

Диаграмма последовательности

Назначение диаграммы последовательности

В языке UML взаимодействие элементов рассматривается в информационном аспекте их коммуникации – взаимодействующие объекты обмениваются между собой некоторой информацией.

Это полностью согласуется с принципами ООАП, когда *любые виды информационного взаимодействия между элементами системы должны быть сведены к отправке и приему сообщений между ними.*

Назначение диаграммы последовательности

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы *взаимодействия*.

Рассматриваются два аспекта взаимодействия:

- взаимодействия объектов можно рассматривать во времени — для этого используются диаграммы последовательности.
- можно рассматривать структурные особенности взаимодействия объектов — для этого используются диаграммы кооперации.

Объекты

На диаграмме последовательности изображаются те объекты, которые непосредственно участвуют во взаимодействии и *не показываются статические ассоциации с другими объектами.*

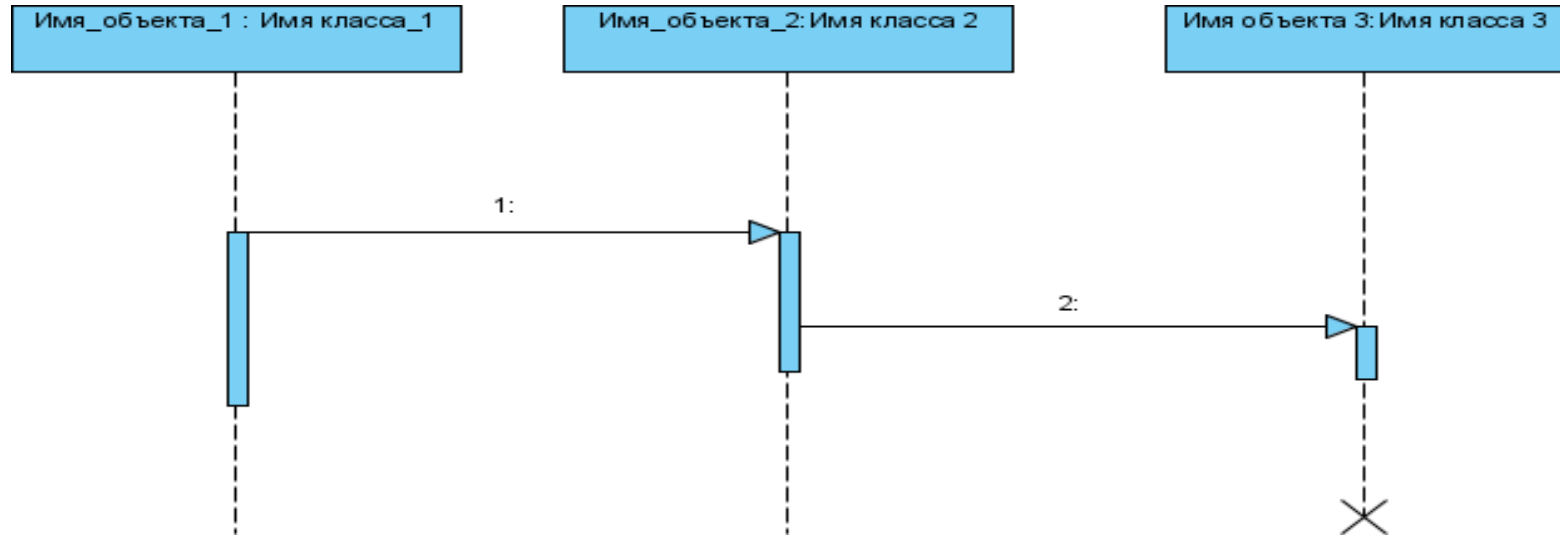
Диаграмма последовательности имеет как бы два измерения.

Первое измерение – слева направо в виде вертикальных линий каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии.

Крайний слева объект – инициатор взаимодействия. Правее изображается другой объект, который непосредственно

Объекты

Графически каждый объект изображается прямоугольником, внутри которого записываются имя объекта и имя класса, разделенные двоеточием.



Объекты

Второе измерение диаграммы последовательности – вертикальная временная ось, направленная сверху вниз.

Начальному моменту времени соответствует самая верхняя часть диаграммы.

Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения.

Сообщения, расположенные на диаграмме последовательности выше иницируются раньше тех, которые расположены ниже

Линия жизни объекта

Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, связанной с единственным объектом на диаграмме последовательности.

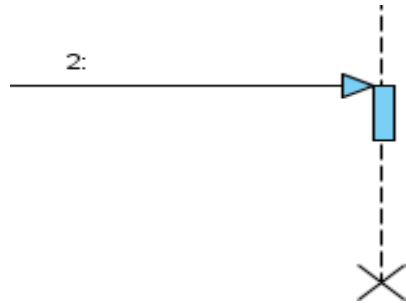
Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе – может участвовать во всех ее взаимодействиях.

Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности.

Линия жизни объекта

Отдельные объекты, выполнив свою роль в системе, могут быть разрушены, чтобы освободить занимаемые ими ресурсы.

Для таких объектов линия жизни обрывается в момент его уничтожения. В языке UML для этого используется специальный символ в форме латинской буквы «X».



**Ниже этого символа
пунктирная линия не
изображается**

Фокус управления

В процессе функционирования объектно-ориентированных систем объекты могут находиться в активном состоянии или в состоянии пассивного ожидания.

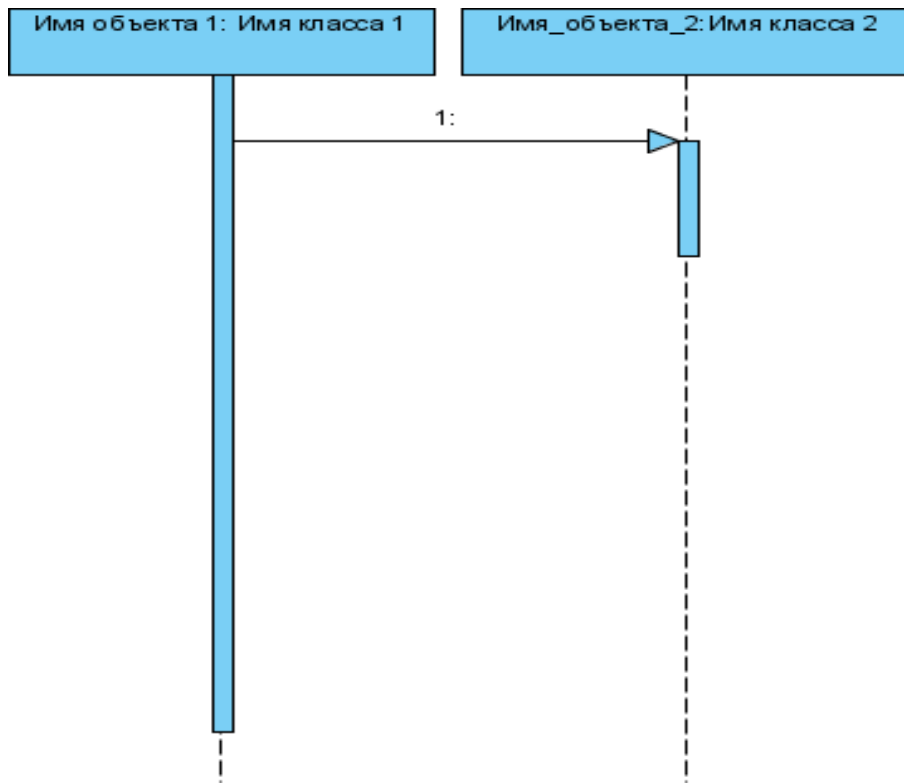
Для выделения подобной активности объектов в языке UML применяется понятие *фокуса управления* (focus of control).

Фокус управления изображается в форме вытянутого узкого прямоугольника. Верхняя сторона – начало активности, нижняя сторона – окончание активности.

Данный прямоугольник может заменить линию жизни

Фокус управления

Пример постоянно
активного объекта:



Фокус управления

Периоды активности объекта могут чередоваться с периодами его пассивности (ожидания).

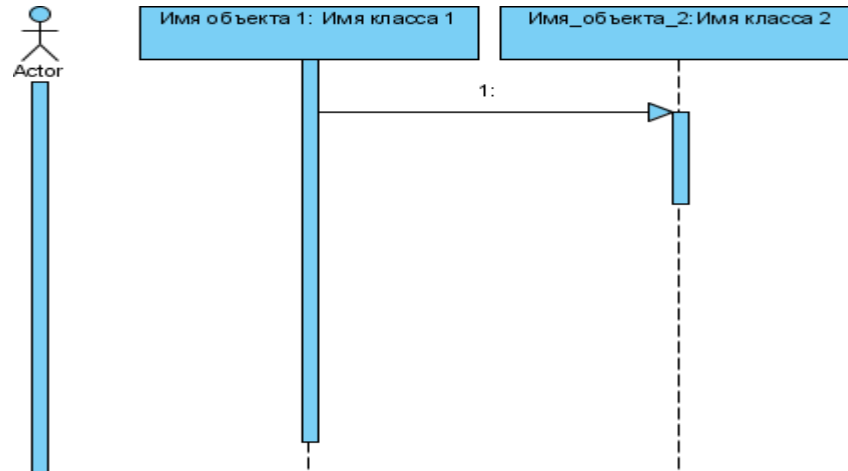
В этом случае у такого объекта имеется несколько фокусов управления.

Получить фокус управления может только существующий объект, у которого в этот момент имеется линия жизни.

Если некоторый объект был уничтожен, то вновь возникнуть в системе он уже не может.

Фокус управления

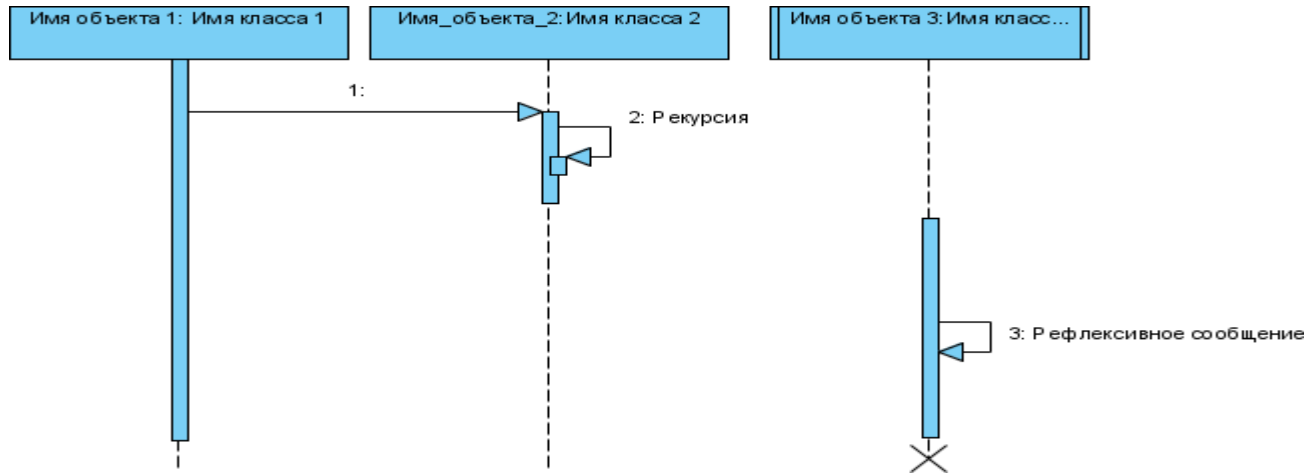
Инициатором взаимодействия в системе может быть актер или внешний пользователь. В этом случае актер изображается на диаграмме последовательности самым первым объектом слева со своим фокусом управления.



Фокус управления

Иногда некоторый объект может инициировать рекурсивное взаимодействие с самим собой.

На диаграмме последовательности рекурсия обозначается небольшим прямоугольником, присоединенным к правой стороне фокуса управления.



Сообщения

Каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой.

Сообщение (message) – законченный фрагмент информации, который отправляется одним объектом другому.

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

Сообщения

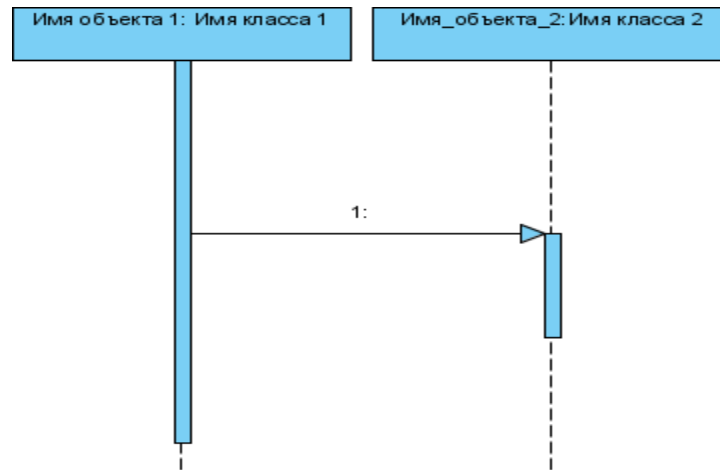
Иногда отправителя сообщения называют клиентом, а получателя – сервером.

При этом сообщение от клиента имеет форму запроса некоторого сервиса, а реакция сервера на запрос после получения сообщения может быть связана с выполнением определенных действий или передачи клиенту необходимой информации тоже в форме сообщения.

В языке UML могут встречаться несколько разновидностей сообщений, каждое из которых имеет свое графическое изображение.

Сообщения

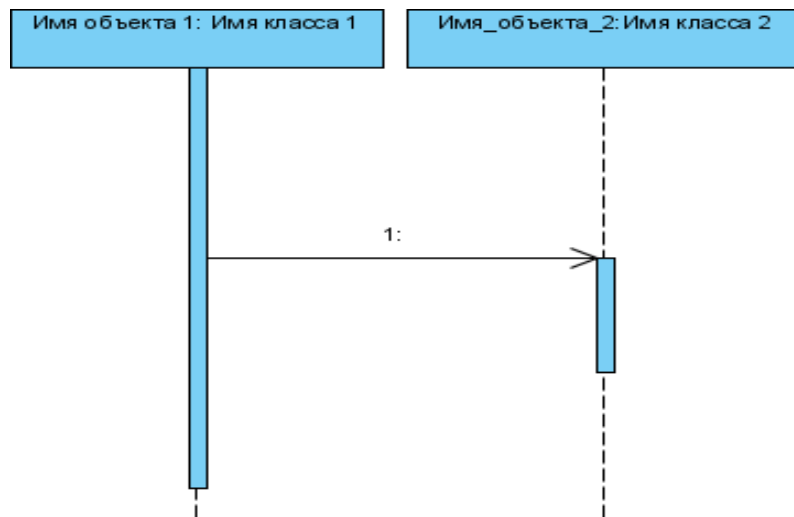
Первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления:



Сообщения

Вторая разновидность явно обозначает асинхронное сообщение между двумя объектами в некоторой процедурной последовательности.

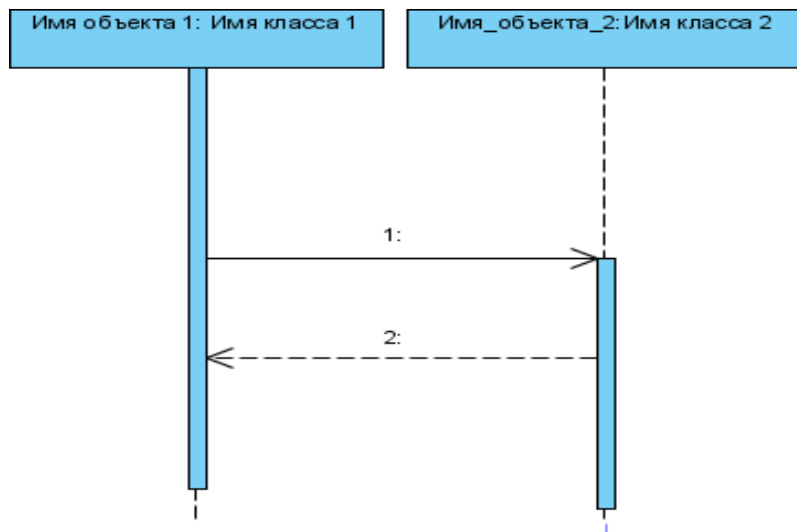
Пример: прерывание операции при возникновении исключительной ситуации.



Сообщения

Третья разновидность сообщения используется для возврата из вызова процедуры.

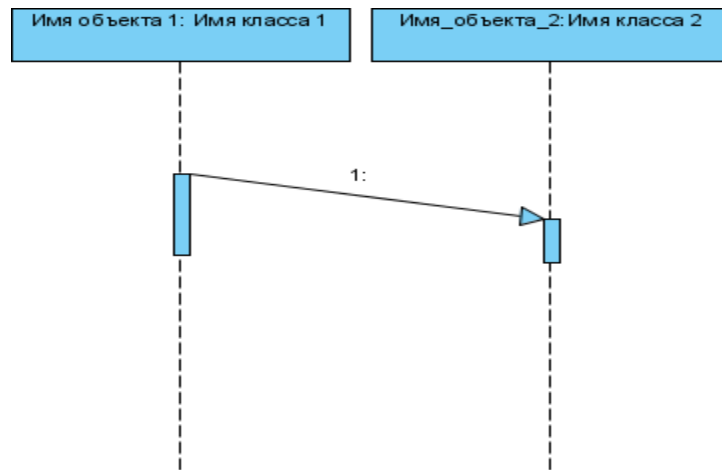
Пример: сообщение о завершении некоторых вычислений без предоставления результата расчета объекту-клиенту.



Сообщения

Обычно сообщения изображаются горизонтальными стрелками. При этом предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами.

Если это не так, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.



Сообщения

В отдельных случаях объект может посылать сообщения самому себе, инициируя *рефлексивные сообщения*.

Такие сообщения изображаются прямоугольником со стрелкой, начало и конец которой совпадают.

Примеры: обработка нажатий на клавиши клавиатуры при вводе текста, при наборе цифр номера телефона абонента.

Стереотипы сообщений

В языке UML предусмотрены стандартные действия, выполняемые в ответ на получение соответствующего сообщения.

Они могут быть явно указаны на диаграмме последовательности в форме стереотипа.

Используются следующие обозначения для моделирования действий:

- «call» (вызывать) – сообщение, требующее вызова операции или процедуры принимающего объекта;

Стереотипы сообщений

- «create» (создать) – сообщение, требующее создания другого объекта для выполнения определенных действий.
Созданный объект может получить фокус управления, а может и не получить его;
- «destroy» (уничтожить) – сообщение с явным требованием уничтожить соответствующий объект;
- «send» (послать) – обозначает посылку другому объекту некоторого сигнала, который асинхронно инициируется одним объектом и принимается другим.

Временные ограничения

В отдельных случаях выполнение тех или иных действий на диаграмме последовательности может потребовать явной спецификации *временных ограничений*, накладываемых на сам интервал выполнения операций или передачу сообщений.

В языке UML для записи временных ограничений используются фигурные скобки.

Примеры временных ограничений:

- {время_приема_сообщения < 1 сек.};

Пример построения диаграммы последовательности

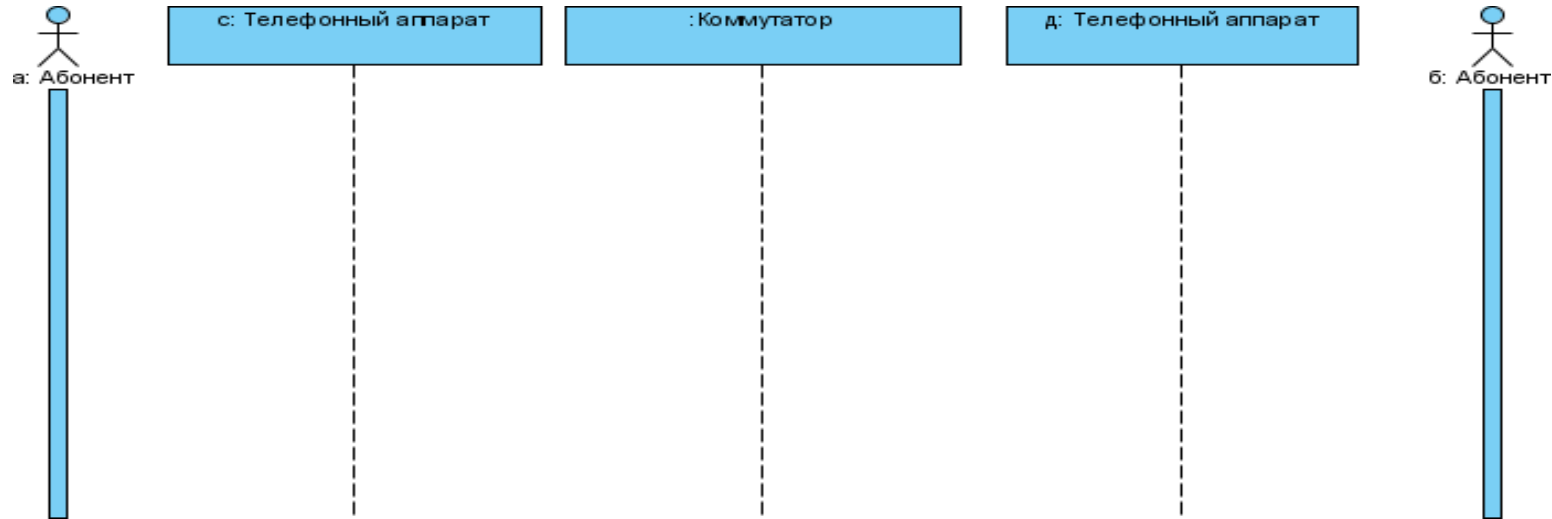
В качестве примера рассматривается построение диаграммы последовательности для моделирования процесса телефонного разговора с использованием обычной телефонной сети.

Объектами являются: два абонента **а** и **б**, два телефонных аппарата **с** и **д**, коммутатор и сам разговор как объект моделирования.

Коммутатор и разговор являются анонимными объектами.

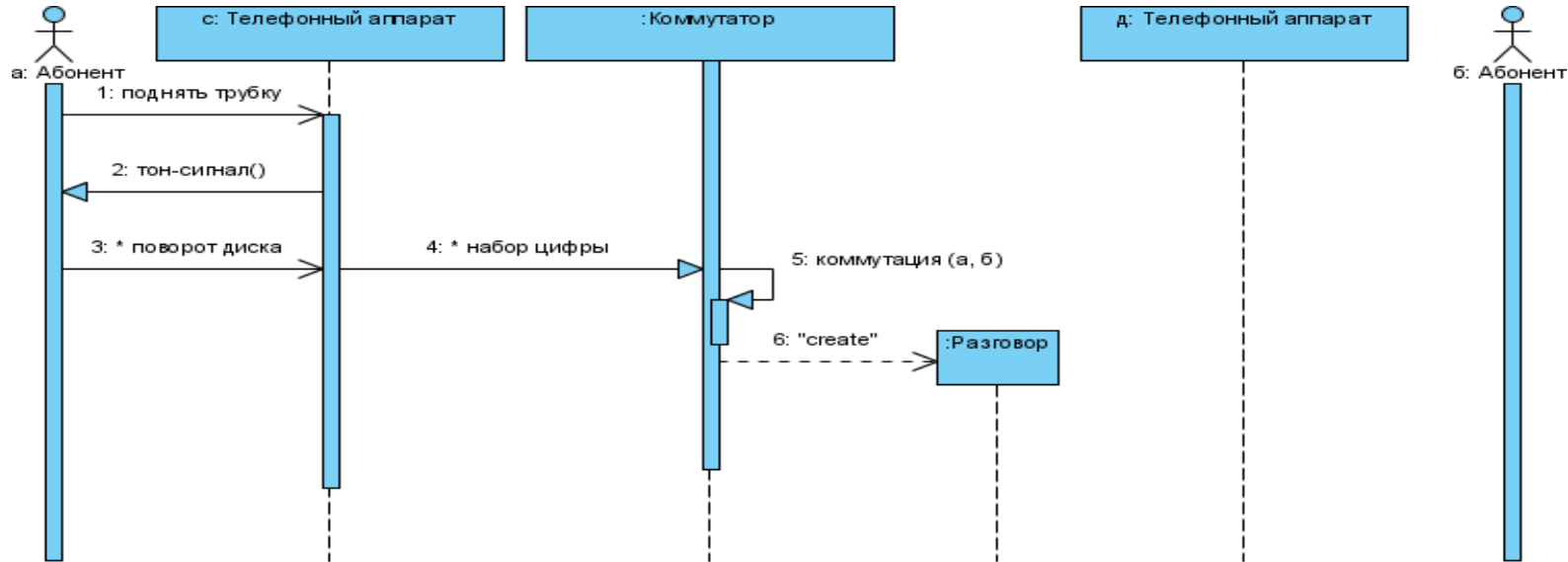
Пример построения диаграммы последовательности

На первом этапе располагаем выбранные объекты на предполагаемой диаграмме. Абоненты рассматриваются как актеры.



Пример построения диаграммы последовательности

Разговор как объект появляется только после установки соединения и уничтожается с его прекращением.



Пример построения диаграммы последовательности

