое представление системы. Используется при моделировании функциональности системы и выделяет поток управления между объектами.
- **Компонентная диаграмма** – показывает организацию набора компонентов и зависимости между компонентами. Компонентные диаграммы обеспечивают статическое представление реализации системы.
- **Диаграмма размещения** (диаграмма развертывания) – показывает конфигурацию обрабатывающих узлов периода выполнения, а также компоненты, живущие в них. Обеспечивает статическое представление размещения системы.

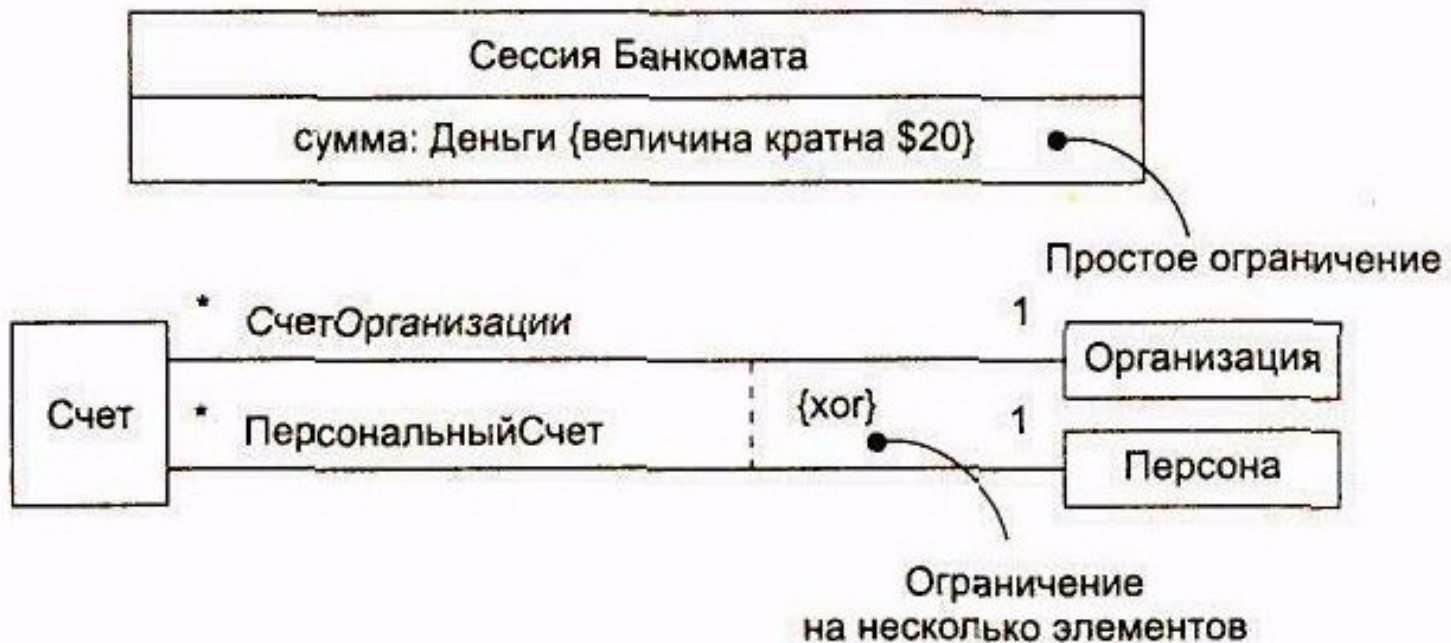
# Взаимосвязи между диаграммами UML



# Механизмы расширения UML

- ограничение
- **теговая величина**
- **стереотип**

Ограничение показывают как текстовую строку, заключенную в фигурные скобки { }



# Механизмы расширения UML

- ограничение
- теговая величина
- стереотип

Теговую величину  
показывают как строку в  
фигурных скобках { }

Строка имеет вид:  
*имя теговой величины =  
значение*

ТекстовыйПроцессор  
{версия = 4.2, автор = Бендер}



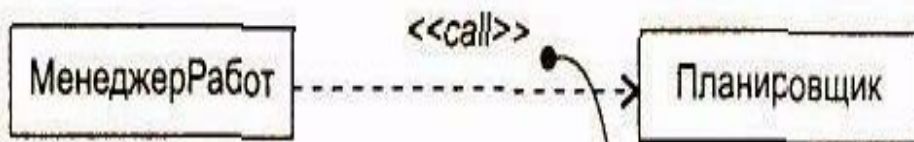
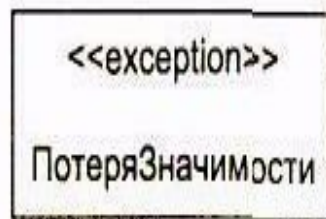
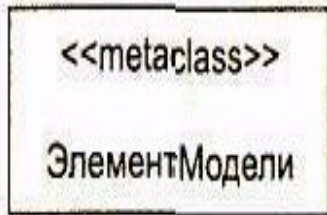
# Механизмы расширения UML

- ограничение
- теговая величина
- стереотип

Элемент со стереотипом является вариацией существующего элемента, имеющей такую же форму, но отличающуюся по сути

У него могут быть дополнительные ограничения и теговые величины, а также другое визуальное представление

Отображают стереотип как имя, указываемое в двойных угловых скобках (или в угловых кавычках)



Исходная операция вызывает целевую