



Диаграммы взаимодействия

Диаграммы
последовательности,
диаграммы кооперации

Практическая работа № 4. Построение UML – модели системы. Диаграмма последовательности.

Цель работы: изучить структуру модели анализа, правила построения диаграмм последовательности, кооперации.

Задачи: научиться отображать взаимодействие объектов в динамике.




Диаграмма последовательностей отображает взаимодействие объектов в динамике. Диаграмма последовательности относится к диаграммам взаимодействия UML, описывающим поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени.

Диаграмма последовательности отображает временные особенности передачи и приема сообщений объектами.

Кооперация (collaboration) служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы. Цель самой кооперации состоит в том, чтобы специфицировать особенности реализации отдельных наиболее значимых операций в системе. Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации.

Кооперация может быть представлена на двух уровнях:

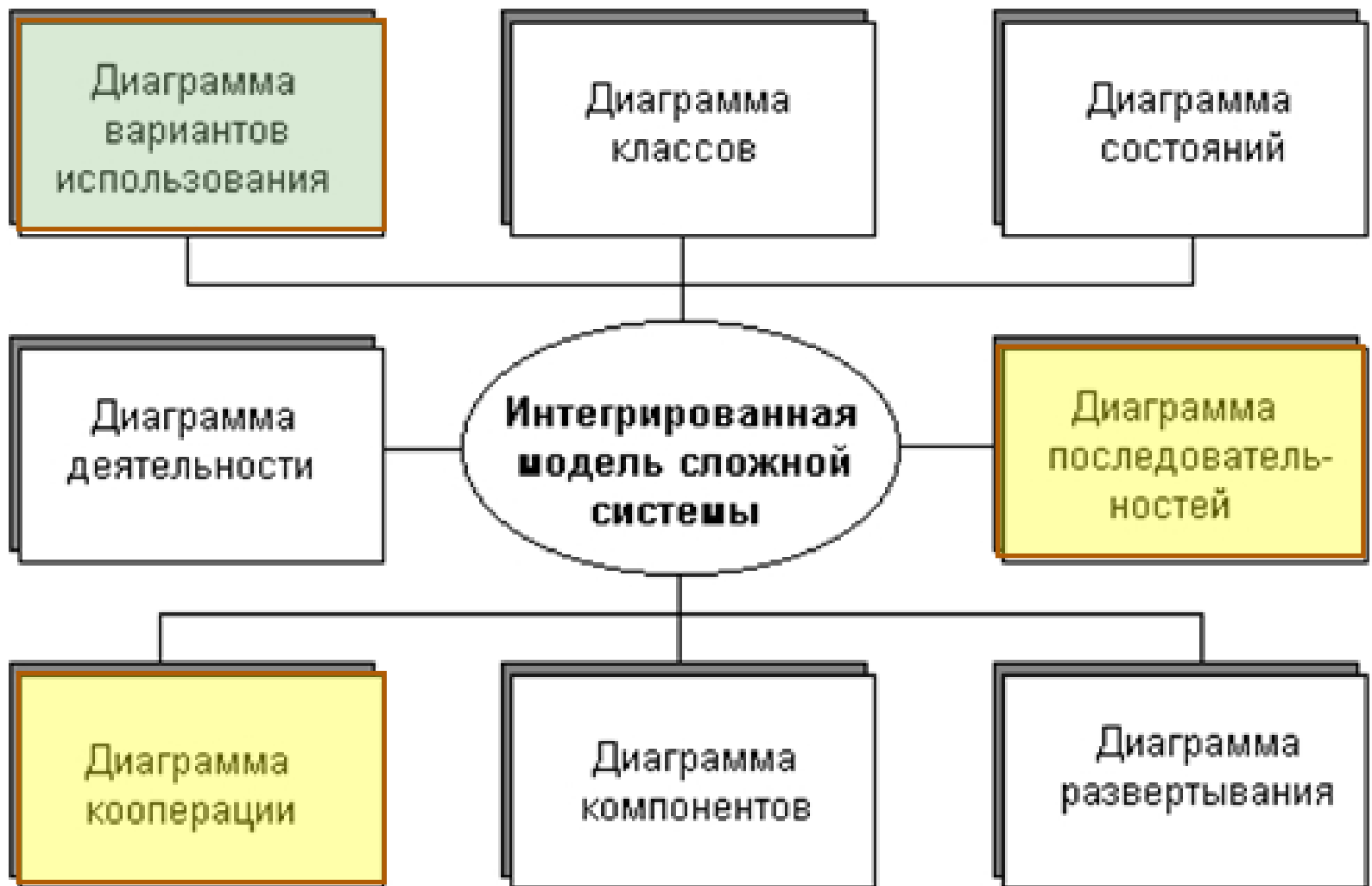
- На уровне спецификации – показывает роли классификаторов и роли ассоциаций в рассматриваемом взаимодействии.
- На уровне примеров – указывает экземпляры и связи, образующие отдельные роли в кооперации.

Построение диаграммы последовательности начинается с размещения на ней объектов, которые будут обмениваться сообщениями. Сначала необходимо разместить объекты, которые посылают сообщения, а потом объекты, получающие их.

При разработке диаграмм следует придерживаться следующих правил:

1. Для выбранного варианта использования необходимо перенести с диаграммы классов анализа все участвующие в нем классы, а с диаграммы вариантов использования – актеров.
2. На диаграмме коммуникации между классами следует отобразить ассоциации, перенесенные с диаграммы классов анализа, а также добавить ассоциации, связывающие актеров с граничными классами.
3. На стадии анализа имена сообщениям можно давать произвольно или в виде стереотипов. В дальнейшем (в модели проектирования) имена сообщений должны соответствовать методам классов.
4. Имена сущностей на диаграммах (экземпляры актеров и объекты) должны быть подчеркнуты и обозначены соответствующим образом.
5. На диаграммах последовательности символ уничтожения объектов следует задавать только для тех объектов, которые во время взаимодействия действительно уничтожаются.

Экземпляры актеров и объекты классов сущностей (долгоживущая информация), как правило, существуют до начала и после окончания взаимодействия. Для них символ уничтожения не показывается. Объекты граничных и управляющих классов, напротив, в большинстве случаев создаются на момент взаимодействия и по его окончании уничтожаются. В связи с этим для них требуется отображать символ уничтожения.



Виды диаграмм взаимодействия

Существуют 2 вида диаграмм взаимодействий:

- 1) диаграммы последовательности действий – *sequence diagram*;
- 2) диаграммы кооперации (кооперативные диаграммы) – *collaboration diagram*

Диаграмма последовательности

Диаграмма взаимодействия предназначена для моделирования отношений между объектами (ролями, классами, компонентами) системы в рамках одного прецедента.

Данный вид диаграмм отражает следующие аспекты проектируемой системы:

- обмен сообщениями между объектами (в том числе в рамках обмена сообщениями со сторонними системами)
- ограничения, накладываемые на взаимодействие объектов
- события, инициирующие взаимодействия объектов.

Диаграмма последовательности является одной из разновидностей диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

Диаграмма последовательности

Диаграмма последовательности — показывает взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления.

Основными элементами диаграммы последовательности являются обозначения **объектов**, вертикальные «**линии жизни**», отображающие течение времени, **элементы**, отражающие деятельность объекта или исполнение им определенной функции, и **стрелки**, показывающие обмен сигналами или сообщениями между объектами.

Объекты располагаются с лева на права таким образом, чтобы крайним с лева был тот **объект**, который инициирует взаимодействие.

Объекты

- Объект – экземпляр класса.

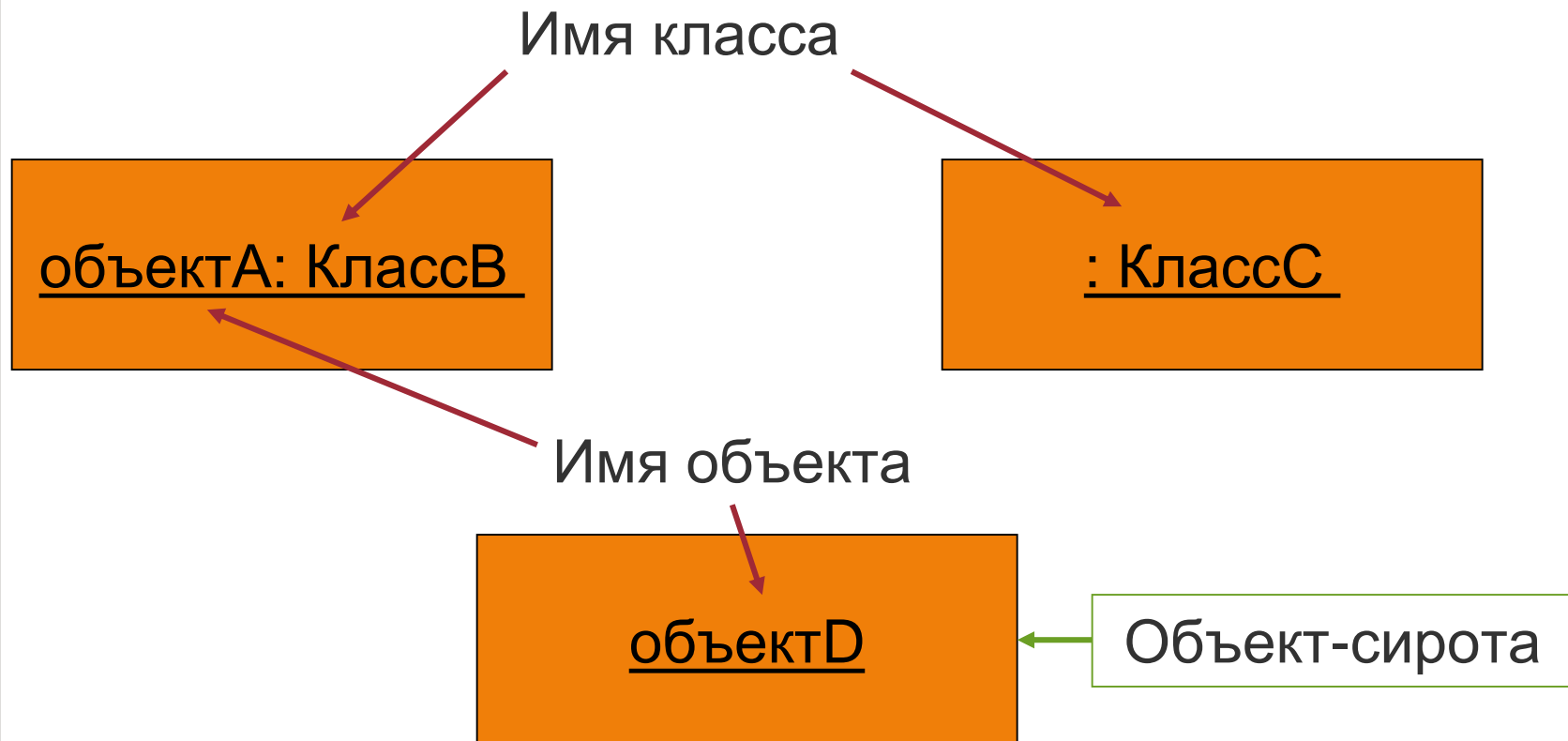
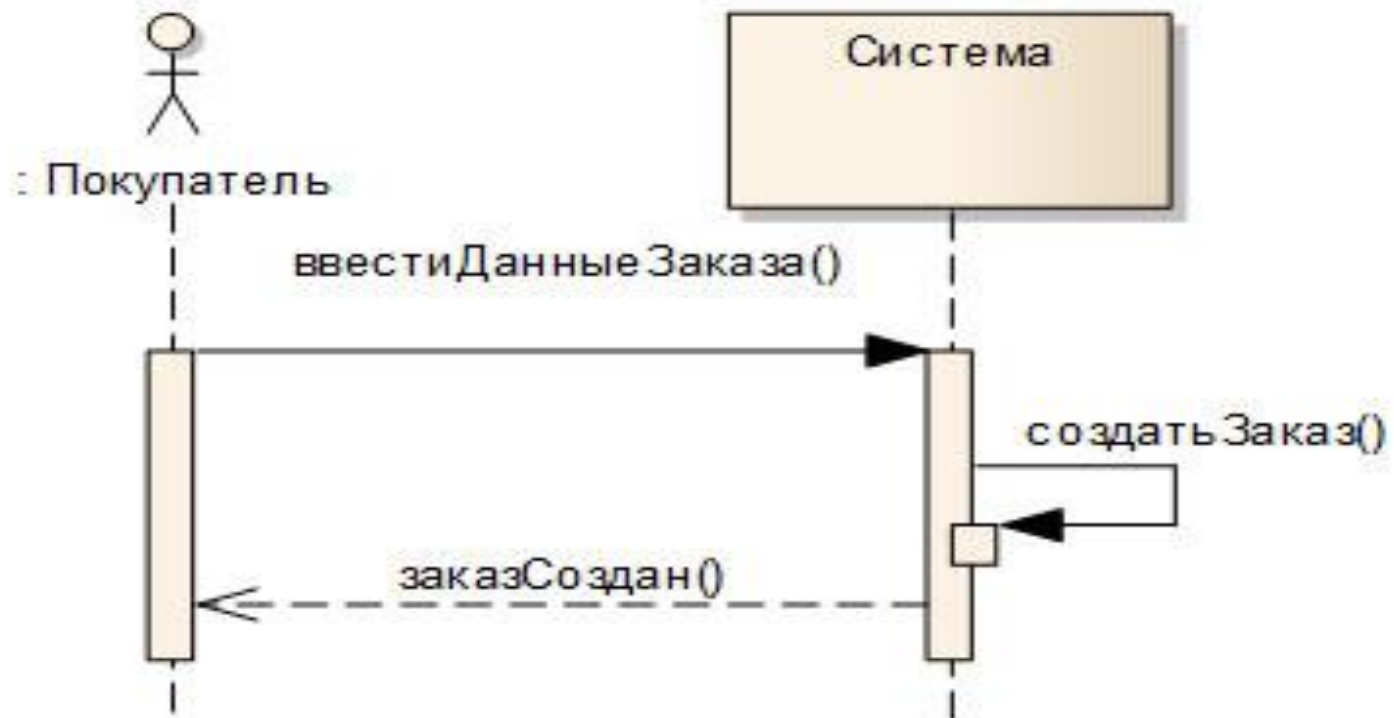


Диаграмма последовательности

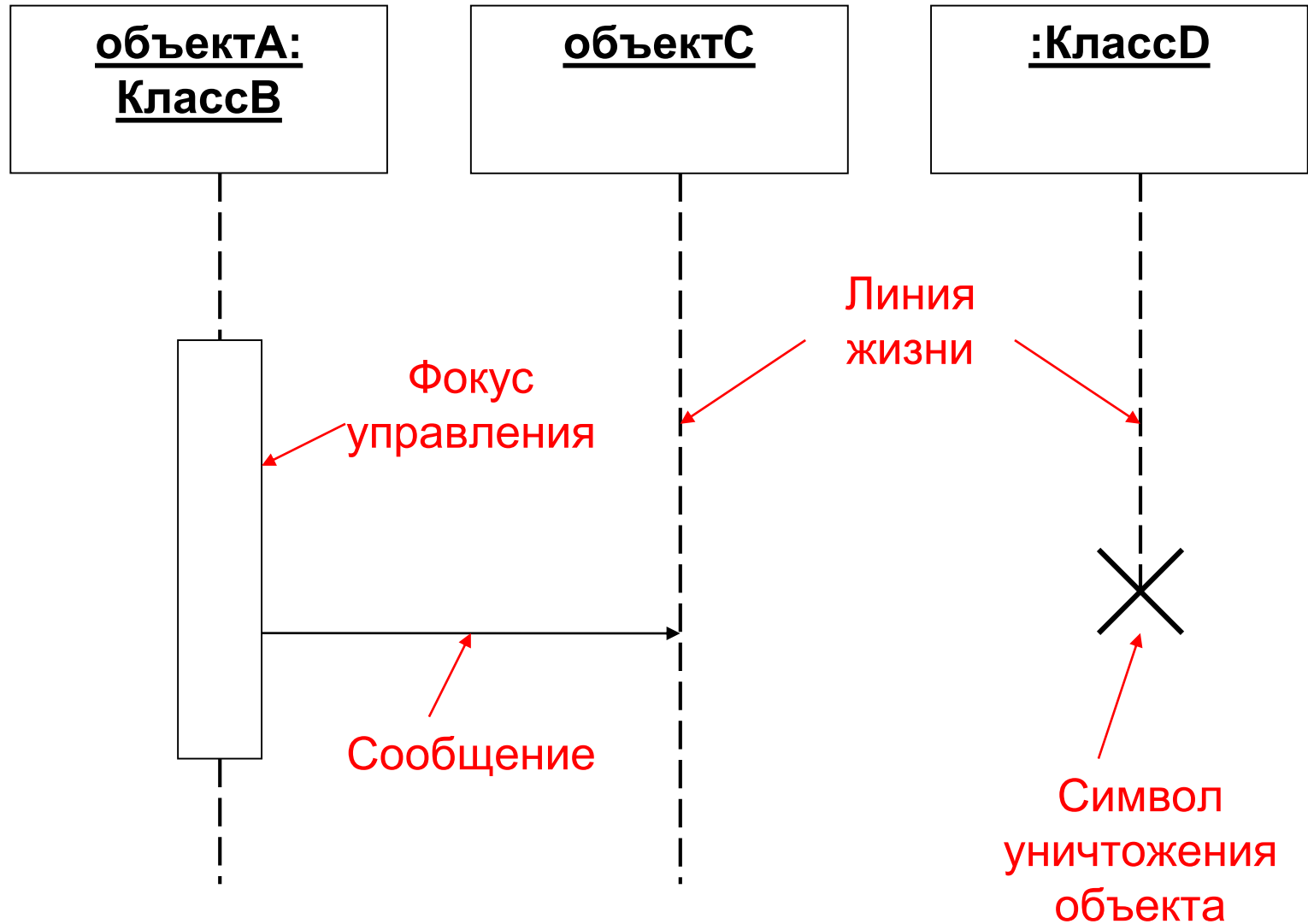
Линия жизни показывает время, в течение которого объект существует в Системе.

Периоды активности объекта в момент взаимодействия показываются с помощью фокуса управления.

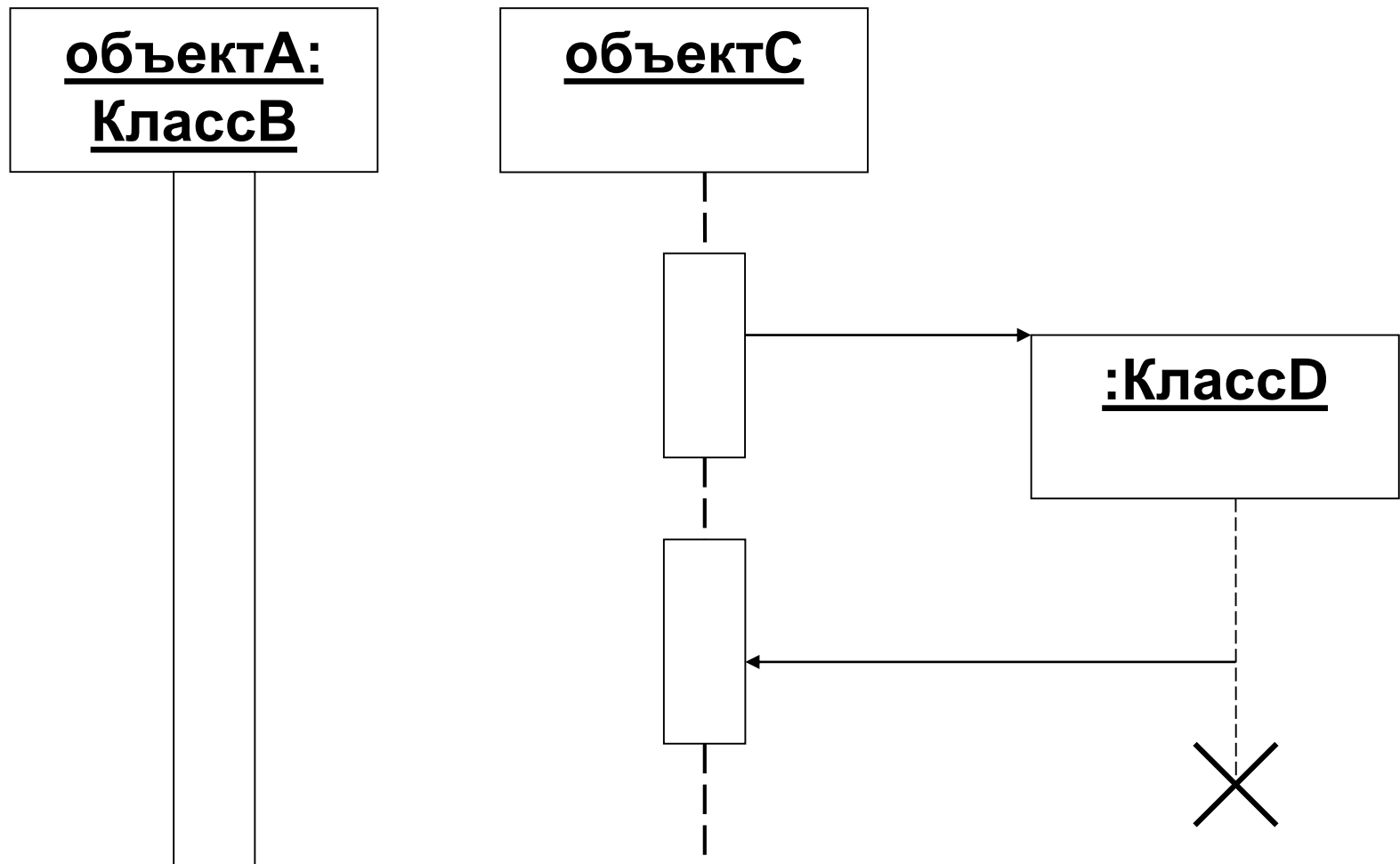
Временная шкала на диаграмме направлена сверху вниз.



Графические элементы диаграммы последовательности



Линия жизни и фокус управления

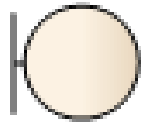


Нотации UML



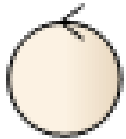
Покупатель

Actor – экземпляр участника процесса



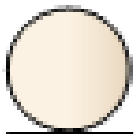
Форма заказа

Boundary (граничный класс) – Класс-Разграничитель - используется для классов, отделяющих внутреннюю структуру системы от внешней среды (экранная форма, пользовательский интерфейс, устройство ввода-вывода)



Менеджер
заказа

Control (управляющий класс) – Класс-контроллер - активный элемент, который используется для выполнения некоторых операций над объектами (программный компонент, модуль, обработчик)



Заказ

Entity (класс-сущность) - обычно применяется для обозначения классов, которые хранят некую информацию о бизнес-объектах (соответствует таблице или элементу БД)

Сообщения на диаграмме последовательности

На диаграмме деятельности выделяются **сообщения**, инициирующие ту или иную **деятельность** или являющиеся ее **следствием**. На диаграмме состояний частично показан обмен сообщениями в рамках сообщений инициирующих изменение состояния объекта.

На диаграмме последовательности мы можем увидеть следующие аспекты:

- Сообщения, побуждающие объект к действию;
- Действия, которые вызываются сообщениями (методы) – передача сообщения следующему объекту или возвращение определенных данных объекта;
- Последовательность обмена сообщениями между объектами.

Прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

Виды сообщений

Синхронное сообщение (synchCall) - соответствует синхронному вызову операции и подразумевает ожидание ответа от объекта получателя. Пока ответ не поступит, никаких действий в Системе не производится.

Асинхронное сообщение (asynchCall) - которое соответствует асинхронному вызову операции и подразумевает, что объект может продолжать работу, не ожидая ответа.

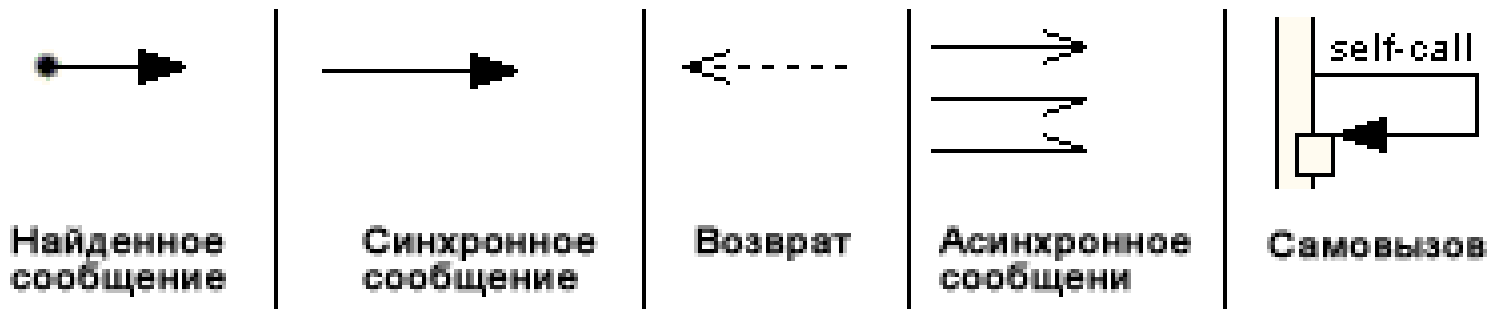
Ответное сообщение (reply) – ответное сообщение от вызванного метода. Данный вид сообщения показывается на диаграмме по мере необходимости или, когда возвращаемые им данные несут смысловую нагрузку.

Потерянное сообщение (lost) – сообщение, не имеющее адресата сообщения, т.е. для него существует событие передачи и отсутствует событие приема.

Найденное сообщение (found) – сообщение, не имеющее инициатора сообщения, т.е. для него существует событие приема и отсутствует событие передачи

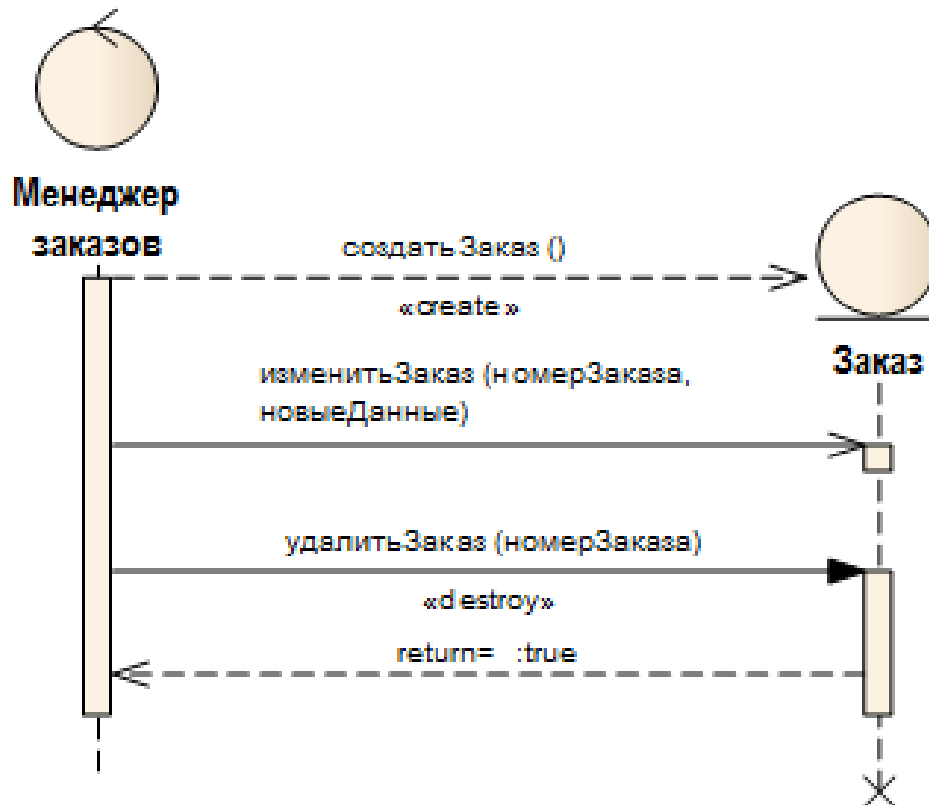
Сообщение

- Представляет собой законченный фрагмент информации, который отправляется одним объектом другому;
- Прием сообщения инициирует выполнение определенных действий;
- **разновидности сообщений:**



Для сообщений также доступен ряд predefined стереотипов.

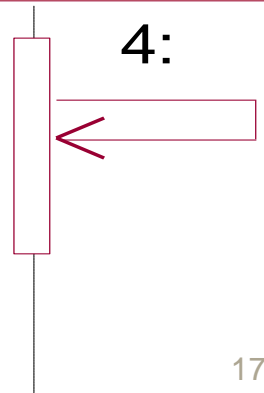
- Сообщение со стереотипом **create**, вызывает в классе метод, который создает экземпляр класса.
- Для уничтожения экземпляра класса используется сообщение со стереотипом **destroy**, при этом в конце линии жизни объекта отображаются две перекрещенные линии.



Сообщение, отправленное самому себе – рефлексивное (саморегулирование).

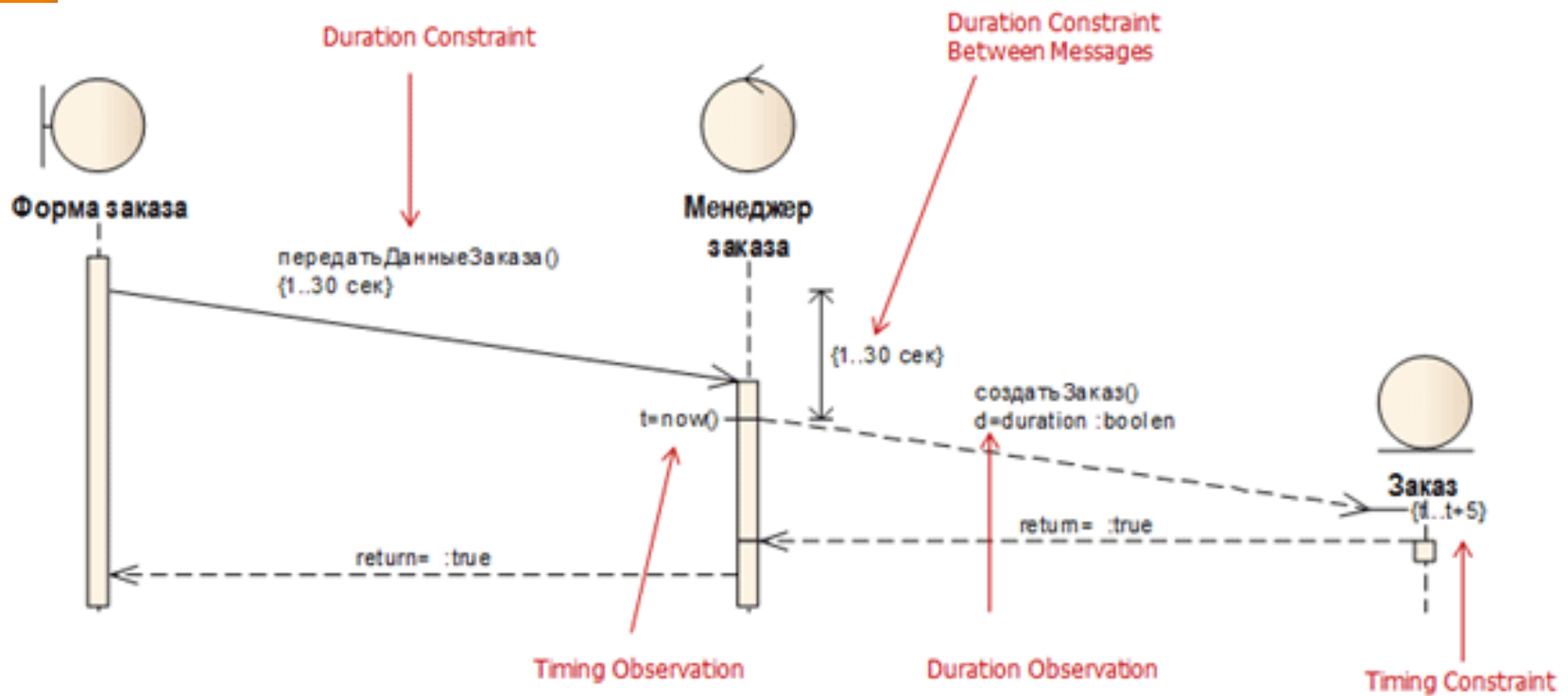
ИмяОбъекта4 :
ИмяКласса4

4:

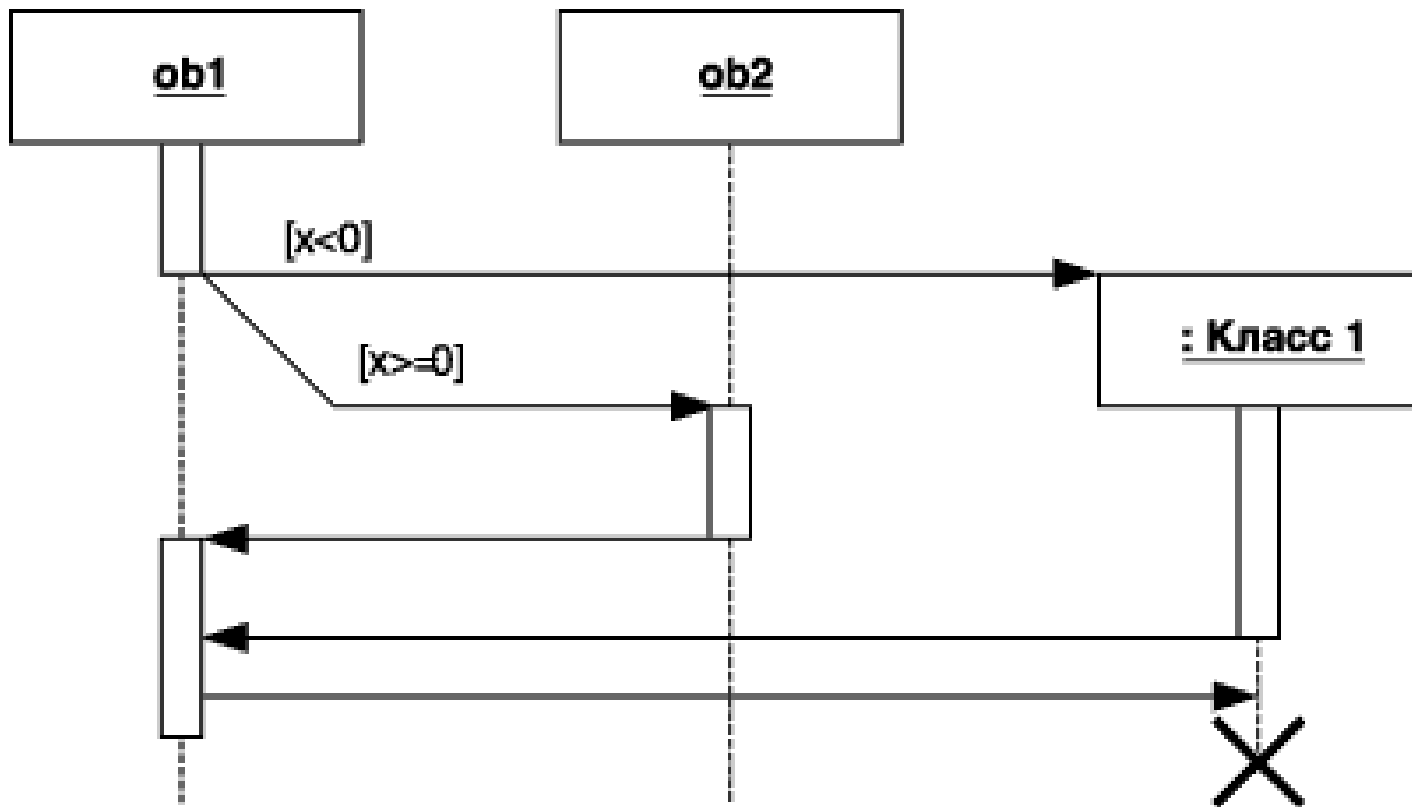


При отображении работы с сообщениями можно указать следующие временные параметры:

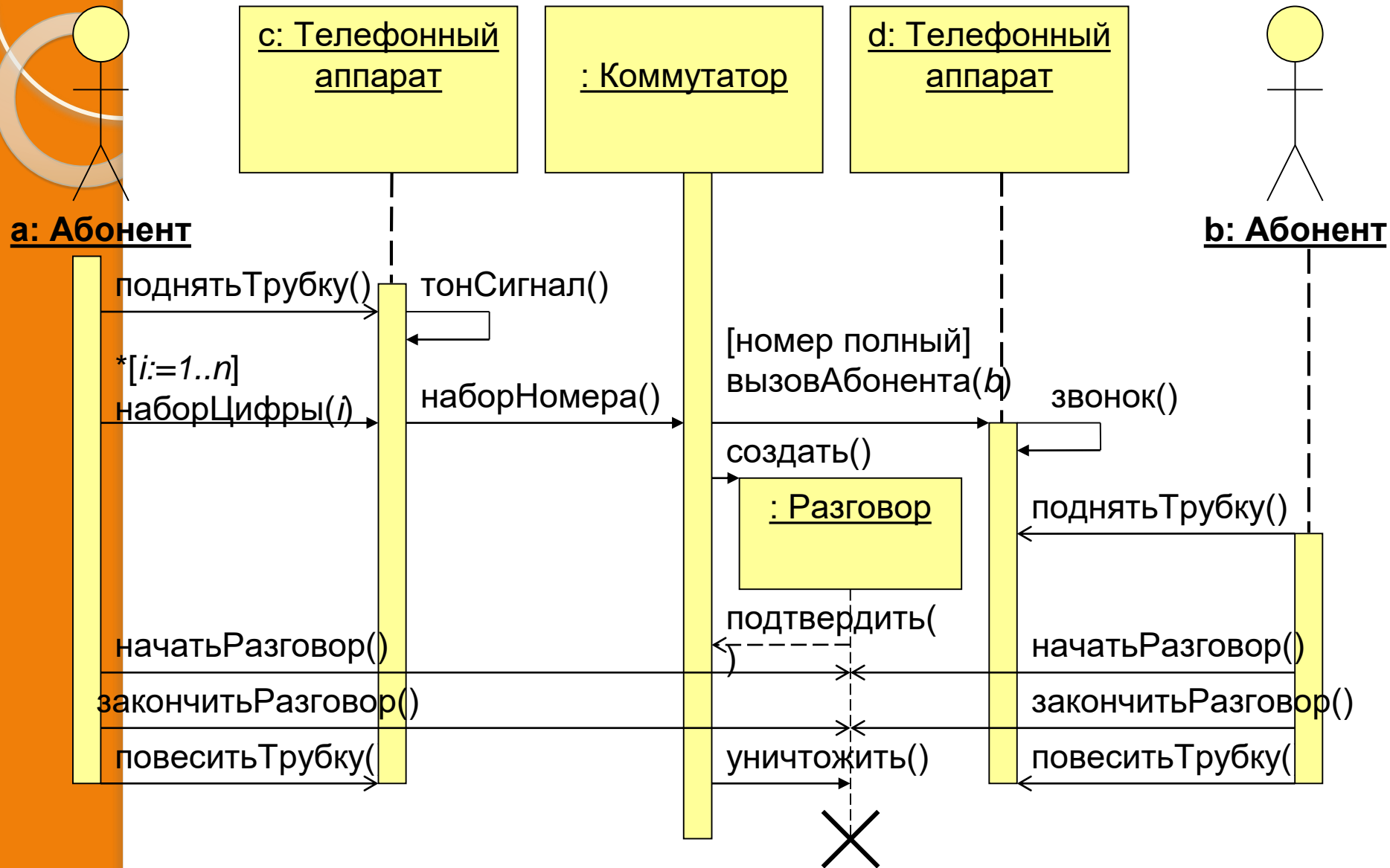
- **ограничение продолжительности** (Duration Constraint) – минимальное и максимальное значение продолжительности передачи сообщения
- **ограничение продолжительности ожидания** между передачей и получением сообщения (Duration Constraint Between Messages)
- **перехват продолжительности сообщения** (Duration Observation)
- **временное ограничение** (Timing Constraint) – временной интервал, в течение которого сообщение должно прийти к цели (устанавливается на стороне получателя)
- **перехват времени**, когда сообщение было отправлено (Timing Observation)



Ветвление потока



Пример диаграммы последовательности



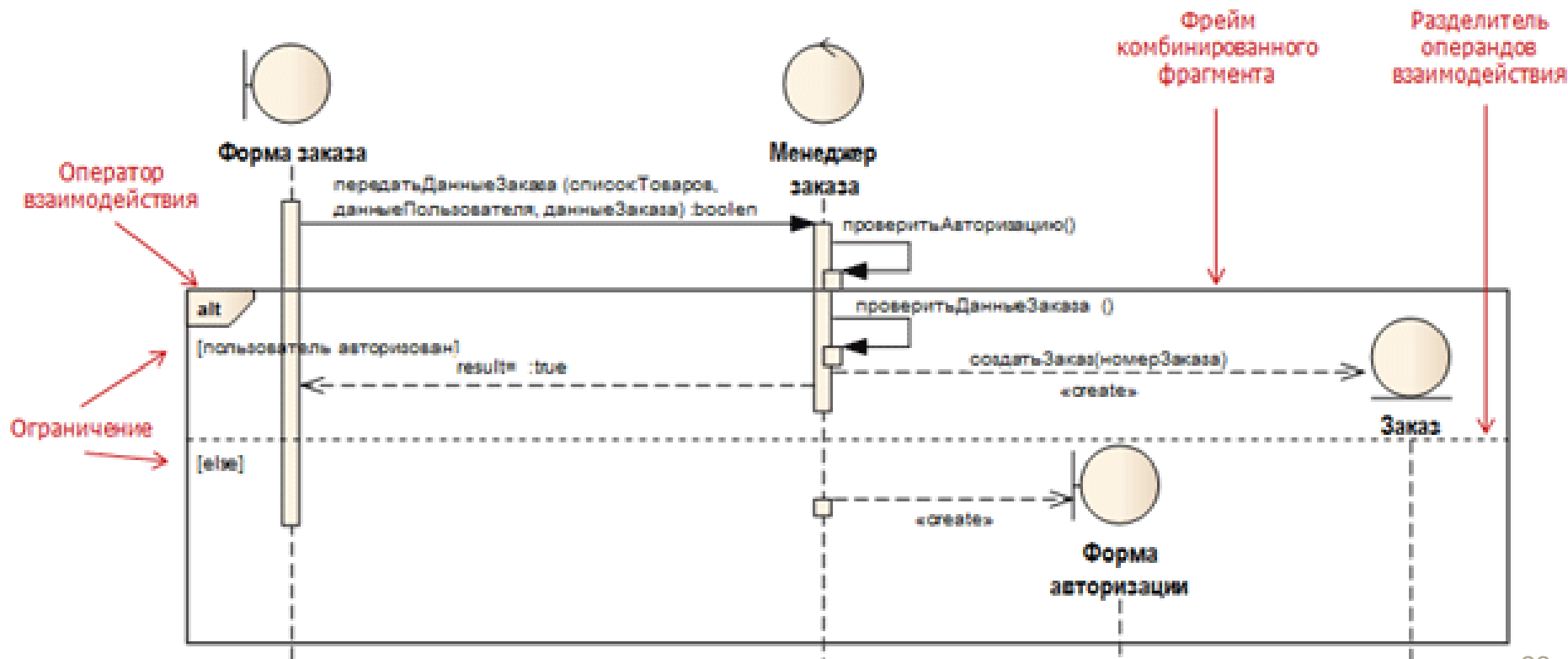
Отдельные фрагменты диаграммы взаимодействия можно выделить с помощью фрейма.

Фрейм должен содержать **метку оператора взаимодействия**. UML содержит следующие операнды:

- **Alt** - Несколько альтернативных фрагментов (alternative); выполняется только тот фрагмент, условие которого истинно
- **Opt** - Необязательный (optional) фрагмент; выполняется, только если условие истинно. Эквивалентно **alt** с одной веткой
- **Par** - Параллельный (parallel); все фрагменты выполняются параллельно
- **loop** - Цикл (loop); фрагмент может выполняться несколько раз
- **region** - Критическая область (critical region); фрагмент может иметь только один поток, выполняющийся за один прием
- **Neg** - Отрицательный (negative) фрагмент; обозначает неверное взаимодействие
- **ref** - Ссылка (reference); ссылается на взаимодействие, определенное на другой диаграмме.
- **Sd** - Диаграмма последовательности (sequence diagram); используется для очерчивания всей диаграммы последовательности, если это необходимо.

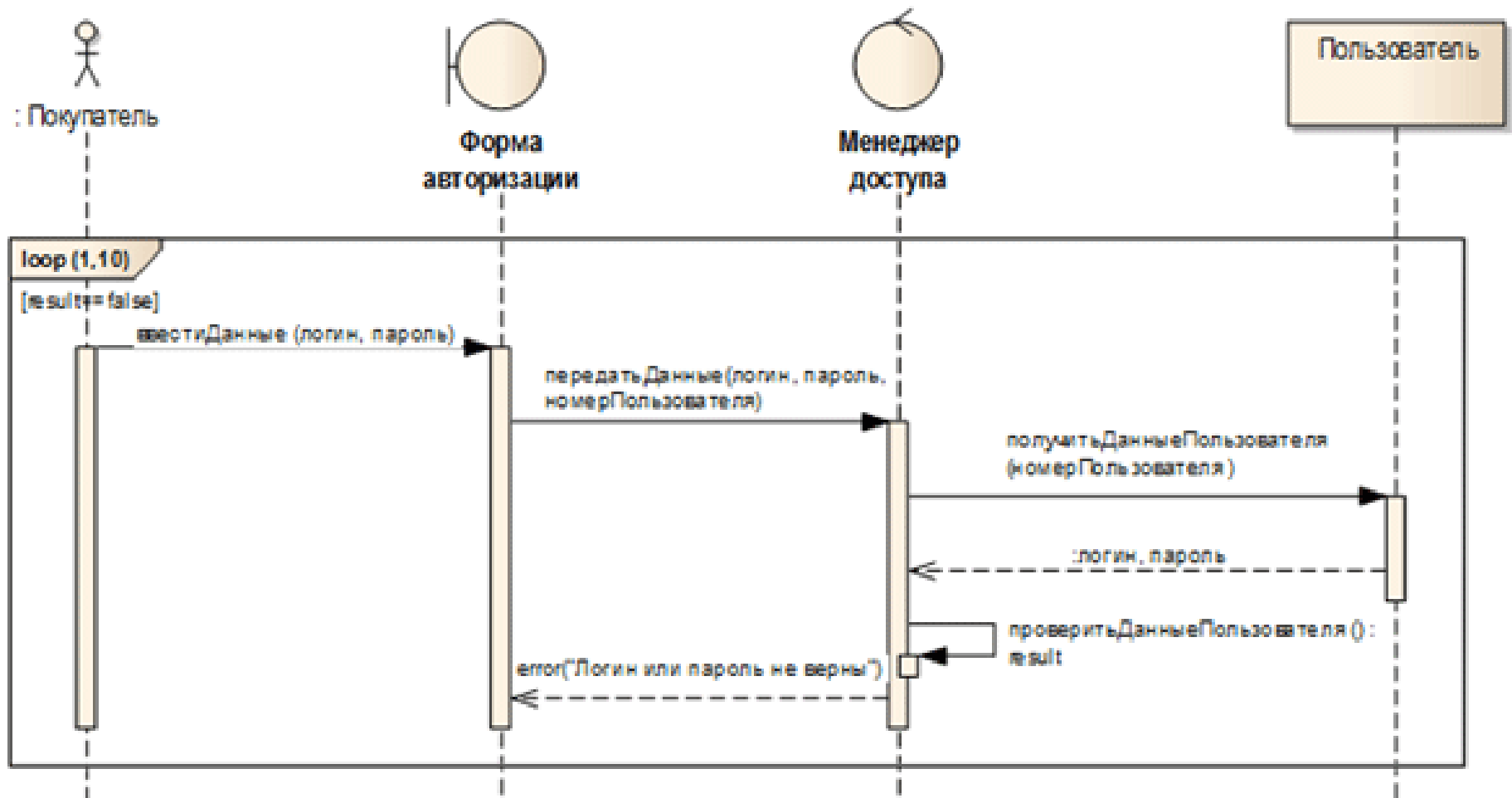
При использовании условного или параллельного операнда фрейм делится на регионы взаимодействия с помощью разделителя операторов взаимодействия.

Операнд **alt** используется при моделировании расширения сценария, т.е. при наличии альтернативного потока взаимодействия. Оператор **opt** используется, если сообщение должно быть передано, только при истинности какого-то условия. Данный фрейм используется без разделения на регионы.

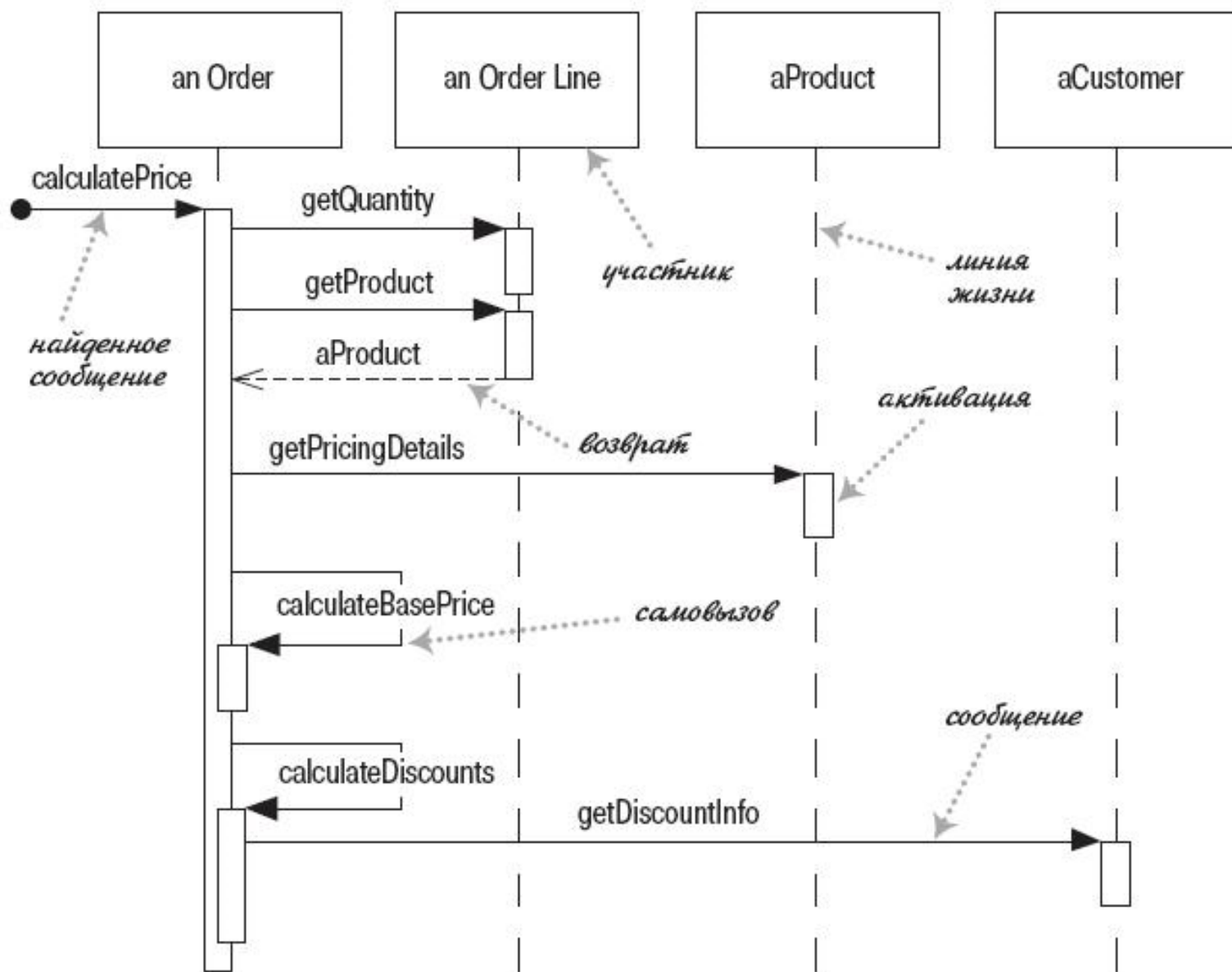


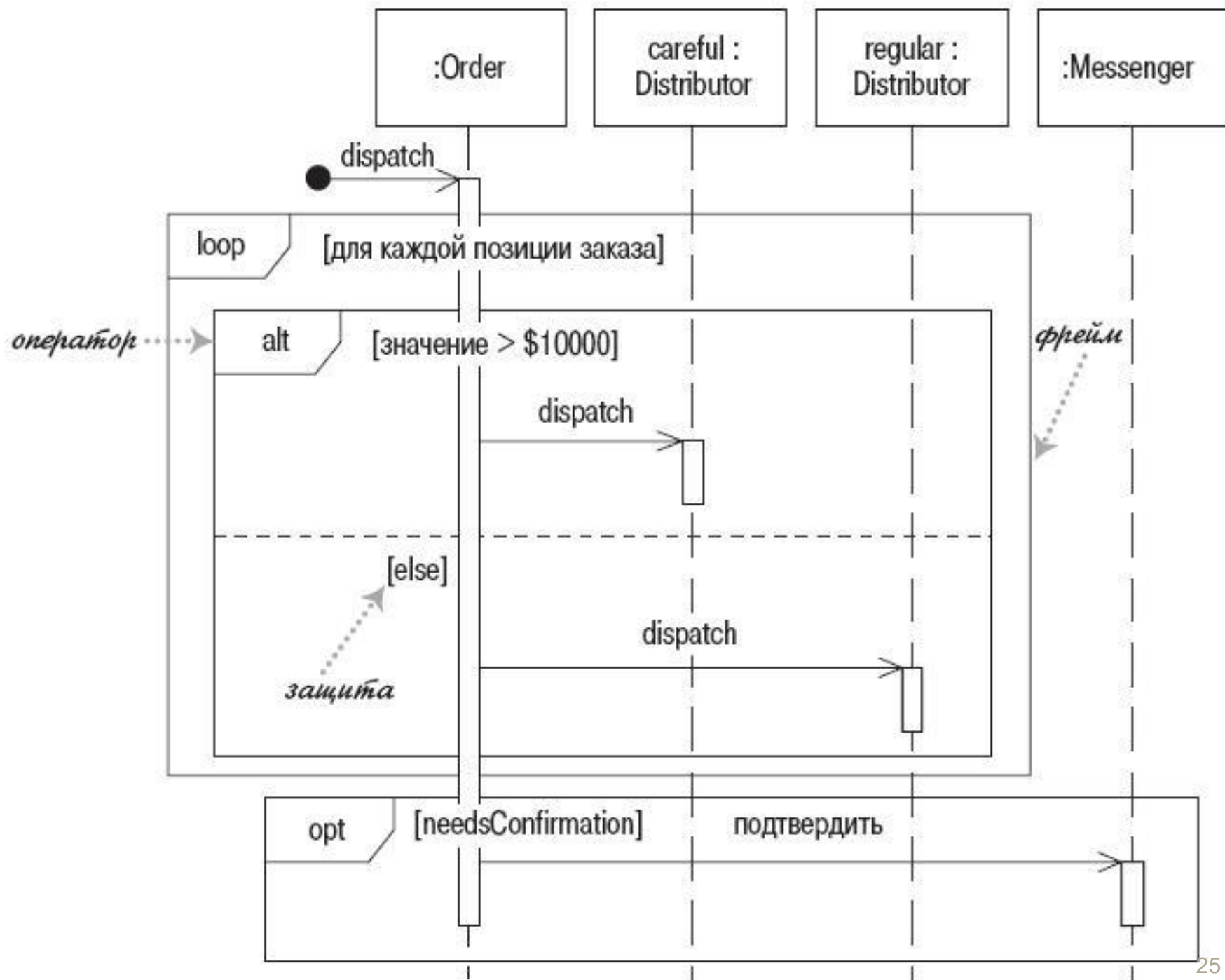
Цикличность потока взаимодействия может быть представлена на диаграмме последовательности с помощью операнда **loop**. При использовании оператора цикла можно указать **минимальное** и **максимальное** число итераций.




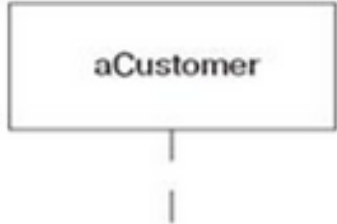
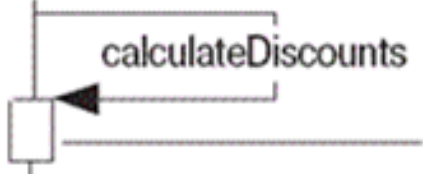
Также фрейм должен содержать условие, при наступлении которого взаимодействие повторяется.


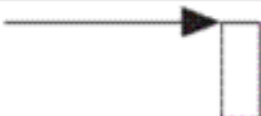
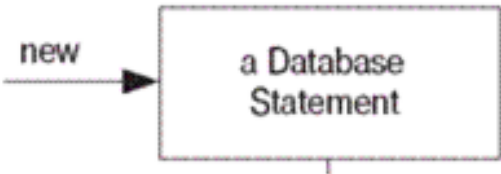





- У первого сообщения нет участника, пославшего его, поскольку оно приходит из неизвестного источника. Оно называется **найденным сообщением**





| Термин | Изображение | Описание |
|---------------------|--|---|
| Найденное сообщение |  | У первого сообщения нет участника, пославшего его, поскольку оно приходит из неизвестного источника. Оно называется найденным сообщением (found message). |
| Сообщение |  | Команда, отправляемая другому участнику. Может содержать только передаваемые данные. |
| Линия жизни |  | Каждая линия жизни имеет полосу активности, которая показывает интервал активности участника при взаимодействии. Она соответствует времени нахождения в стеке одного из методов участника. В языке UML полосы активности не обязательны, но я считаю их исключительно удобными при пояснении поведения. |
| Участник |  | В большинстве случаев можно считать участников диаграммы взаимодействия объектами, как это и было в действительности в UML 1. Но в UML 2 их роль значительно сложнее. Поэтому здесь употребляется термин участники (participants), который формально не входит в спецификацию UML. |
| Самовывоз |  | Участник отправляет сообщение (команду) самому себе. |

| Термин | Изображение | Описание |
|-----------------------------|---|--|
| Возврат |  | Передача управления обратно участнику, который до этого инициировал сообщение. |
| Активация |  | На изображении это - белый вертикальный прямоугольник. Момент, когда участник начинает действовать в ответ на принятое сообщение. |
| Создание |  | В случае создания участника надо нарисовать стрелку сообщения, направленную к прямоугольнику участника. Если применяется конструктор, то имя сообщения не обязательно, но можно маркировать его словом «new» в любом случае. Если участник выполняет что-нибудь непосредственно после создания, например команду запроса, то надо начать активацию сразу после прямоугольника участника. |
| Самоудаление |  | Удаление участника обозначается большим крестом (X). X в конце линии жизни показывает, что участник удаляет сам себя. |
| Удаление из другого объекта |  | Удаление участника обозначается большим крестом (X). Стрелка сообщения, идущая в X, означает, что один участник явным образом удаляет другого. |

| Термин | Изображение | Описание |
|-----------------------|--|---|
| Фреймы взаимодействия |  | <p>Общая проблема диаграмм последовательности заключается в том, как отображать циклы и условные конструкции. Прежде всего надо усвоить, что диаграмма последовательности для этого не предназначена. Подобные управляющие структуры лучше показывать с помощью диаграммы деятельности или собственно кода. Диаграмма последовательности применяется для визуализации процесса взаимодействия объектов, а не как средство моделирования алгоритма управления.</p> <p>Как было сказано, существуют дополнительные обозначения. И для циклов, и для условий используются фреймы взаимодействий (interaction frames), представляющие собой средство разметки диаграммы взаимодействия.</p> |

Диаграммы последовательности

Диаграммы последовательности предназначены для моделирования взаимодействия между несколькими объектами. Д

Диаграммы последовательности создаются для моделирования взаимодействия в рамках одного прецедента.

Также диаграммы последовательности подойдут для моделирования взаимодействия пользователя и Системы в целом.

На уровне детальной спецификации требований диаграммы последовательности используются для моделирования взаимодействия компонентов Системы и пользовательских классов в рамках выбранного прецедента.

На уровне реализации с помощью диаграммы последовательности моделируется взаимодействие между отдельными компонентами Системы. На данном уровне детализации лучше подойдет диаграмма коммуникации.

Поведенческие диаграммы

- **Диаграммы коммуникаций (кооперации)** – это особый вид диаграмм взаимодействия, акцентированных на обмене данными между различными участниками взаимодействия.
- **Коммуникационные диаграммы** допускают произвольное размещение участников, позволяя рисовать связи, показывающие отношения участников, и использовать нумерацию для представления последовательности сообщений.



Диаграмма кооперации

Поведение системы описывается на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь определенной цели или реализовать некоторый вариант использования.

Кооперация. (*collaboration*) - служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы.

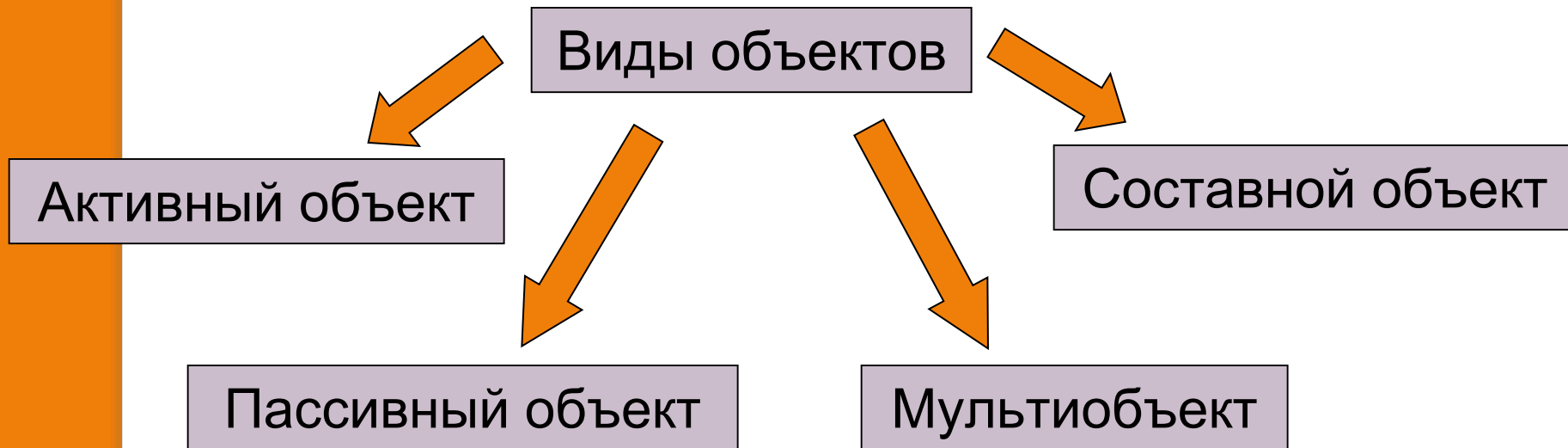
Основные компоненты

Основные компоненты диаграммы
кооперации:

- **объекты;**
- **связи;**
- **сообщения.**

Объекты

- **Объект** является отдельным экземпляром класса, который создается на этапе реализации модели (выполнения программы)



Мультиобъект

- Представляет собой множество объектов, которые могут быть образованы на основе класса.

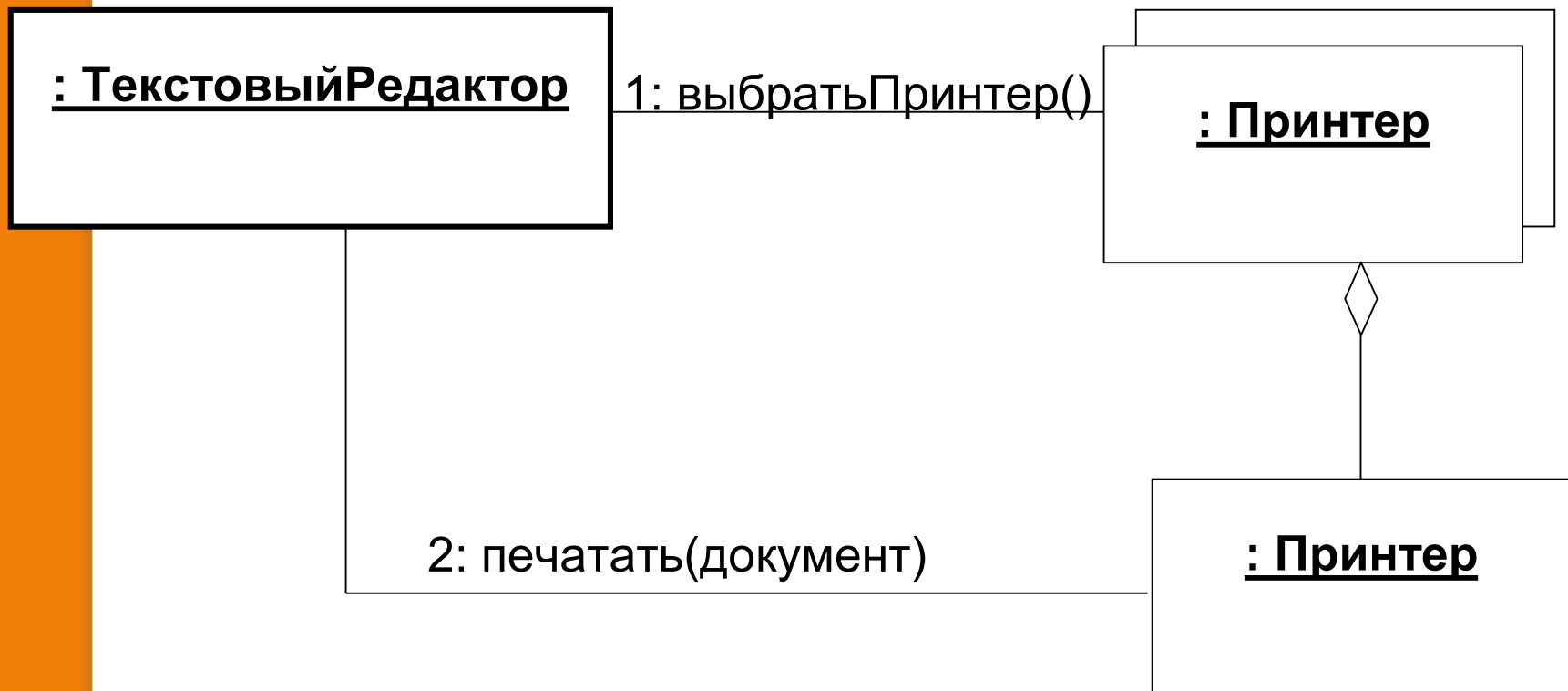


The diagram consists of two overlapping rectangular boxes. The front box is white with a black border and contains the text **: Мультиобъект**. The back box is also white with a black border and is partially obscured by the front box.

: Мультиобъект

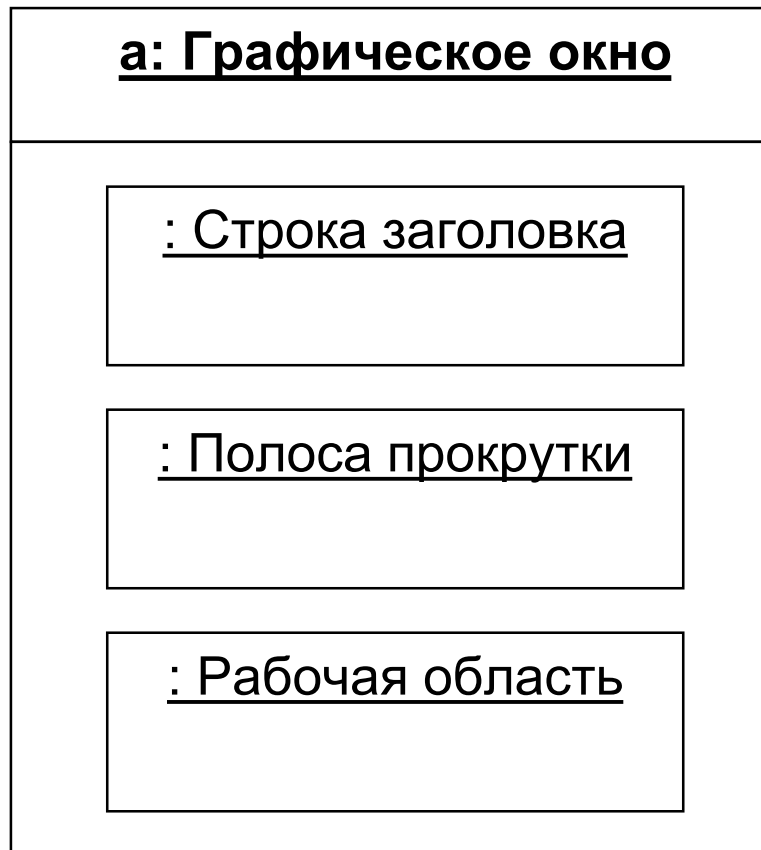
Активный объект

- В контексте языка UML объекты делятся на активные и пассивные.
- *Активный объект* имеет свой собственный поток управления и может инициировать деятельность по управлению другими объектами.



Составной объект

- Предназначен для представления объекта, имеющего сложную структуру и внутренние потоки управления.



Сообщение

- В общем смысле под сообщением понимается *законченный фрагмент информации*, посылаемый одним объектом другому.



1)

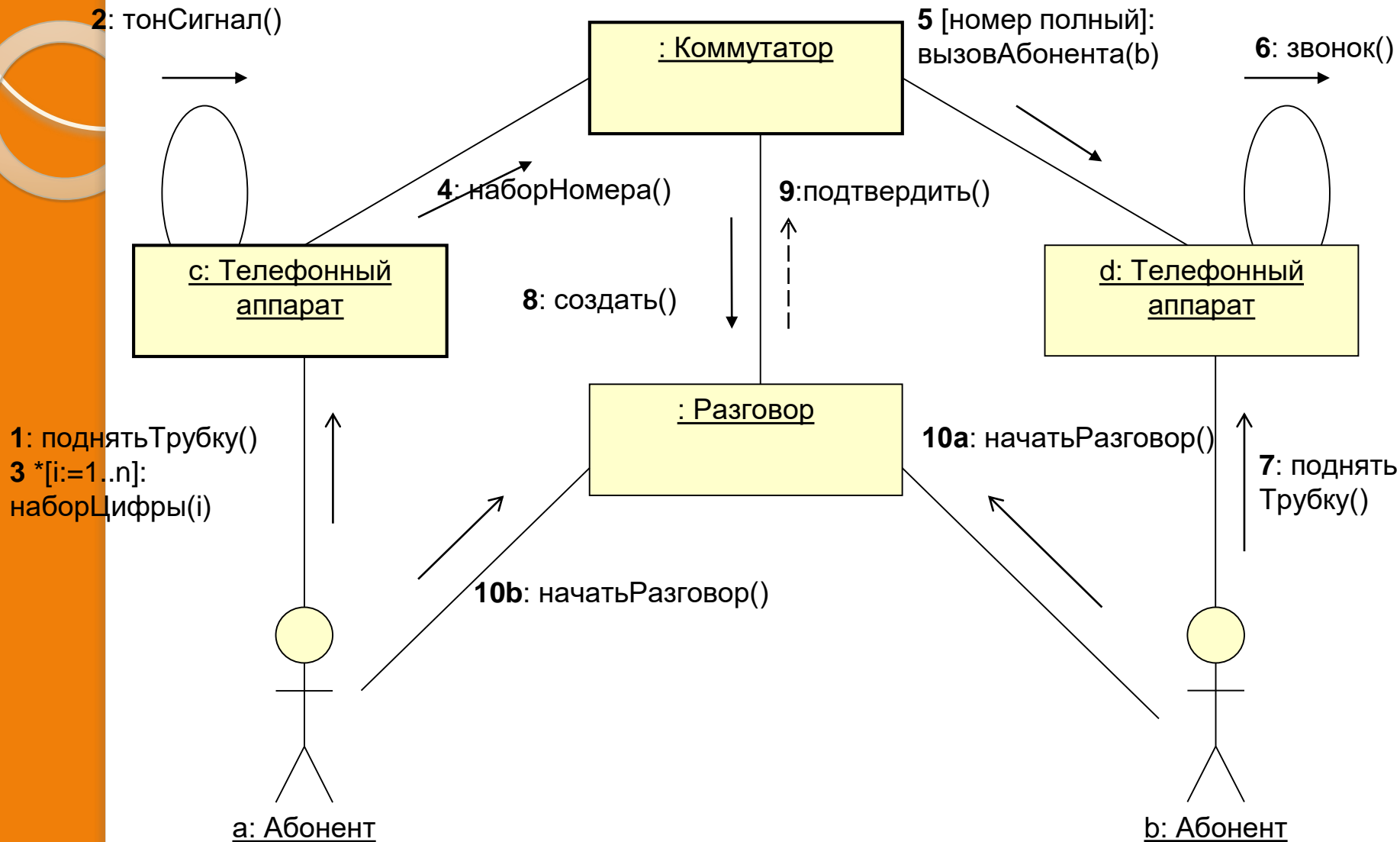


2)



3)

Пример диаграммы кооперации



Основное отличие между диаграммой последовательности и кооперации

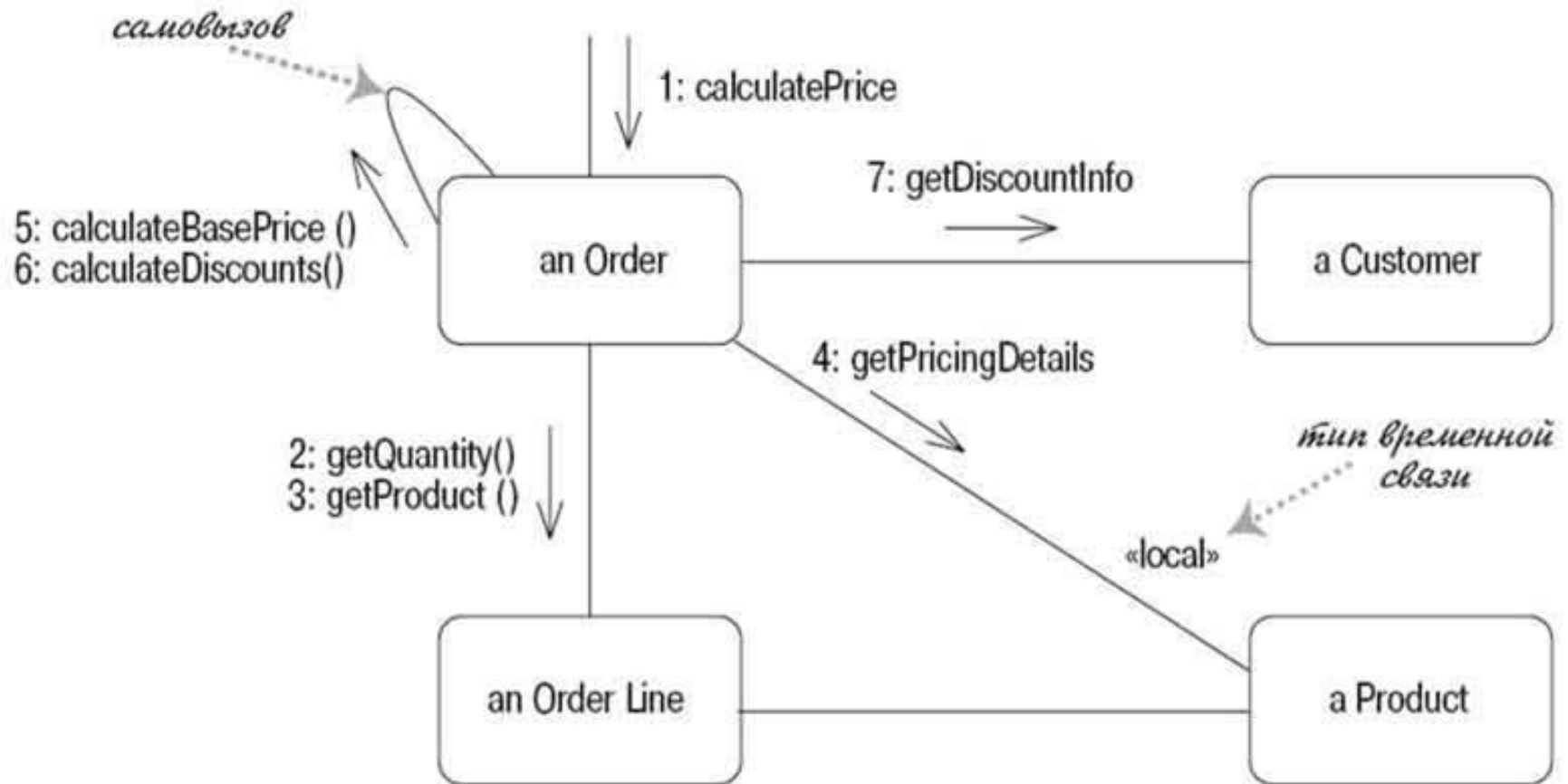
- На **диаграмме кооперации** изображаются только такие отношения между объектами, которые играют роль информационных каналов при взаимодействии.
- На **диаграмме кооперации** не указывается время в виде дополнительного измерения.
- Таким образом, в диаграмме последовательности делается акцент на временной аспект, в диаграмме кооперации – на статическое взаимодействие объектов системы.

На диаграмме коммуникации отображаются связи:

- связей, которые представляют собой экземпляры ассоциаций
можно также показать временные связи, возникающие только в контексте взаимодействия:
- связь «*local*» (локальная) от объекта *Order* (Заказ) к объекту *Product* (Продукт) – это локальная переменная,
- «*parameter*» (параметр)
- «*global*» (глобальная)

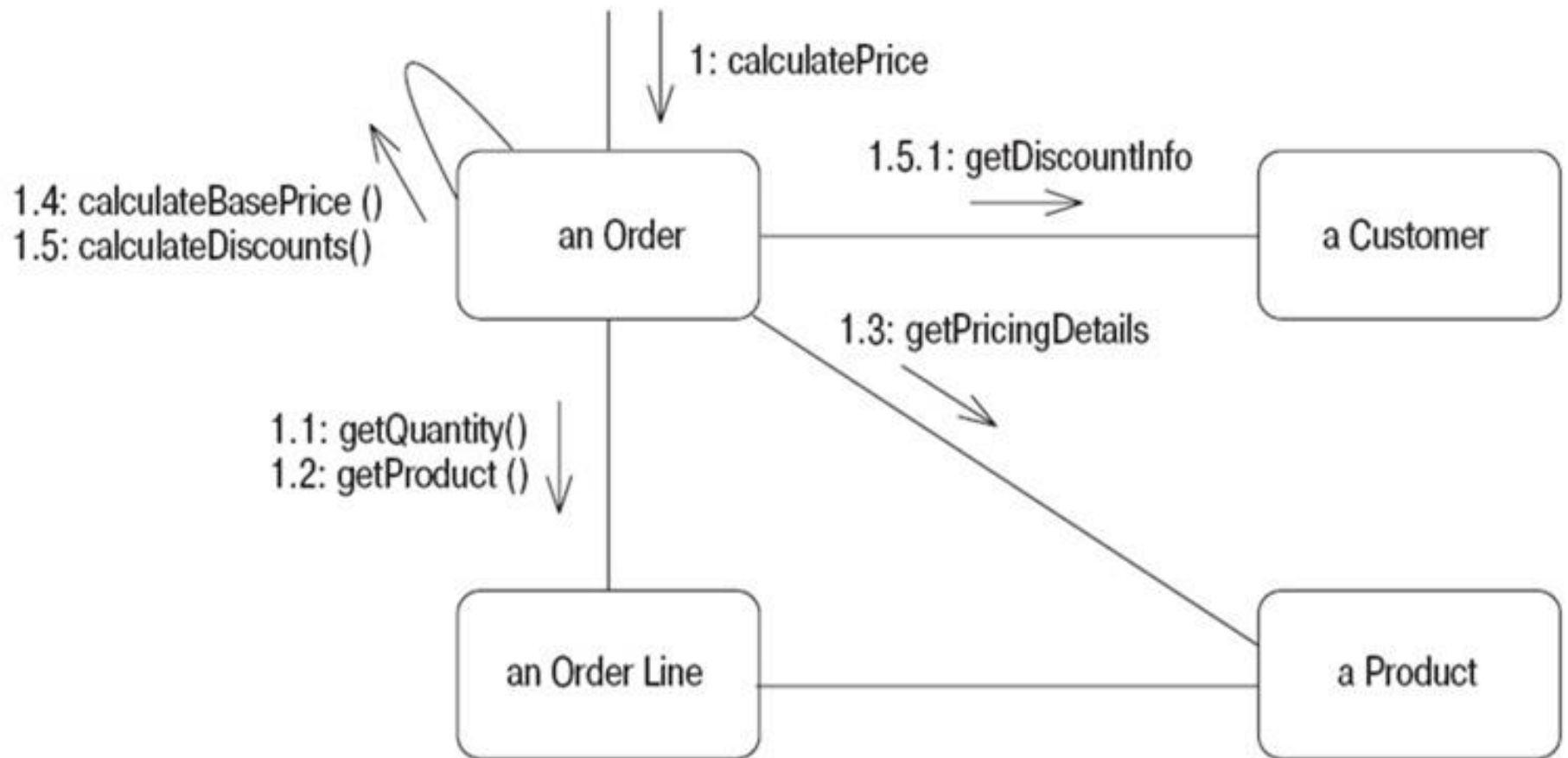
Стиль простой нумерации в языке UML не разрешен.

В соответствии с правилами UML необходимо придерживаться вложенной десятичной нумерации.



Вложенная десятичная нумерация нужна, потому что требуется исключить неопределенность при самовывозах.

Метод *getDiscountInfo* вызывается из метода *calculateDiscount*.



Диаграммы взаимодействия

- Диаграммами взаимодействия следует пользоваться в том случае, когда необходимо описать поведение нескольких объектов в рамках одного варианта использования.
- Диаграммы взаимодействия удобны для изображения кооперации объектов и вовсе не так хороши для точного представления их поведения.
- Если необходимо описать поведение единственного объекта во многих вариантах использования, то следует применять диаграмму состояний. Если же возникает необходимость описать поведение, охватывающее несколько вариантов использования или несколько нитей процесса, следует рассматривать диаграмму деятельности.