

Элементы графической нотации диаграммы последовательности

Аннотация: Назначение диаграммы последовательности. Объекты, их графическое представление. Линия жизни и фокус управления. Особенности изображения моментов создания и уничтожения объектов. Ветвление и условия их выполнения. Рекомендации по построению диаграмм последовательности.

Ключевые слова: [диаграмма последовательности](#), [значение](#), [объект](#), [UML](#), [сообщение](#), [график](#), [линия жизни](#), [имя класса](#), [запись](#), [анонимный](#), [специальный символ](#), [производительность](#), [прямоугольник](#), [фокус управления](#), [пассивное ожидание](#), [актер](#), [пользователь](#), [активность](#), [подпроцесс](#), [рекурсия](#), [параметры операций](#), [ветвление](#), [альтернативные](#), [выражение](#), [булевское выражение](#), [конфликт](#), [условие ветвления](#), [текущий счет](#), [семантика](#), [синтаксис](#), [операции](#), [время выполнения](#), [поток](#), [визуализация](#), [язык объектных ограничений ocl](#)

Диаграмма последовательности (sequence diagram) - диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления.

Особенности взаимодействия элементов моделируемой системы могут быть представлены на диаграммах кооперации и последовательности. Диаграммы кооперации используются для спецификации динамики поведения систем, хотя время в явном виде в них отсутствует. Однако временной аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействие объектов. Именно для этой цели в языке UML используются диаграммы последовательности, которые и станут предметом изучения в настоящей лекции.

На диаграмме последовательности неявно присутствует ось времени, что позволяет визуализировать временные отношения между передаваемыми сообщениями. С помощью диаграммы последовательности можно представить взаимодействие элементов модели как своеобразный временной график "жизни" всей совокупности объектов, связанных между собой для реализации варианта использования программной системы, достижения бизнес-цели или выполнения какой-либо задачи.

Объекты и их изображение на диаграмме последовательности

На диаграмме последовательности также изображаются объекты, которые непосредственно участвуют во взаимодействии, при этом никакие статические связи с другими объектами не визуализируются. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения. Одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Второе измерение диаграммы последовательности - вертикальная временная ось, направленная сверху вниз.

Каждый объект графически изображается в форме прямоугольника и располагается в верхней части своей линии жизни (рис. 3-4.1). Внутри прямоугольника записываются собственное имя объекта со строчной буквы и имя класса, разделенные двоеточием. При этом вся запись подчеркивается, что является признаком объекта, который, как указывалось ранее, представляет собой экземпляр класса.

Для объектов диаграммы последовательности остаются справедливыми правила именования, рассмотренные ранее применительно к диаграммам кооперации. Если на диаграмме последовательности отсутствует собственное имя объекта, то при этом должно быть указано имя класса. Такой объект считается **анонимным**. Может отсутствовать и имя класса, но при этом должно быть указано собственное имя объекта. Такой объект считается **сиротой**. Роль классов в именах объектов на диаграммах последовательности, как правило, не указывается.

Крайним слева на диаграмме изображается объект - инициатор моделируемого процесса взаимодействия (объект **a** на рис. 3-4.1). Правее - другой объект, который непосредственно взаимодействует с первым. Таким образом, порядок расположения объектов на диаграмме последовательности определяется исключительно соображениями удобства визуализации их взаимодействия друг с другом.

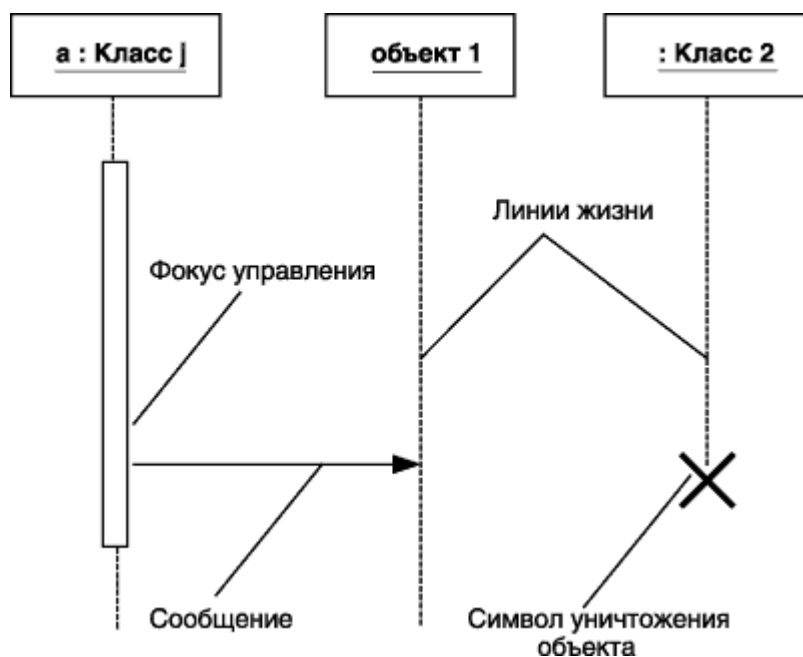


Рис. 3-4.1. Графические элементы диаграммы последовательности

Начальному моменту времени соответствует самая верхняя часть диаграммы. При этом процесс взаимодействия объектов реализуется посредством *сообщений*, которые посылаются одними объектами другим. *Сообщения* изображаются в виде горизонтальных стрелок с именем сообщения и образуют определенный порядок относительно времени своей инициализации. Другими словами, сообщения, расположенные на диаграмме последовательности выше, передаются раньше тех, которые расположены ниже. При этом масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа "раньше-позже".

Линия жизни объекта (object lifeline) - вертикальная линия на диаграмме последовательности, которая представляет существование объекта в течение определенного периода времени.

Линия жизни объекта изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей рабочей области диаграммы последовательности от самой верхней ее части до самой нижней (объект 1 и анонимный объект Класса 2 на рис. 3-4.1).

Отдельные *объекты*, закончив выполнение своих операций, могут быть уничтожены, чтобы освободить занимаемые ими ресурсы. Для таких *объектов* *линия жизни* обрывается в момент его уничтожения. Для обозначения момента уничтожения *объекта* в языке *UML* применяется *специальный символ* в форме латинской буквы "X". На [рис. 3-4.2](#) этот символ используется для уничтожения анонимного *объекта*, образованного от Класса 3. Ниже этого символа пунктирная линия не изображается, поскольку соответствующего *объекта* в системе уже нет, и этот *объект* должен быть исключен из всех последующих взаимодействий.

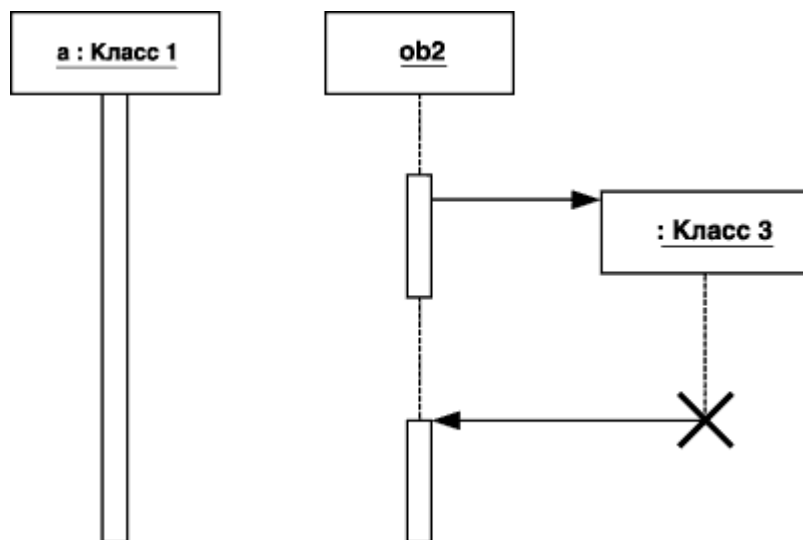


Рис. 3-4.2. Графическое изображение линий жизни и фокусов управления объектов

Вовсе не обязательно создавать все *объекты* в начальный момент времени. Отдельные *объекты* в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее *производительность*. В этом случае *прямоугольник* такого *объекта* изображается не в верхней части *диаграммы последовательности*, а в той, которая соответствует моменту создания *объекта* (*анонимный объект*, образованный от Класса 3 на рис. 3-4.2). При этом *прямоугольник* *объекта* вертикально располагается в том месте *диаграммы*, которое по оси времени совпадает с моментом его возникновения в системе. *Объект* создается со своей *линией жизни* *a*, возможно, и с *фокусом управления*.

В процессе функционирования объектно-ориентированных систем одни *объекты* могут находиться в активном состоянии, непосредственно выполняя определенные действия, или в состоянии *пассивного ожидания сообщений* от других *объектов*. **Фокус управления** - символ, применяемый для того, чтобы явно выделить подобную активность *объектов* на *диаграммах последовательности*.

Фокус управления (focus of control) - специальный символ на *диаграмме последовательности*, указывающий период времени, в течение которого *объект* выполняет некоторое действие, находясь в активном состоянии.

Фокус управления изображается в форме вытянутого узкого *прямоугольника* (*объект a* на рис. 3-4.1), верхняя сторона которого обозначает начало получения *фокуса управления объектом* (начало активности), а ее нижняя сторона - окончание *фокуса управления* (окончание активности). Этот *прямоугольник* располагается ниже обозначения соответствующего *объекта* и может заменять его *линию жизни* (*объект a* на рис. 3-4.2), если на всем ее протяжении он активен.

Периоды активности *объекта* могут чередоваться с периодами его пассивности или ожидания. В этом случае у такого *объекта* *фокусы управления* изменяют свое изображение на *линию жизни* и

наоборот (*объект* сирота **ob2** на рис. 3-4.2). Важно понимать, что получить *фокус управления* может только *объект*, у которого в этот момент имеется *линия жизни*. Если же *объект* был уничтожен, то вновь возникнуть в системе он уже не может. Вместо него может быть создан лишь экземпляр этого же класса, который, строго говоря, будет другим *объектом*.

В отдельных случаях инициатором взаимодействия в системе может быть *актер* или внешний *пользователь*. При этом *актер* изображается на *диаграмме последовательности* самым первым *объектом* слева со своим *фокусом управления* (рис. 3-4.3). Наиболее часто *актер* и его *фокус управления* будут существовать в системе постоянно, отмечая характерную для пользователя *активность* в иницировании взаимодействий с системой. *Актер* может иметь собственное имя либо оставаться анонимным.

В отдельных случаях *объект* может посылать *сообщения* самому себе, иницируя так называемые **рефлексивные сообщения**. Для этой цели служит специальное изображение (*сообщение у объекта а* на рис. 3-4.3). Такие *сообщения* изображаются в форме *сообщения*, начало и конец которого соприкасаются с *линией жизни* или *фокусом управления* одного и того же *объекта*. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Если в результате рефлексивного *сообщения* создается новый *подпроцесс* или нить управления, то говорят о рекурсивном или вложенном *фокусе управления*. На *диаграмме последовательности* рекурсия обозначается небольшим прямоугольником, присоединенным к правой стороне *фокуса управления* того *объекта*, для которого изображается данное рекурсивное взаимодействие (*анонимный объект* Класса 2 на рис. 3-4.3).



Рис. 3-4.3. Графическое изображение актера, рефлексивного сообщения и рекурсии на диаграмме последовательности

Сообщения на диаграмме последовательности

Сообщения как элементы языка *UML*, уже рассматривались ранее при изучении диаграммы кооперации (лекция 7). Стрелки *сообщений* изображаются аналогично рассмотренным ранее, но применительно к *диаграммам последовательности* сообщения имеют дополнительные семантические особенности. При этом на *диаграмме последовательности* все *сообщения* упорядочены по времени своей передачи в моделируемой системе, хотя номера у них могут не указываться.

На *диаграммах последовательности* могут присутствовать три разновидности *сообщений*, каждое из которых имеет свое графическое изображение (рис. 3-4.4).

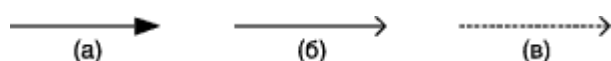


Рис. 3-4.4. Графическое изображение различных видов сообщений между объектами на диаграмме последовательности

Первая разновидность *сообщения* (рис. 3-4.4, а) наиболее распространена и используется для вызова процедур, выполнения операций или обозначения отдельных вложенных потоков управления. Начало этой стрелки, как правило, соприкасается *сфокусом управления* того объекта-клиента, который инициирует это *сообщение*. Конец стрелки соприкасается с *линией жизненного объекта*, который принимает это *сообщение* и выполняет в ответ определенные действия. При этом принимающий *объект* может получить *фокус управления*, становясь в этом случае активным. Передающий *объект* может потерять *фокус управления* или остаться активным.

Вторая разновидность *сообщения* (рис. 3-4.4, б) используется для обозначения простого асинхронного *сообщения*, которое передается в произвольный момент времени. Передача такого *сообщения* обычно не сопровождается получением *фокуса управления* объектом-получателем.

Третья разновидность *сообщения* (рис. 3-4.4, в) используется для возврата из вызова процедуры. Примером может служить простое *сообщение* о завершении вычислений без предоставления результата расчетов объекту-клиенту. В процедурных потоках управления эта стрелка может быть опущена, поскольку ее наличие неявно предполагается в конце активизации *объекта*. В то же время считается, что каждый вызов процедуры имеет свою пару - возврат вызова. Для непроцедурных потоков управления, включая параллельные и асинхронные *сообщения*, стрелка возврата должна указываться явным образом.

Обычно *сообщения* изображаются горизонтальными стрелками, соединяющими *линии жизни* или *фокусы управления* двух объектов на *диаграмме последовательности*. При этом неявно предполагается, что время передачи *сообщения* достаточно мало по сравнению с процессами выполнения действий *объектами*. Считается также, что за время передачи *сообщения* с соответствующими *объектами* не может произойти никаких событий. Другими словами, состояния *объектов* не изменяются. Если же это предположение не может быть признано справедливым, то стрелка *сообщения* изображается под наклоном, так чтобы конец стрелки располагался ниже ее начала.

Каждое *сообщение* на *диаграмме последовательности* ассоциируется с определенной операцией, которая должна быть выполнена принявшим его *объектом*. При этом операция может иметь аргументы или параметры, значения которых влияют на получение различных результатов. Соответствующие *параметры операции* будет иметь и вызывающее это действие *сообщение*. Более того, значения параметров отдельных *сообщений* могут содержать условные выражения, образуя *ветвление* или *альтернативные* пути основного потока управления.

Ветвление потока управления

Одна из особенностей *диаграммы последовательности* - возможность визуализировать простое *ветвление* процесса. Для изображения *ветвления* используются две или более стрелки, выходящие из одной точки *фокуса управления объекта* (*объект ob1* на рис. 3-4.5). При этом

рядом с каждой из них должно быть явно указано соответствующее условие ветви в форме булевского выражения.

Количество ветвей может быть произвольным, однако наличие *ветвлений* может существенно усложнить интерпретацию *диаграммы последовательности*. Предложение-условие должно быть явно указано для каждой ветви и записывается в форме обычного текста, псевдокода или выражения языка программирования. Это *выражение* всегда должно возвращать некоторое *булевское выражение*. Запись этих условий должна исключать одновременную передачу альтернативных *сообщений* по двум и более ветвям. В противном случае на *диаграмме последовательности* может возникнуть *конфликт ветвления*.

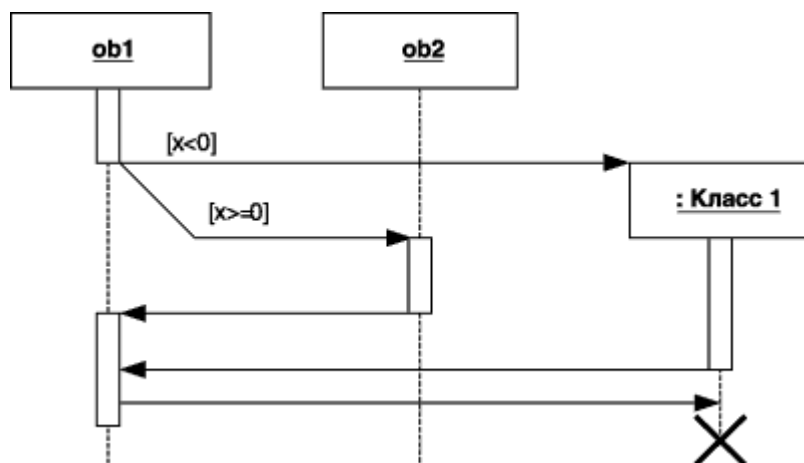


Рис. 3-4.5. Графическое изображение бинарного ветвления потока управления на диаграмме последовательности

С помощью *ветвления* можно изобразить и более сложную логику взаимодействия *объектов* между собой (*объект ob1* на рис. 3-4.6). Если условий более двух, то для каждого из них необходимо предусмотреть ситуацию единственного выполнения. Описанный ниже пример относится к моделированию взаимодействия программной системы обслуживания клиентов в банке. В этом примере *диаграммы последовательности объект ob1* вызывает выполнение действий у одного из трех других *объектов*.

Условием ветвления может служить сумма снимаемых клиентом средств со своего *текущего счета*. Если эта сумма превышает 1500\$, то могут потребоваться дополнительные действия, связанные с созданием и последующим разрушением *объекта* Класса 1. Если же сумма превышает 100\$, но не превышает 1500\$, то вызывается операция или процедура *объекта ob3*. И, наконец, если сумма не превышает 100\$, то вызывается операция или процедура *объекта ob2*. При этом *объекты ob1, ob2 и ob3* постоянно существуют в системе. Последний *объект* создается от Класса 1 только в том случае, если справедливо первое из альтернативных условий. В противном случае он может быть никогда не создан.

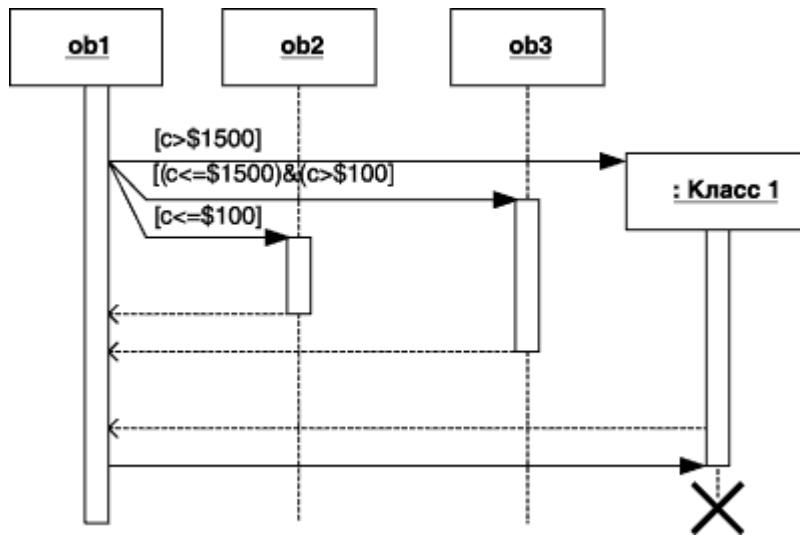


Рис. 3-4.6. Графическое изображение тернарного ветвления потока управления на диаграмме последовательности

Объект *ob1* имеет постоянный фокус управления, а все остальные объекты - получают фокус управления только для выполнения ими соответствующих операций.

На диаграммах последовательности при записи сообщений также могут использоваться стереотипы, рассмотренные ранее при построении диаграммы кооперации (лекция 3 часть 3). Их семантика и синтаксис остаются без изменения, как они определены в нотации языка *UML*. Ниже представлена диаграмма последовательности для описанного выше случая ветвления, дополненная стереотипными значениями отдельных сообщений (рис. 3-4.7). Очевидно, эта диаграмма последовательности является более выразительной и простой для своей содержательной интерпретации.

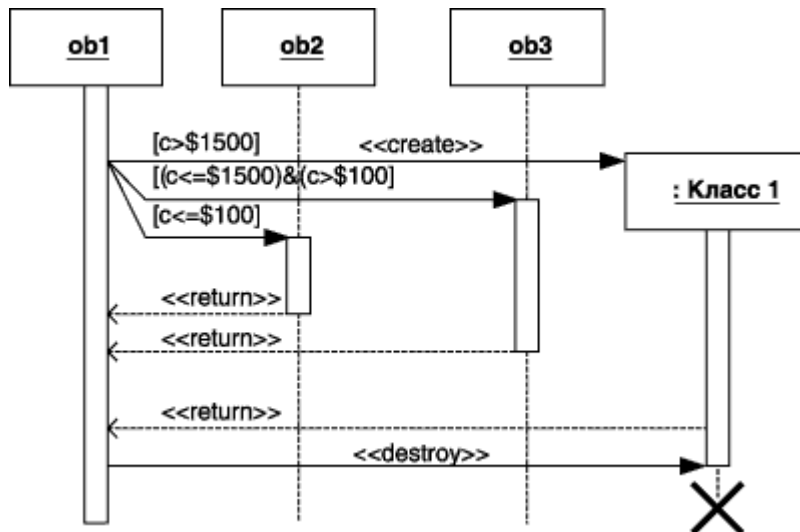


Рис. 3-4.7. Диаграмма последовательности со стереотипными значениями сообщений

Как уже отмечалось ранее, сообщения могут иметь собственное имя, в качестве которого выступает имя операции, вызов которой инициируют эти сообщения у принимающего объекта. В этом случае рядом со стрелкой записывается имя операции с круглыми скобками, в которых могут указываться параметры или аргументы соответствующей операции. Если параметры отсутствуют, то скобки после имени операции все равно должны быть изображены.

Рекомендации по построению диаграмм последовательности

Построение *диаграммы последовательности* целесообразно начинать с выделения из всей совокупности классов только тех, *объекты* которых участвуют в моделируемом взаимодействии. После этого все *объекты* наносятся на диаграмму, с соблюдением порядка инициализации *сообщений*. Здесь необходимо установить, какие *объекты* будут существовать постоянно, а какие временно - только на период выполнения ими требуемых действий.

Когда *объекты* визуализированы, можно приступать к спецификации *сообщений*. При этом необходимо учитывать те *операции*, которые имеют классы соответствующих *объектов* в модели системы. При необходимости уточнения этих операций следует использовать их стереотипы. Для уничтожения *объектов*, которые создаются на *время выполнения* своих действий, нужно предусмотреть явное *сообщение*. Наиболее простые случаи *ветвления* процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. В более сложных случаях для моделирования каждой ветви управления может потребоваться отдельная *диаграмма последовательности*. Следует помнить, что каждый альтернативный *поток* управления затрудняет понимание построенной модели.

Общим правилом является *визуализация* особенностей реализации каждого варианта использования на отдельной *диаграмме последовательности*. В этой ситуации отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия. Необходимость синхронизации сложных потоков управления, как правило, требуют введение в модель дополнительных ограничений. При этом *общая запись* таких ограничений должна следовать семантике *языка объектных ограничений OCL*.