

# Анализ и проектирование на UML

Направление подготовки  
“Информационные системы и технологии”

Максим Валерьевич Хлопотов,  
старший преподаватель кафедры ИС

# Темы лекционных занятий

1. Введение в UML
2. Моделирование использования
3. Моделирование структуры
- 4. Моделирование поведения**
5. Дисциплина моделирования
6. Примеры моделей на UML

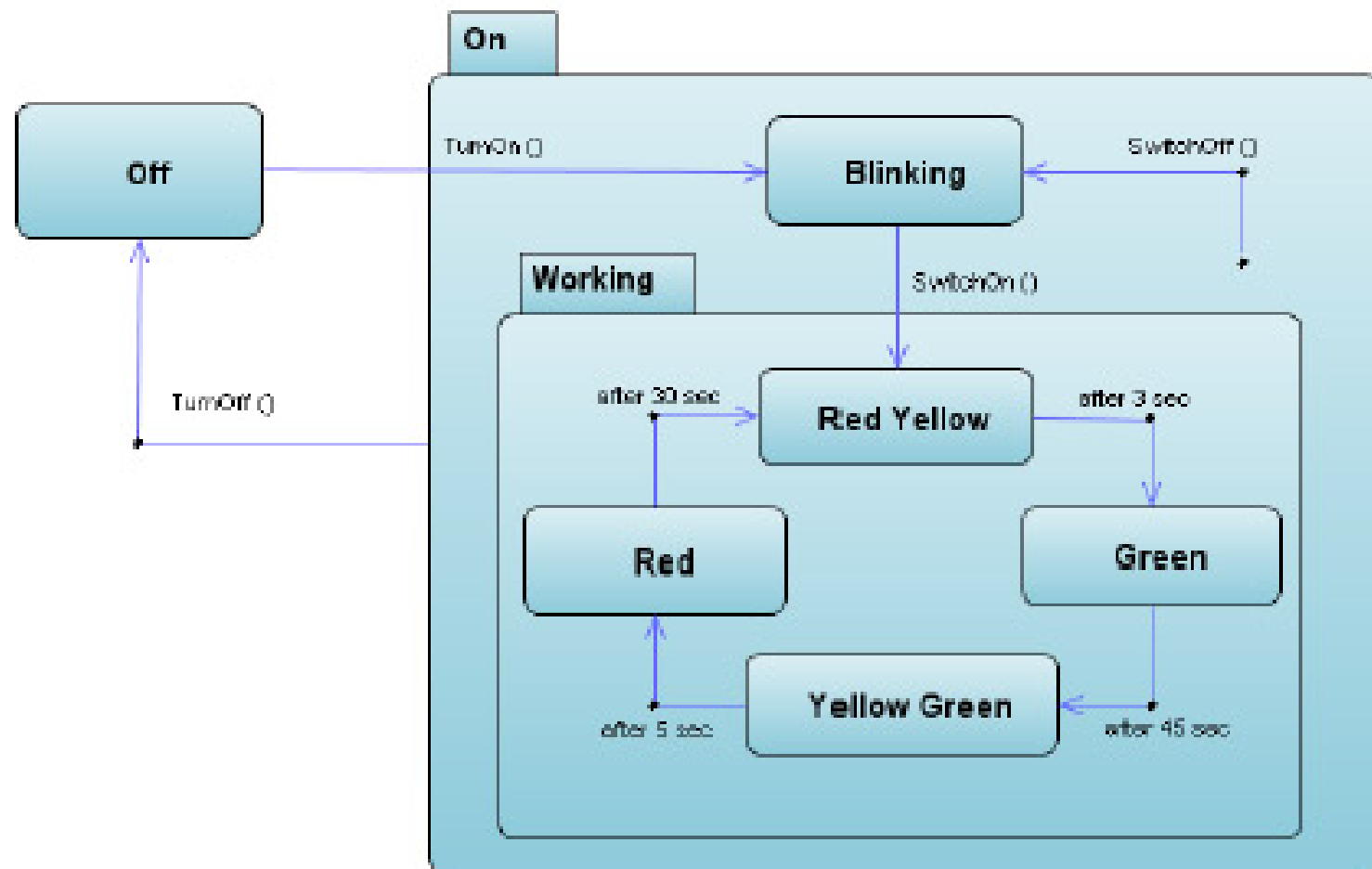
# *Представление поведения*

Представление поведения призвано отвечать на вопрос: как работает система.

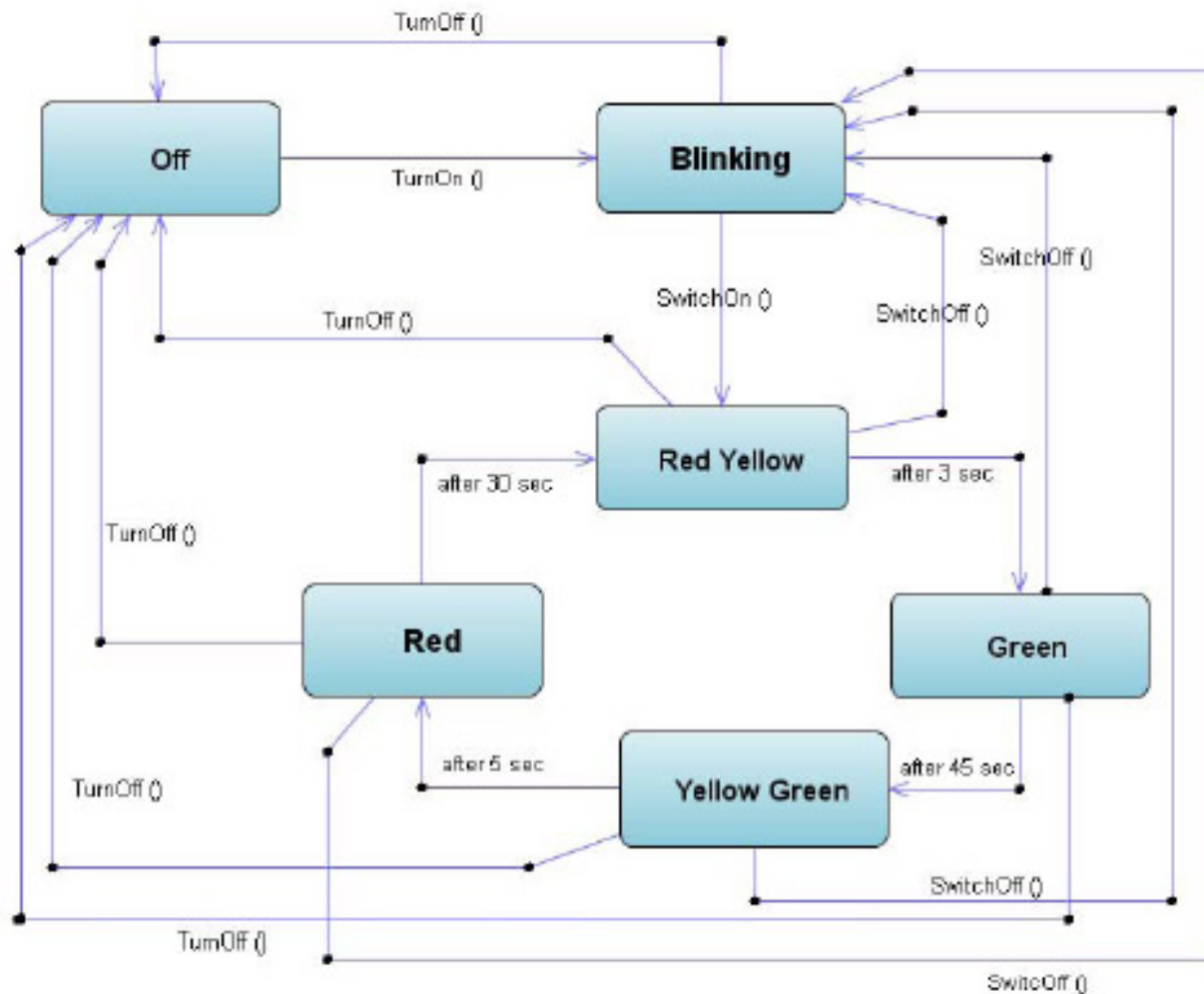
Определяющим признаком для отнесения элементов модели к представлению поведения является явное использования понятия времени, в частности, в форме описания последовательности событий/действий, то есть в форме алгоритма.

Описывается диаграммами состояний и деятельности, а также диаграммами взаимодействия в форме диаграмм кооперации и/или последовательности.

# Составное состояние

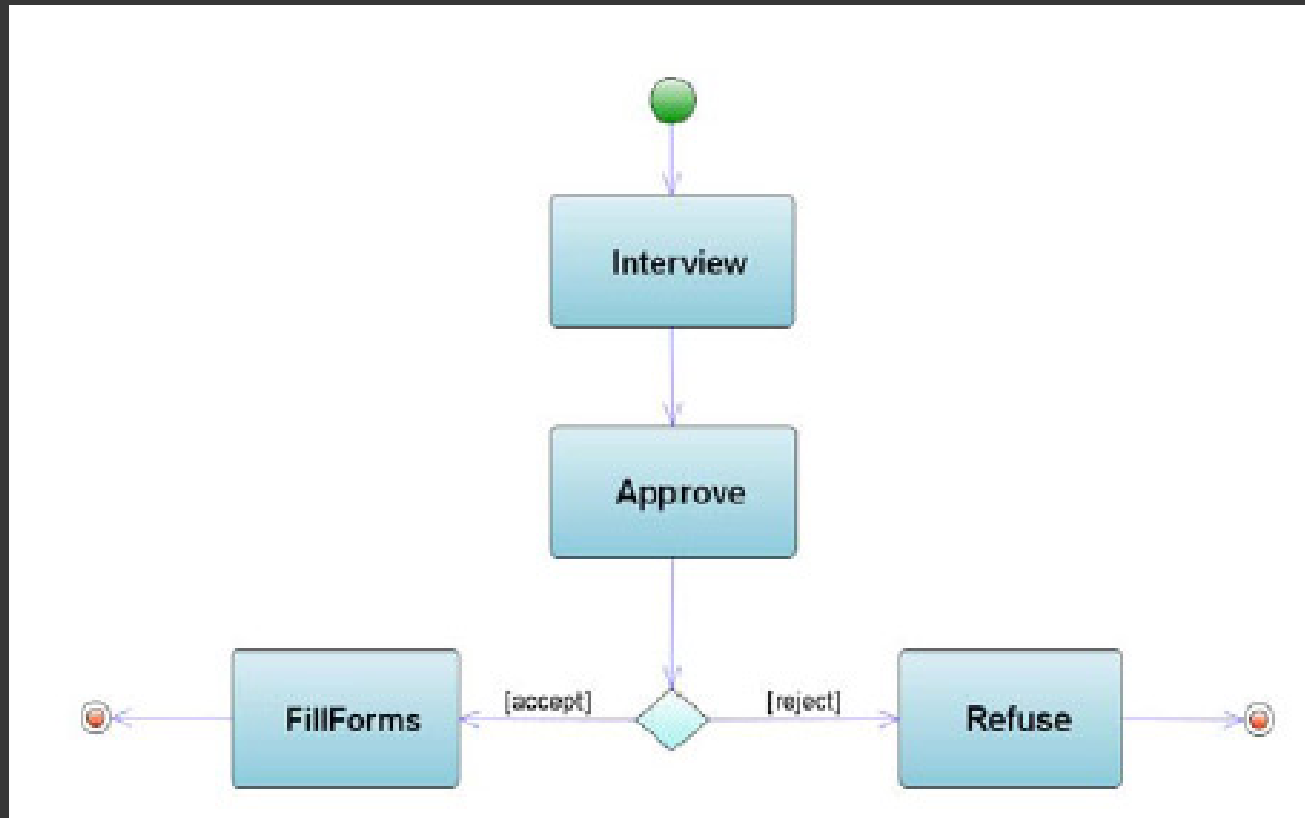


# Составное состояние



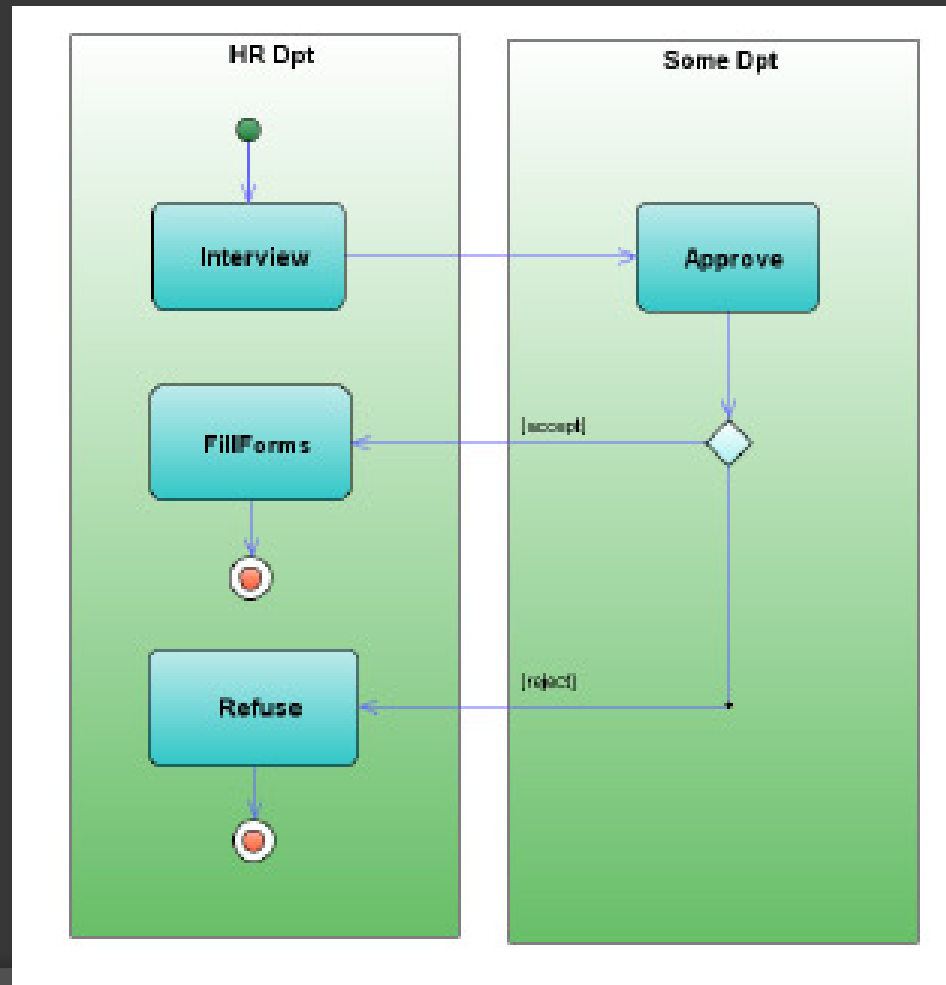
# Диаграмма деятельности

Пример (дорожки)



# Диаграммы деятельности

Дорожка — это графический комментарий, позволяющий классифицировать по некоторому признаку сущности на диаграмме деятельности.



# Диаграммы UML

Диаграммы взаимодействия предназначены для моделирования поведения путем описания взаимодействия объектов для выполнения некоторой задачи или достижения определенной цели.

Взаимодействие происходит путем обмена **сообщениями**. Диаграммы взаимодействия применяются на разных уровнях моделирования: как для описания поведения отдельных операций, так и целых вариантов использования.



# *Диаграммы взаимодействия*

Данный тип диаграмм позволяет описывать не только взаимодействие программных объектов (экземпляров классов), но и взаимодействие экземпляров иных классификаторов: действующих лиц, вариантов использования, подсистем, компонентов, узлов.

Диаграммы взаимодействия графически изображаются в двух формах: диаграммы последовательности и диаграммы кооперации.

# *Диаграммы UML*

Оба типа диаграмм моделируют поведение "по индукции", от частного к общему, путем описания конкретного протокола передачи сообщений (т. е. сценария).

# Диаграммы UML

Сильная сторона состоит в том, что в объектно-ориентированной парадигме обмен сообщениями — это и есть само выполнение программы, поэтому протокол передачи сообщений является наиболее точной моделью поведения. Диаграммы взаимодействия находятся "ближе" к реальному выполнению программы, чем другие средства описания поведения.

Слабость диаграмм взаимодействия состоит в том, что это диаграммы описывают поведение на уровне объектов, а не классов, на уровне протоколов выполнения алгоритма, а не самого алгоритма. Диаграммы взаимодействия менее "алгоритмичны", чем машины состояний и диаграммы деятельности.

# Диаграммы UML

На диаграммах обоих типов основными сущностями являются объекты: экземпляры классификаторов — классов и действующих лиц. Отношениями же являются связи, т. е. экземпляры ассоциаций, по которым передаются сообщения.

# Диаграммы UML

Диаграмма последовательности — это способ описать поведение системы "на примерах".

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

В объектно-ориентированном программировании самым существенным во время выполнения является посылка сообщений взаимодействующими объектами.

# Диаграммы UML

Диаграмма кооперации (в UML 2 – диаграмма коммуникации) семантически эквивалентна диаграмме последовательности.

Фактически, это такое же описание последовательности обмена сообщениями взаимодействующих объектов, только выраженное другими графическими средствами.

# Диаграммы взаимодействия

Диаграммы кооперации и диаграммы последовательности семантически эквиваленты, хотя графически выглядят совсем по-разному.

Семантически эти диаграммы эквиваленты потому, что описывают одно и то же: последовательность передачи сообщений между объектами в процессе взаимодействия объектов. А выглядят по-разному они потому, что в диаграмме последовательности графически подчеркивается упорядоченность во времени передаваемых сообщений, в то время как в диаграмме кооперации на передний графический план выдвигается структура связей между объектами, по которым передаются сообщения.

# Диаграммы взаимодействия

Основные элементы, используемые на этих диаграммах — сообщения.

*Сообщение — это передача управления и информации от одного объекта (отправителя) к другому (получателю).*

Отправка сообщения является действием, а получение сообщения — событием.



# *Диаграммы взаимодействия*

Не все действия связаны с передачей информации и отправкой сообщений. В UML таковыми считаются:

- вызов операции;
- создание объекта;
- уничтожение объекта;
- возврат значения;
- посылка сигнала.

# Диаграммы взаимодействия

Действие записывается в виде текста над (или рядом со) стрелкой, символизирующей сообщение. Если действие имеет параметры (вызов операции, создание объекта, посылка сигнала), то аргументы, соответствующие параметрам по числу и типам, записываются справа от имени действия в круглых скобках. Если действием является вызов операции, возвращающей значения, то слева от имени записывается список переменных для возвращаемых значений (их может быть несколько в случае широковещательного вызова) знак присваивания :=. Таким образом, та часть нотации сообщений, которая относится к выполняемому действию, имеет следующий синтаксис.

переменные := ИМЯ ( аргументы )

# Диаграммы взаимодействия

Сообщение имеет отправителя и получателя.

Получателей может быть несколько, такое сообщение называется *широковещательным*.

Все получатели широковещательного сообщения получают одно и то же сообщение.

Поскольку получение сообщения является событием, то получатель сообщения вместе с информацией получает и управление (для того, чтобы иметь возможность выполнить действия, инициируемые полученным сообщением).

# Типы сообщений



**Вложенный поток управления.** (Отправитель может отправить следующее сообщение только после того, как завершится выполнение всех действий, инициированных данным сообщением)

**Простой поток управления.** (Управление передается от отправителя сообщения получателю. Обычно применяется при моделировании поведения на уровне действующих лиц и вариантов использования.)



**Асинхронный поток управления.** (Сообщение асинхронно передается от отправителя получателю, при этом у отправителя сохраняется свой поток управления, независимый от потока управления получателя)



**Возврат управления.** (Возврат управления после выполнения всех действий, инициированных передачей сообщения с вложенным потоком управления. Можно не отображать на диаграмме, поскольку он подразумевается по умолчанию при вызове операций).



# Диаграммы взаимодействия

Для того, чтобы сообщение могло быть передано от отправителя к получателю, отправитель должен "знать" получателя. Другими словами, должна существовать ассоциация между классами отправителя и получателя, экземпляр которой (связь) и служит тем путем, по которому передается сообщение.

На диаграмме кооперации эта связь всегда изображается в явном виде, на диаграмме последовательности она подразумевается.

# *Диаграммы взаимодействия*

Однако поведение определяется не только и не столько тем, какие объекты посылают какие сообщения, но прежде всего тем, в каком порядке это происходит. UML позволяет определить относительный порядок сообщений во взаимодействии, причем это делается несколькими различными способами.

# Диаграммы взаимодействия

На диаграмме последовательности порядок сообщений определяется временем их отправки, а время считается текущим на диаграмме сверху вниз. Таким образом, сообщения, изображенные выше, предшествуют сообщениям, изображенным ниже.

Порядок можно задать с помощью последовательного *номера сообщения*. Сообщения с меньшими номерами предшествуют сообщениям с большими.

Наконец, порядок можно указать, перечислив (через запятую) номера сообщений, *предшествующих данному*.

# Диаграмма последовательности

Предназначена для моделирования поведения в форме описания протокола сеанса обмена сообщениями между взаимодействующими объектами во время выполнения одного из возможных сценариев.

- На диаграмме присутствуют только те объекты, которые задействованы в данном сеансе. Прочие объекты не показываются, хотя возможно и присутствуют в системе.
- Отображаются только те связи (экземпляры ассоциаций), которые нужны для передачи данной последовательности сообщений, прочие ассоциации не показываются.
- Состав сообщений определяется назначением данного взаимодействия; в других взаимодействиях эти же объекты могут обмениваться другими сообщениями.



# *Диаграмма последовательности*

На диаграмме последовательности выделяют одно направление, соответствующее течению времени. По умолчанию считается, что время течет сверху вниз (не обязательно, например, можно считать, что время течет слева направо, оговорив это специальным примечанием).

Сообщения изображаются прямыми стрелками разного вида. Если передача сообщения считается мгновенной (т. е. время передачи пренебрежимо мало), то стрелка горизонтальна (т. е. перпендикулярна оси времени). Если же нужно отобразить задержанную доставку сообщения, то стрелку немного наклоняют, так чтобы конец стрелки был ниже начала.

# *Диаграмма последовательности*

В направлении оси времени от всех участвующих во взаимодействии объектов отходит прямая пунктирная линия, которая называется линией жизни.

**Линия жизни** представляет объект во взаимодействии: если стрелка отходит от линии жизни объекта, то это означает, что данный объект отправляет сообщение, а если стрелка сообщения входит в линию жизни, то это означает, что данный объект получает сообщение.

# *Диаграмма последовательности*

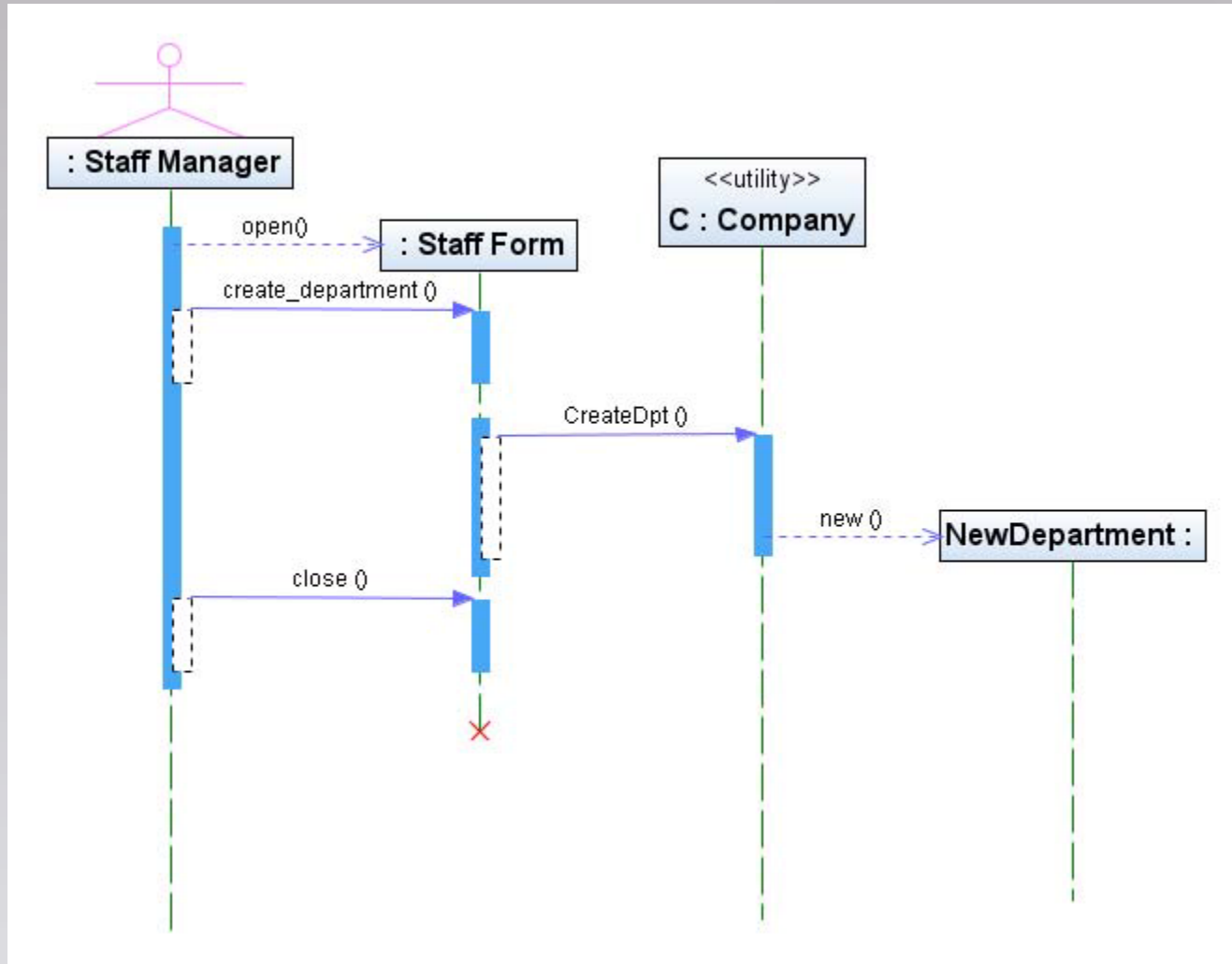
Если же стрелка пересекает линию жизни объекта, то это ничего не значит — сообщение пролетело мимо.

Если в процессе взаимодействия объект заканчивает свое существование, то линия жизни обрывается и в этом месте ставится жирный косой крест

# *Диаграмма последовательности*

Рассмотрим взаимодействие, возникающее при одном из простых сценариев в нашей информационной системе отдела кадров, а именно, создание подразделения. Данное взаимодействие инициируется внешним действующим лицом — менеджером штатного расписания, который открывает соответствующую форму и запускает выполнение операции создания подразделения, после чего закрывает более не нужную ему форму

# Диаграмма последовательности



# Диаграмма последовательности

Подразумеваемая ось времени на диаграмме последовательности не является осью координат, на ней нет никакого масштаба и она задает только отношения "раньше—позже" для сообщений.

Если же нужно в явном виде указать ограничения по времени, например, указать, что время задержки доставки сообщения должно быть ограничено сверху, то на диаграмму в нужном месте (имеет значение только положение по вертикали) рядом с началом или концом стрелки сообщения помещают произвольные идентификаторы, которые называются метками времени, и добавляют ограничение, задающее требуемое условие на значения меток времени.

*Метка времени — это именованная точка на оси времени.*

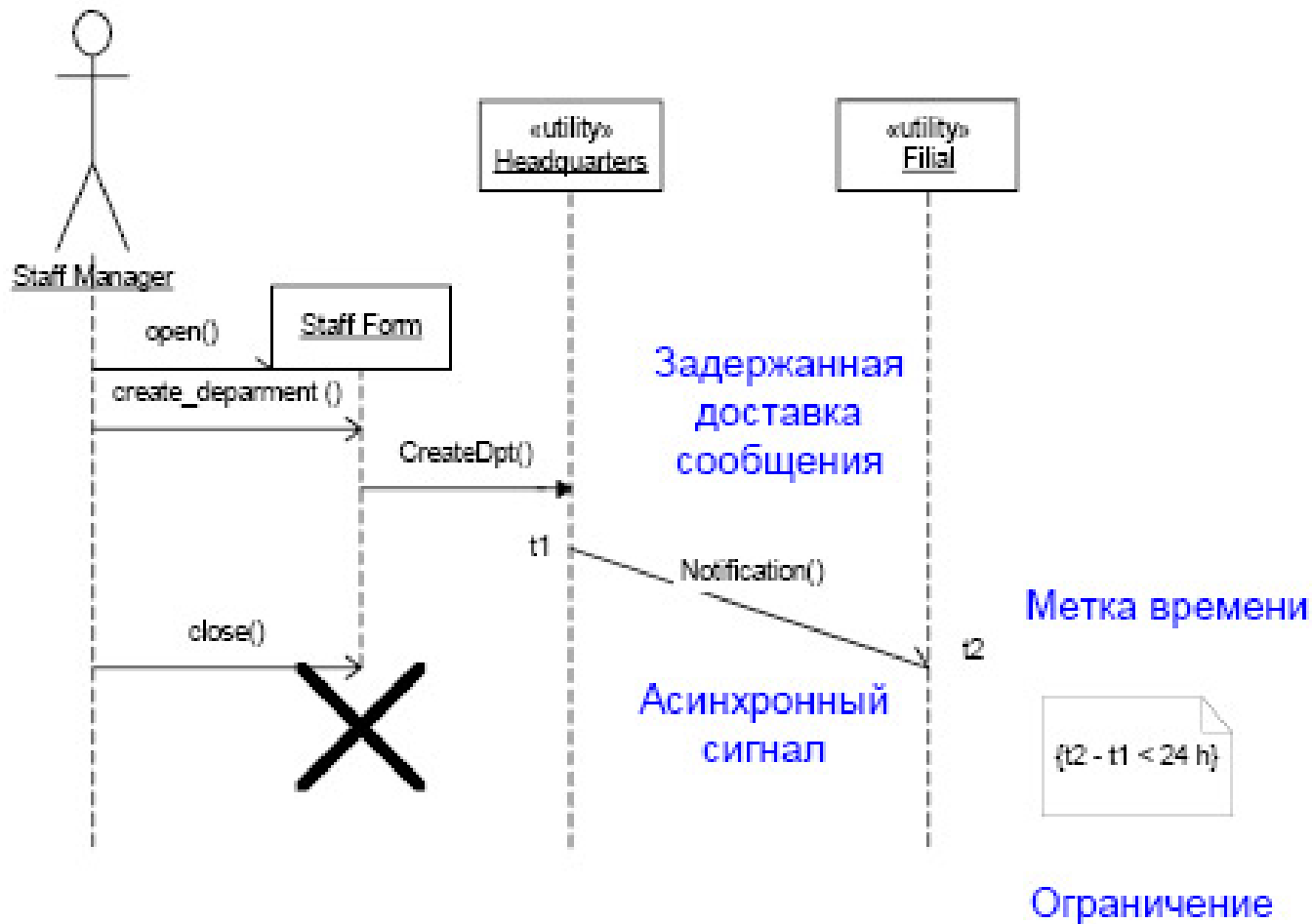
# *Диаграмма последовательности*

Сообщение передает не только данные, но и поток управления.

Чтобы показать, что некоторый объект в определенный период взаимодействия имеет фокус управления, или, как еще говорят, активизирован, на диаграмме последовательности соответствующую часть линии жизни объекта изображают в виде узкой полосы.

Фактически такая полоска означает выполнение операции объекта и называется активацией объекта. Начало активации соответствует приему сообщения вызова операции, а конец активации — завершению выполнения операции и возврату управления.

# Диаграмма последовательности





# *Диаграммы кооперации*

Графических украшений на диаграмме кооперации значительно меньше — данный тип диаграмм графически очень лаконичен и, тем не менее, чрезвычайно выразителен.

# Диаграммы кооперации

Сущностями на диаграммах взаимодействия (в частности, кооперации) являются объекты. Диаграмма кооперации описывает поведение как *взаимодействие*, т. е. как протокол обмена сообщений между объектами. Один и тот же объект может участвовать в различных взаимодействиях, играя в них различные роли. Таким образом, взаимодействие всегда происходит в определенном контексте, который определяется множеством участвующих во взаимодействии объектов и связей.

# Диаграммы кооперации

Обычно контекст выбирается с расчетом на то, чтобы описать взаимодействие, имеющее определенную цель, скажем, выполнить сценарий варианта использования или операцию (это наиболее типичные примеры применения диаграмм кооперации). Варьируя контекстом (добавляя и убирая роли классификаторов и ассоциаций), можно скрывать или показывать детали взаимодействия, описывая поведение на разных уровнях абстракции.

# Диаграммы кооперации

На диаграмме последовательности время жизни объекта относительно данного взаимодействия показывается графически, с помощью смещения вниз символа объекта, создаваемого в процессе взаимодействия и с помощью символа уничтожения объекта (косой крест) на линии жизни уничтожаемого объекта.

На диаграмме кооперации для этой цели используются специальные ключевые слова, которые указываются для ролей классификаторов в форме стандартных стереотипов и/или в форме стандартных ограничений для сообщений

# Диаграммы кооперации

create (стереотип операции в сообщении)

Операция создает объект, т. е. данное сообщение является вызовом конструктора

destroy (стереотип операции в сообщении)

Операция уничтожает объект, т. е. данное сообщение является вызовом деструктора

destroyed (ограничение роли классификатора)

Объект уничтожается в процессе описываемого взаимодействия

New (ограничение роли классификатора)

Объект создается в процессе описываемого взаимодействия

transient (ограничение роли классификатора)

Объект создается и уничтожается в процессе описываемого взаимодействия. Такой объект называется временным. Данное ограничение эквивалентно одновременному указанию ограничений new и destroyed

# Диаграммы кооперации

create (стереотип операции в сообщении)

Операция создает объект, т. е. данное сообщение является вызовом конструктора

destroy (стереотип операции в сообщении)

Операция уничтожает объект, т. е. данное сообщение является вызовом деструктора

destroyed (ограничение роли классификатора)

Объект уничтожается в процессе описываемого взаимодействия

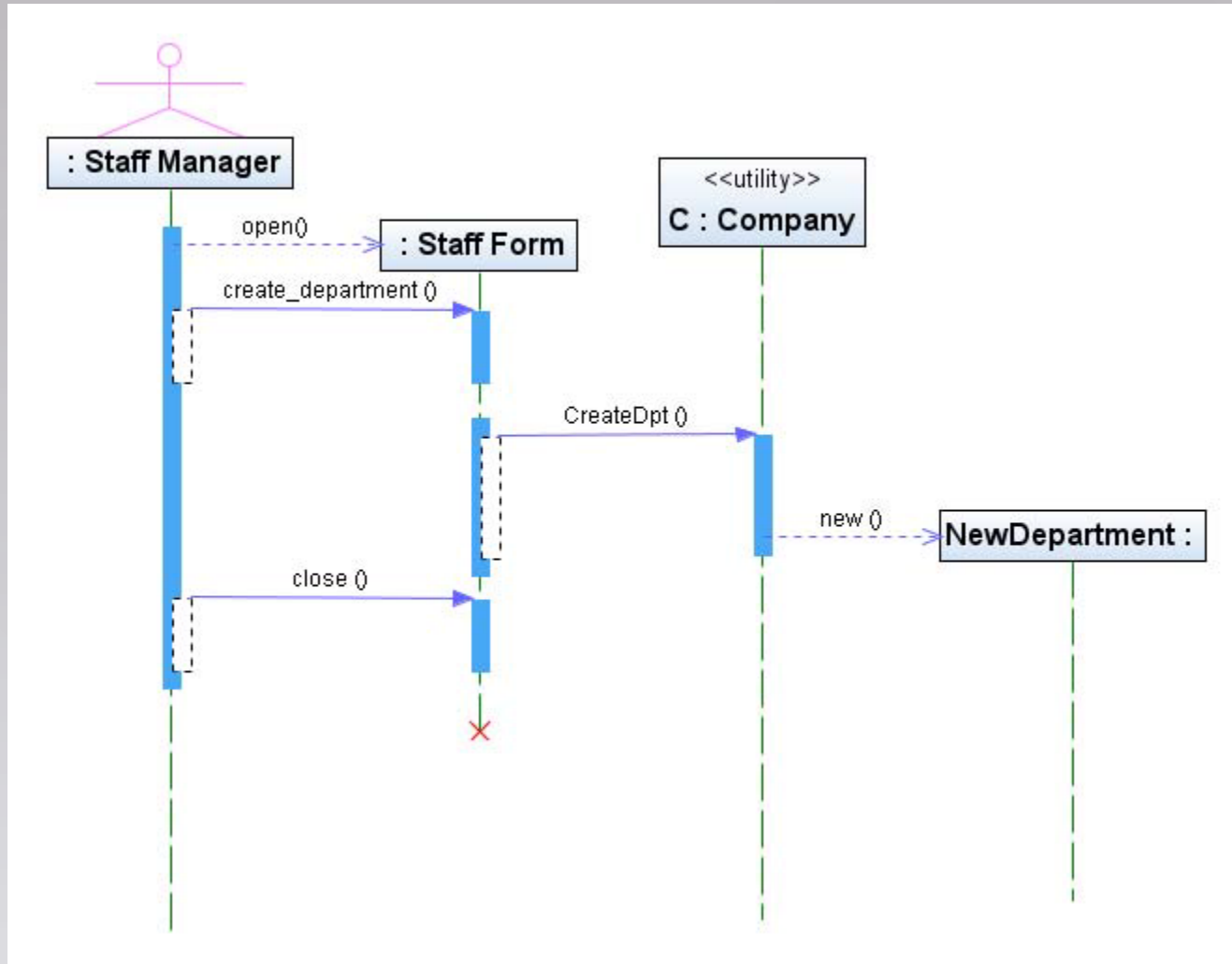
New (ограничение роли классификатора)

Объект создается в процессе описываемого взаимодействия

transient (ограничение роли классификатора)

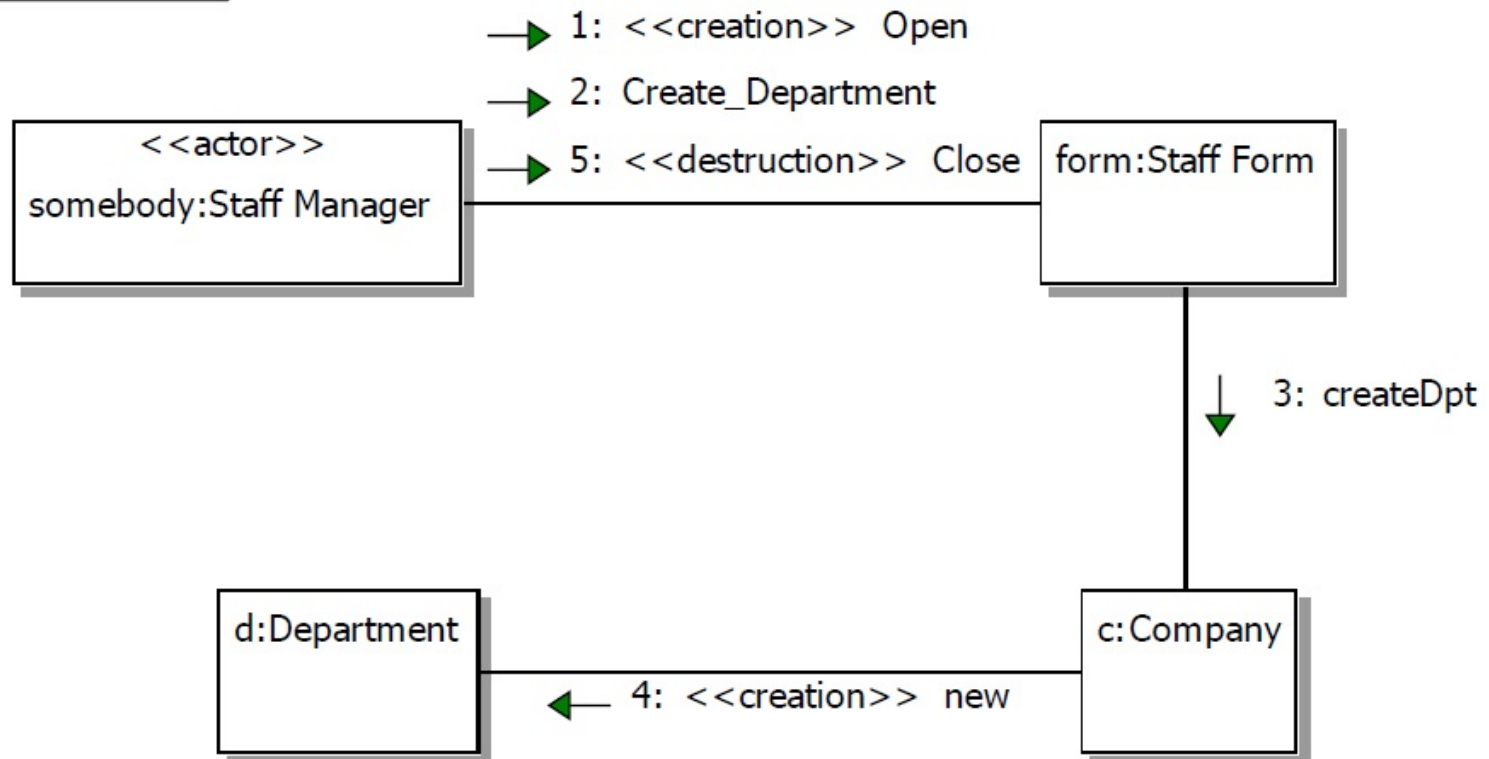
Объект создается и уничтожается в процессе описываемого взаимодействия. Такой объект называется временным. Данное ограничение эквивалентно одновременному указанию ограничений new и destroyed

# Диаграмма последовательности



# Диаграмма кооперации

## sd Create Department





# Выводы

- ❑ Модель поведения — это описание алгоритма работы системы.
- ❑ Для моделирования поведения используются диаграмма состояний, диаграмма деятельности, диаграммы взаимодействия.
- ❑ Диаграммы взаимодействия предназначены для моделирования поведения путем описания взаимодействия объектов для выполнения некоторой задачи или достижения определенной цели.