

Проектирование информационных систем

Диаграммы взаимодействий
(Interaction diagram):

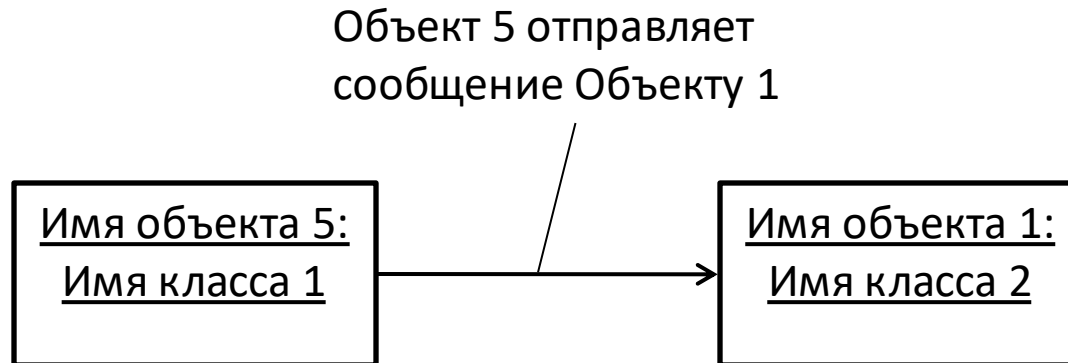
диаграмма последовательностей
(Sequence Diagram)

Классификация диаграмм UML



Взаимодействие элементов в UML

- В UML взаимодействие объектов рассматривается с точки зрения их коммуникации, т.е. обмена некоторой информации.
- Информация представляется в виде законченных сообщений.
- Следовательно объекты отправляют и принимают сообщения друг друга.
- Для отображения последовательности отправки/приема сообщений с учетом времени используется **диаграмма последовательности**.

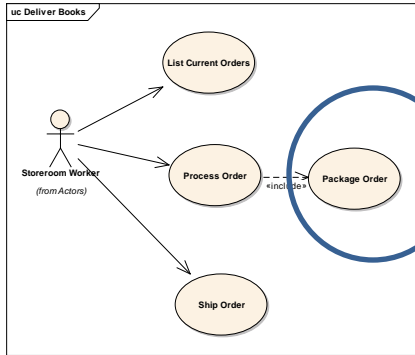


Диаграммы последовательностей (Sequence Diagram)

- **Диаграмма последовательностей** -
графическая модель, которая для
определенного сценария варианта
использования отображает генерируемые
действующими лицами события и их
порядок.
- Отличительная особенность – наличие
времени как самостоятельного измерения.

Связь между анализом и проектированием

Диаграмма Use Case



Какое поведение?

Какая структура?

Диаграмма Последовательности

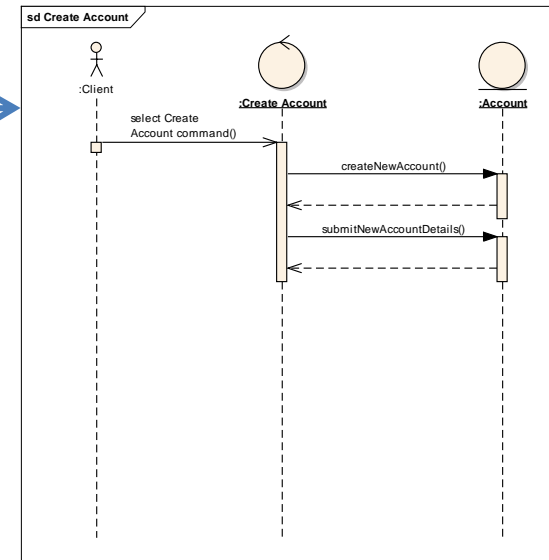
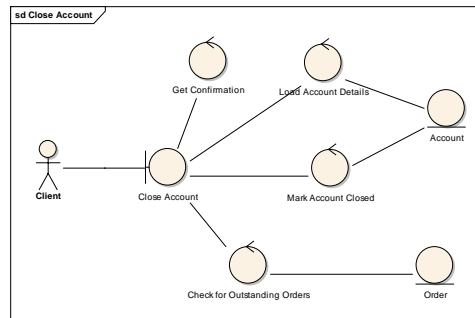


Диаграмма Анализа пригодности



Создание

Диаграмма последовательности.

Структура диаграммы

- Основные элементы диаграммы: **объекты и сообщения**.
- Объекты располагаются в верхней части своей линии жизни. Не исключается ситуация, когда имя объекта может отсутствовать на диаграмме последовательности, а сам объект считается анонимным.
- Крайним слева изображается объект, который является инициатором взаимодействия. Правее изображается другой объект, который непосредственно взаимодействует с первым. Т.о. все объекты образуют некоторый порядок, определяемый их активностью.

Первое измерение - **Объекты и их линии жизни**

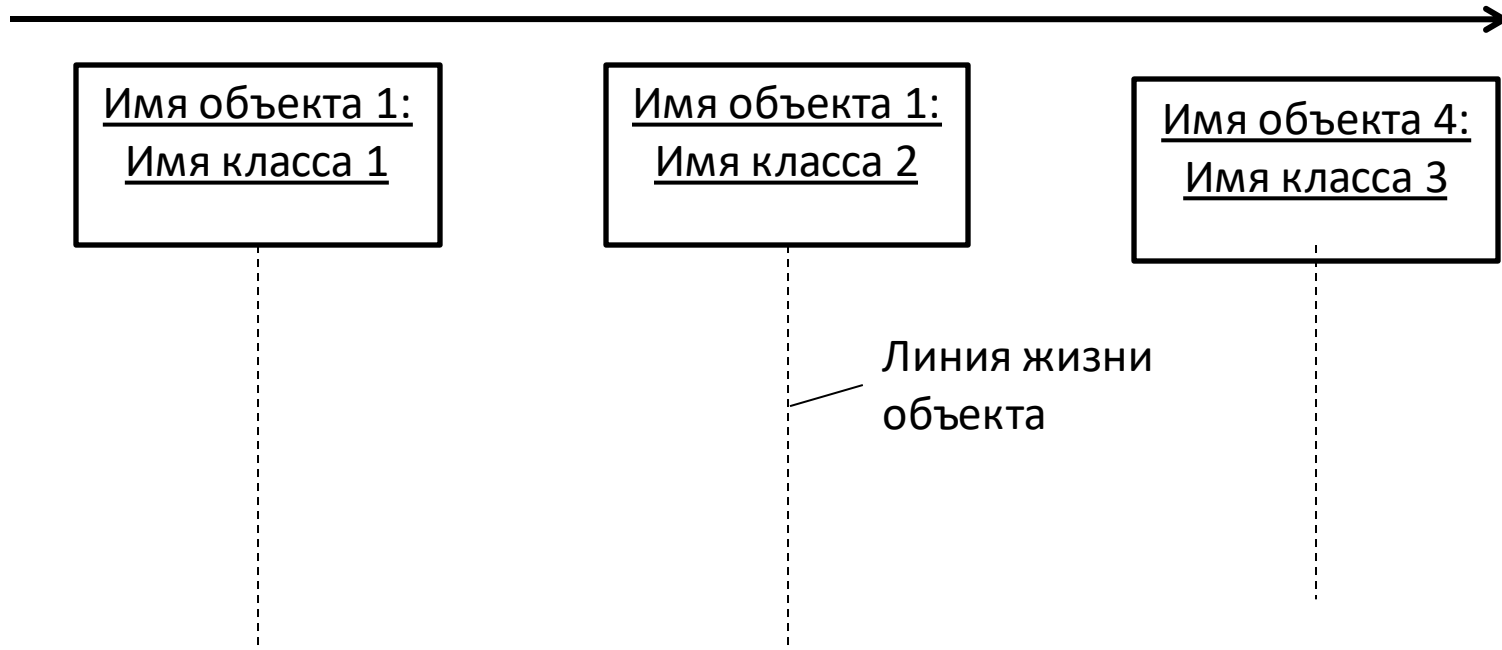


Диаграмма последовательности.

Структура диаграммы. Объекты

- Основные элементы диаграммы: объекты и сообщения.
- Начальному моменту времени соответствует самая верхняя часть диаграммы.
- Взаимодействие объектов реализуется посредством сообщений, которые посылаются одним объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения, и также образуют порядок по времени своего возникновения.
- Сообщения, расположенные выше, инициируются раньше, чем те, которые расположены ниже. Масштаб не указан, упорядоченность «раньше-позже».

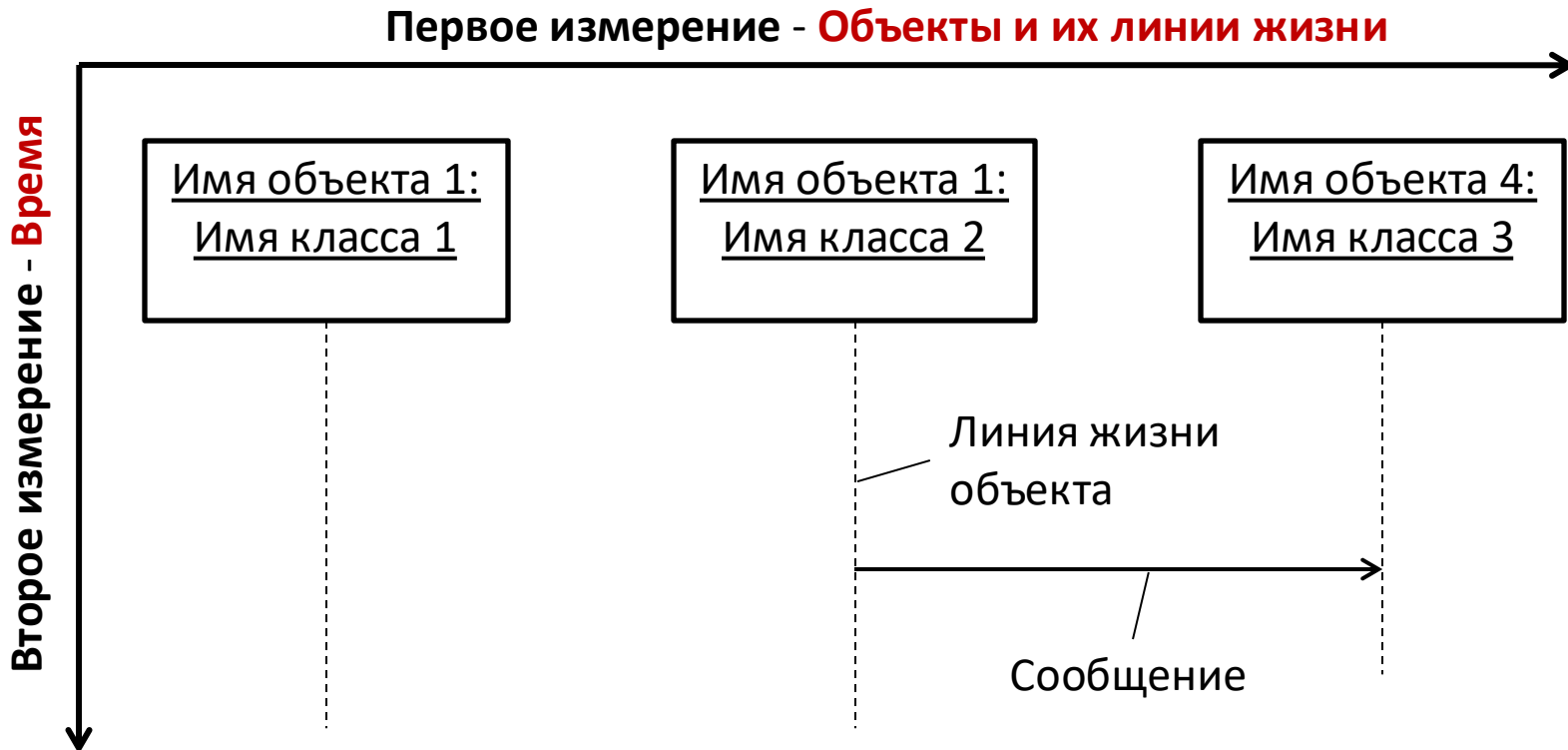


Диаграмма последовательности.

Линия жизни

- **Линия жизни объекта** (object lifeline) 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 .
- Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости от самой верхней ее части до самой нижней.

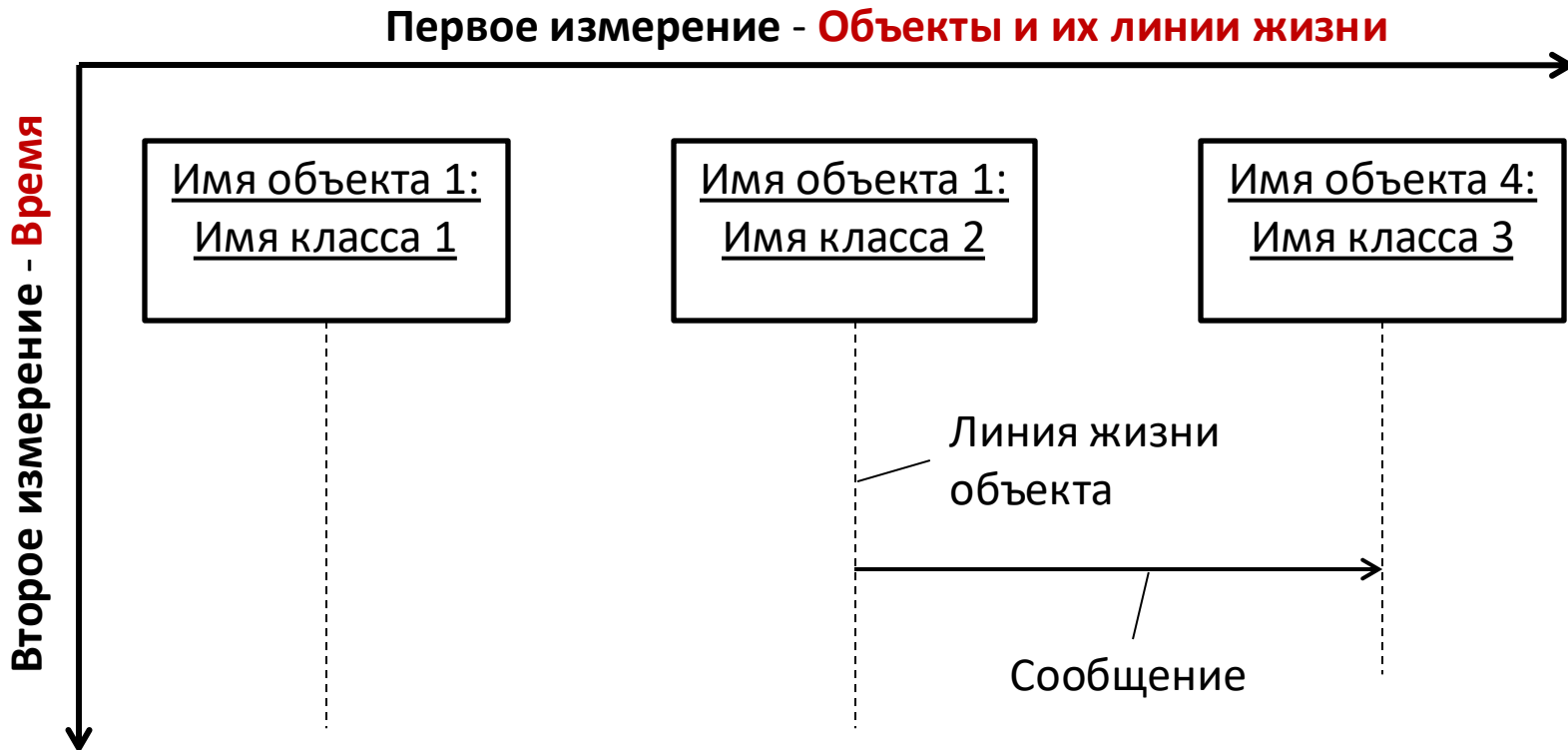


Диаграмма последовательности.

Линия жизни

- Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены (разрушены), чтобы освободить занимаемые ими ресурсы.
- Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта в языке UML используется специальный символ в форме латинской буквы "X".
- Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет.

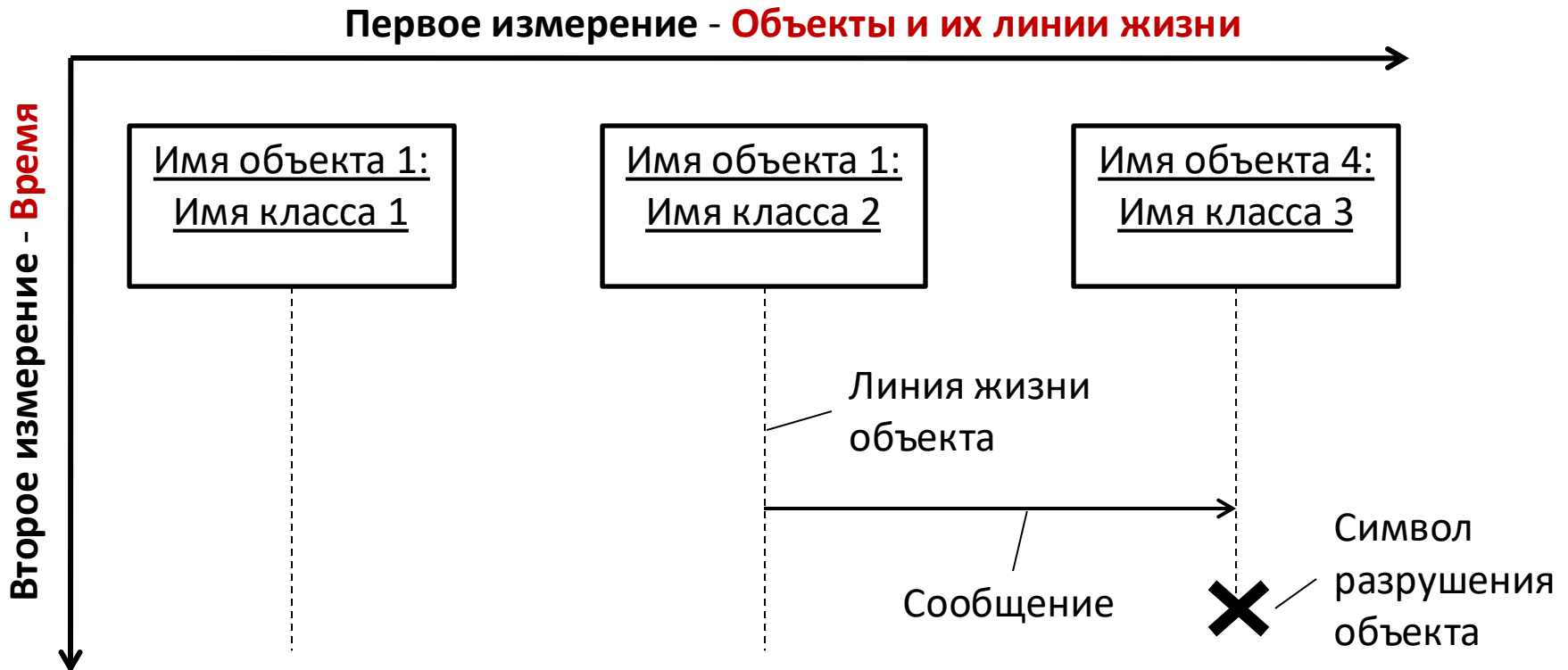


Диаграмма последовательности.

Линия жизни

- Не обязательно создавать все объекты в начальный момент времени. Отдельные объекты в системе могут создаваться по мере необходимости, экономя ресурсы системы.
- В этом случае прямоугольник такого объекта изображается в той части диаграммы, которая соответствует моменту создания объекта.
- При этом прямоугольник объекта вертикально располагается в том месте диаграммы, которое по оси времени совпадает с моментом его возникновения в системе.



Диаграмма последовательности.

Фокус управления

- В процессе функционирования одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия или в состоянии пассивного ожидания сообщений от других объектов.
- Чтобы явно выделить подобную активность объектов, в языке UML применяется специальное понятие, получившее название фокуса управления (focus of control).
- Фокус управления изображается в форме вытянутого узкого прямоугольника, верхняя сторона которого обозначает начало получения фокуса управления объекта (начало активности), а ее нижняя сторона — окончание фокуса управления (окончание активности).
- Этот прямоугольник располагается ниже обозначения соответствующего объекта и может заменять его линию жизни, если на всем ее протяжении он является активным.

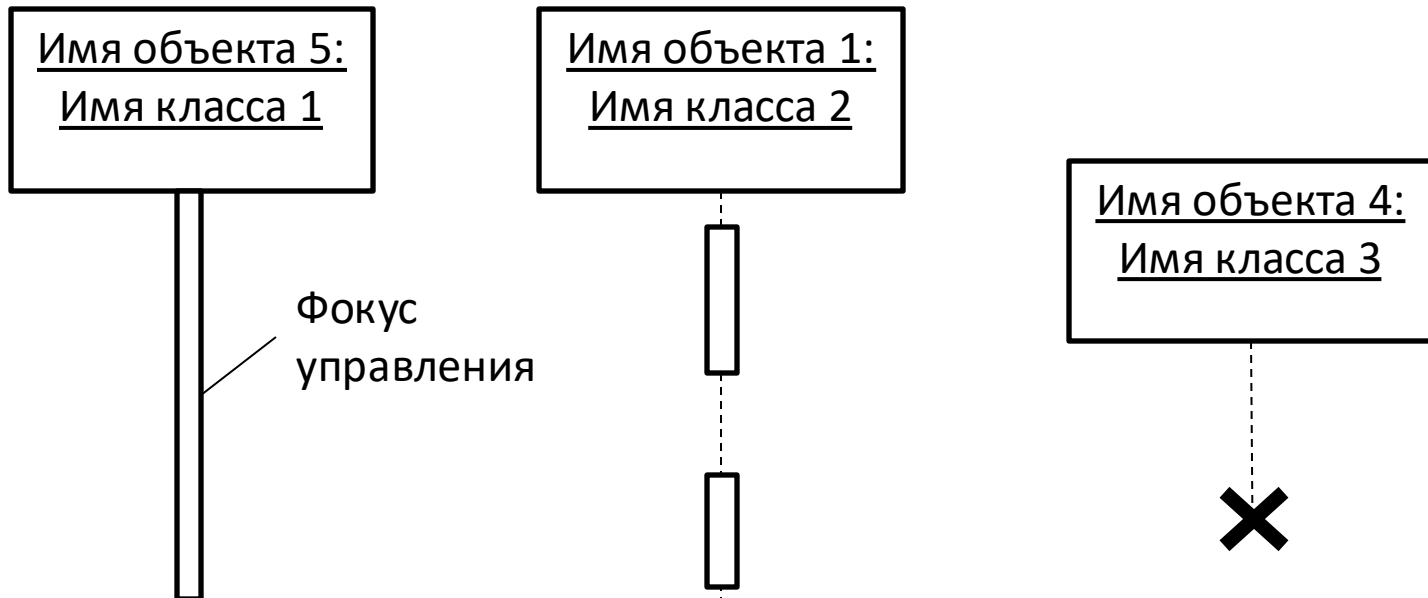


Диаграмма последовательности. Сообщения

- Каждое взаимодействие между объектами описывается совокупностью сообщений, которыми обмениваются объекты между собой.
- В этом смысле **сообщение** (message) представляет собой законченный фрагмент информации, который отправляется одним объектом другому.
- При этом прием сообщения означает:
 - передача некоторой информации;
 - инициирование выполнения определенных действий.

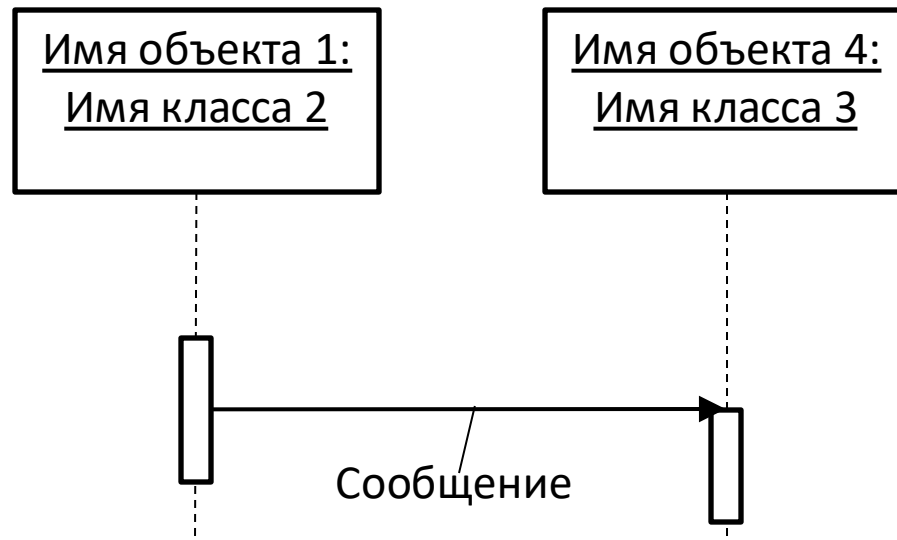
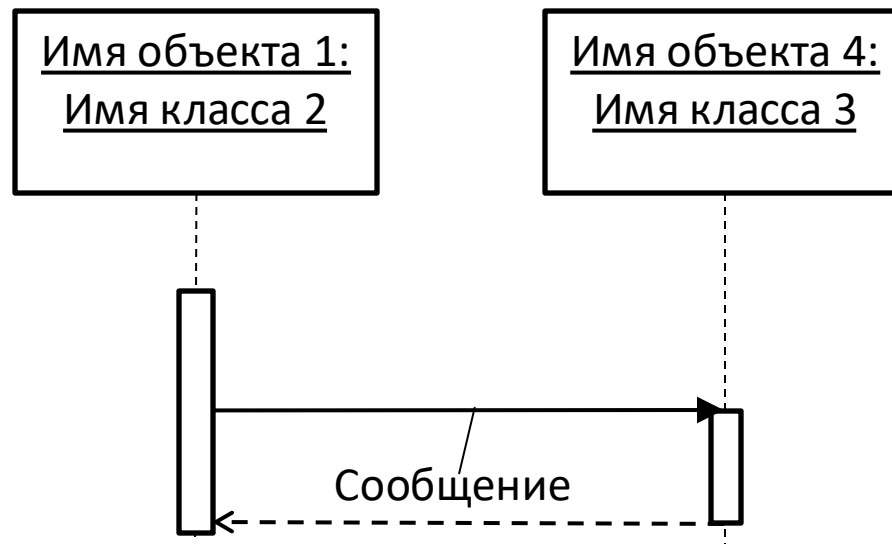


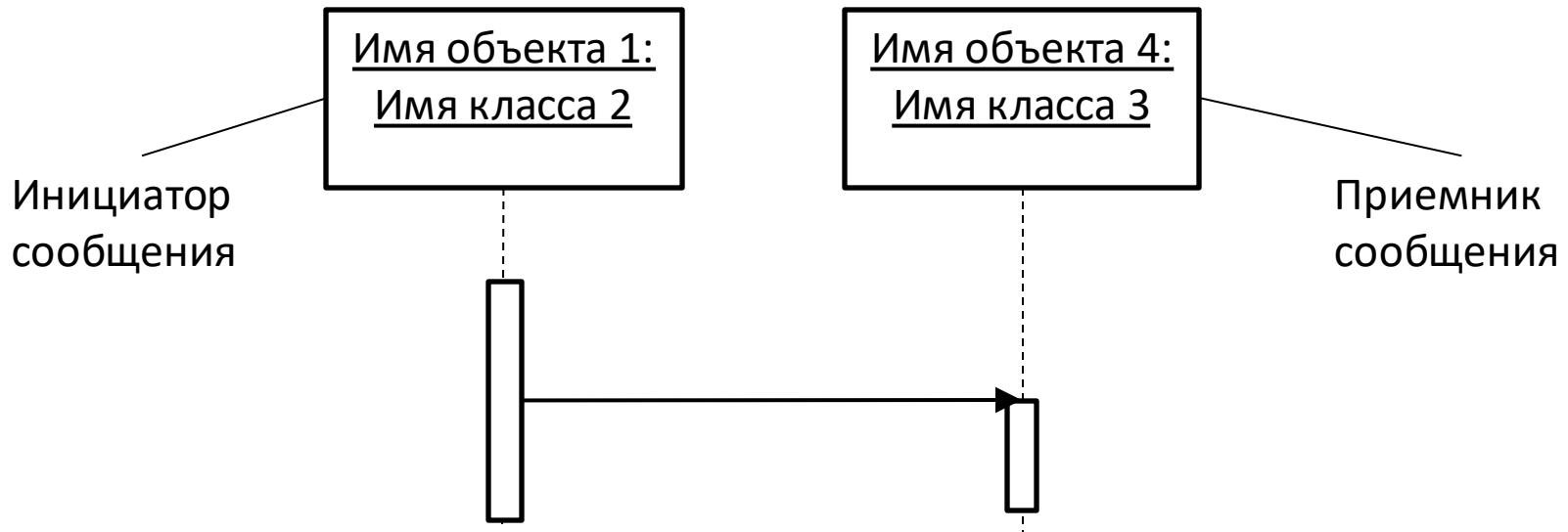
Диаграмма последовательности. Сообщения

- На диаграмме последовательности все сообщения упорядочены по времени своего возникновения.
- Каждое сообщение имеет направление от объекта, который иницирует и отправляет сообщение, к объекту, который его получает.



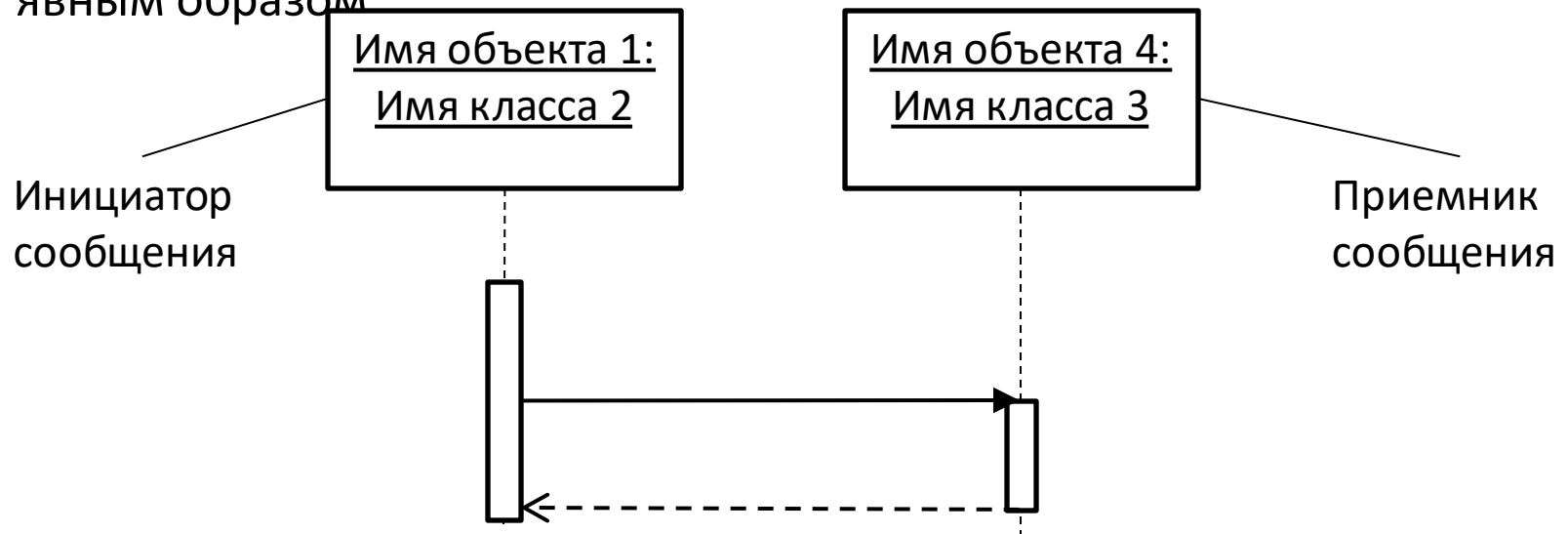
Виды сообщений

- Первая разновидность сообщения является наиболее распространенной и используется для вызова процедур, выполнения операций или обозначения отдельных потоков управления.
- Начало этой стрелки всегда соприкасается с фокусом управления или линией жизни того объекта, который инициирует это сообщение.
- Конец стрелки соприкасается с линией жизни того объекта, который принимает это сообщение и выполняет в ответ определенные действия. При этом принимающий объект зачастую получает и фокус управления, становясь активным.



Виды сообщений

- Вторая разновидность сообщения используется для возврата из вызова процедуры.
- Примером может служить простое сообщение о завершении некоторых вычислений без предоставления результата расчетов объекту.
- 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

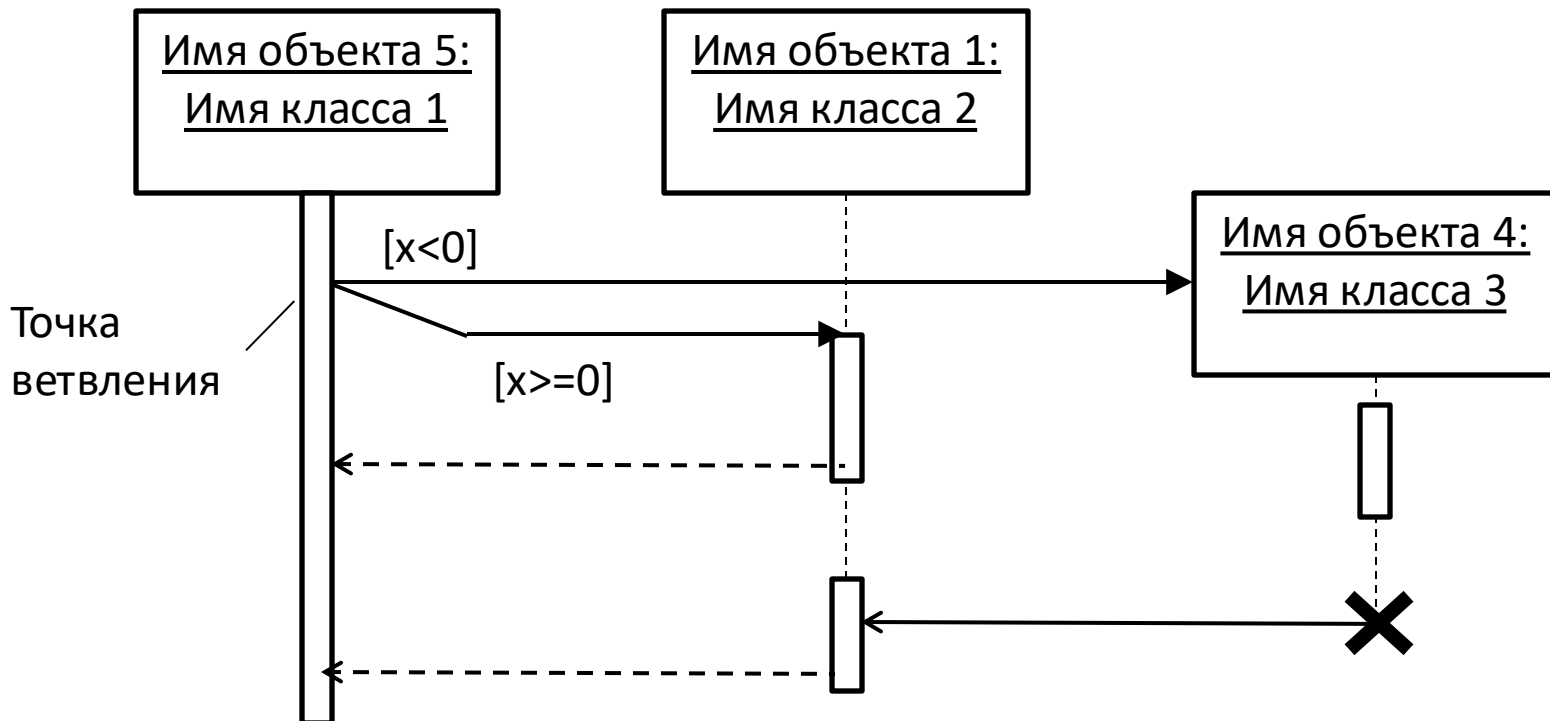


Виды сообщений

- 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 остаются без изменения.
- Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.
- В отдельных случаях объект может посылать сообщения самому себе, иницилируя так называемые рефлексивные сообщения. Такие сообщения изображаются прямоугольником со стрелкой, начало и конец которой совпадают. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Ветвление потока управления

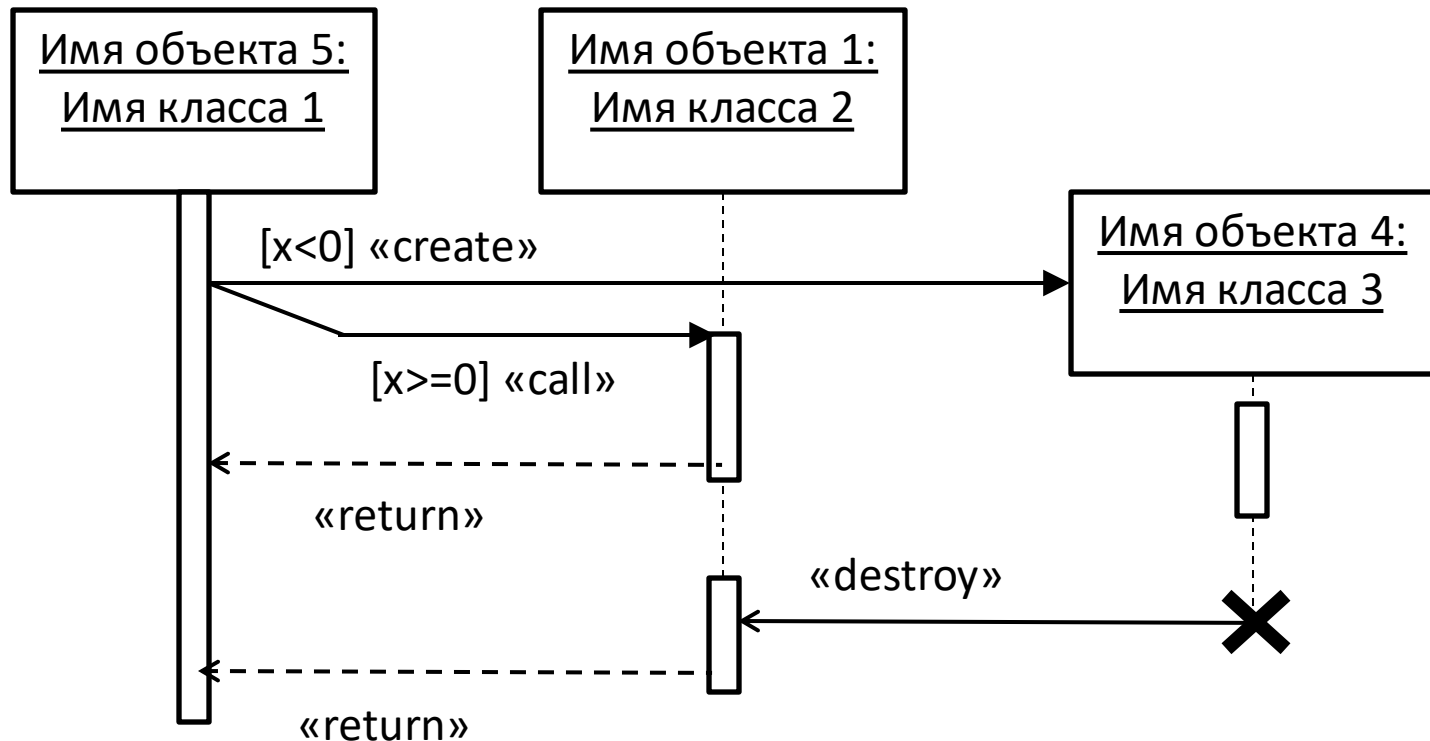
- Для изображения ветвления рисуются две или более стрелки, выходящие из одной точки фокуса управления объекта .
- При этом соответствующие условия должны быть явно указаны рядом с каждой из стрелок в форме сторожевого условия. Условия должны взаимно исключать одновременную передачу альтернативных сообщений.



Стереотипы сообщений

- **"call" (вызвать)** — сообщение, требующее вызова операции или процедуры принимающего объекта. Если сообщение с этим стереотипом рефлексивное, то оно инициирует локальный вызов операции у самого пославшего это сообщение объекта;
- **"return" (возвратить)** — сообщение, возвращающее значение выполненной операции или процедуры вызвавшему ее объекту. Значение результата может инициировать ветвление потока управления;
- **"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. Посылается в том случае, когда необходимо прекратить нежелательные действия со стороны существующего в системе объекта, либо когда объект больше не нужен и должен освободить задействованные им системные ресурсы;
- **"send" (послать)** — обозначает посылку другому объекту некоторого сигнала, который асинхронно инициируется одним объектом и принимается (перехватывается) другим. Отличие сигнала от сообщения заключается в том, что сигнал должен быть явно описан в том классе, объект которого инициирует его передачу.

Диаграмма последовательности со стереотипами сообщений



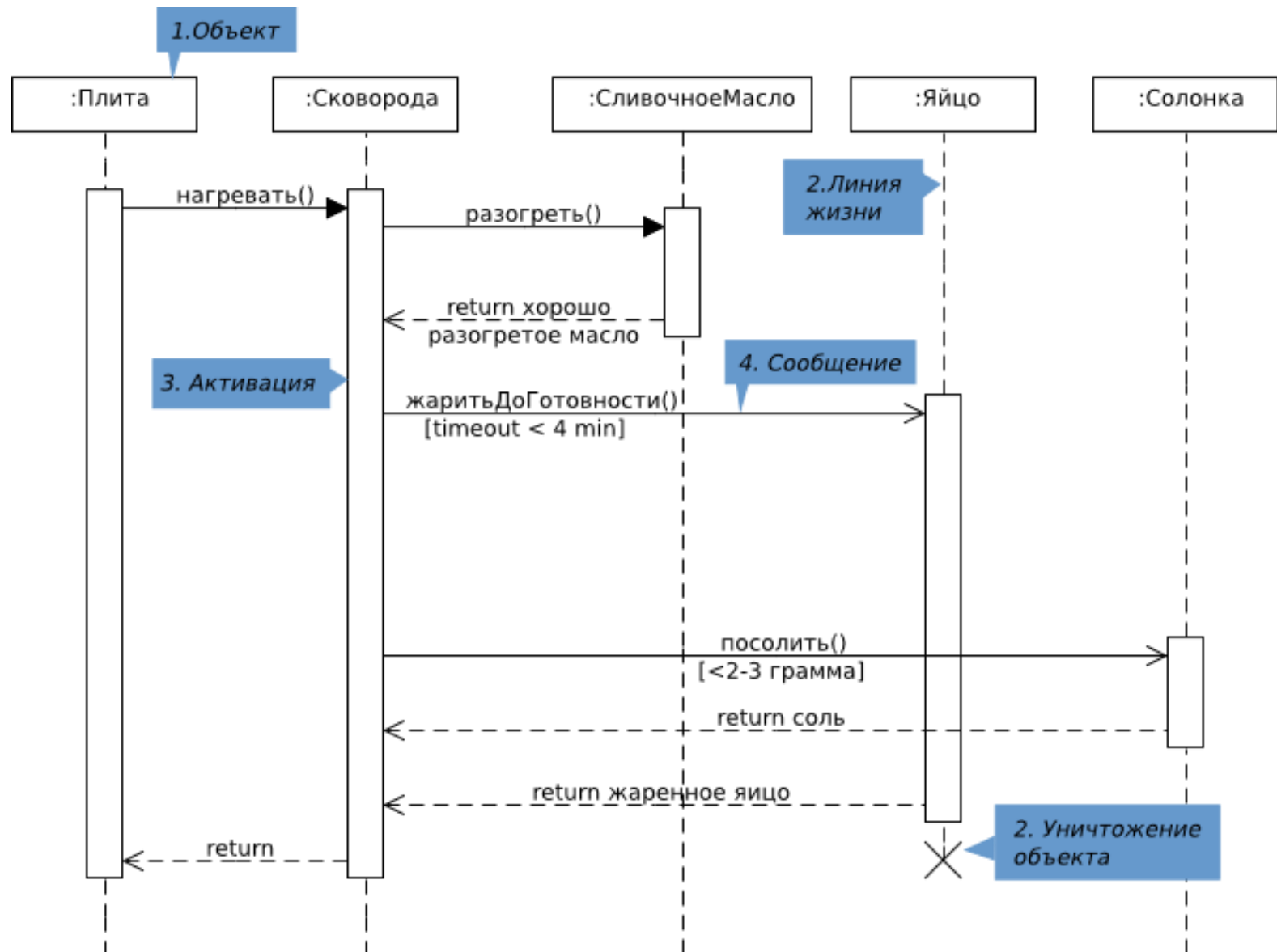
Операции в сообщениях

- Сообщения могут иметь собственное обозначение операции, вызов которой они инициируют 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 все равно должны присутствовать после имени операции.
- Примерами таких операций могут служить следующие: "выдать клиенту наличными сумму (n)", "установить соединение между абонентами (a, b)", "сделать вводимый текст невидимым ()", "подать звуковой сигнал тревоги ()".
- **Примечание**
- Согласно принятой в языке UML системе обозначений такие имена операций записываются на английском с малой буквы и одним словом, возможно, состоящим из нескольких сокращенных слов, написанных без пробела и без кавычек.
- Дело вкуса отечественного разработчика, какие обозначения ему использовать в русскоязычной транслитерации. Возможно, для этой цели больше подходит вариант с нижней черточкой, исключающей пробелы в имени операции: "сделать_вводимый_текст_невидимым ()", чем вариант с заглавными буквами в середине имени операции: "сделатьВводимыйТекстНевидимым()".

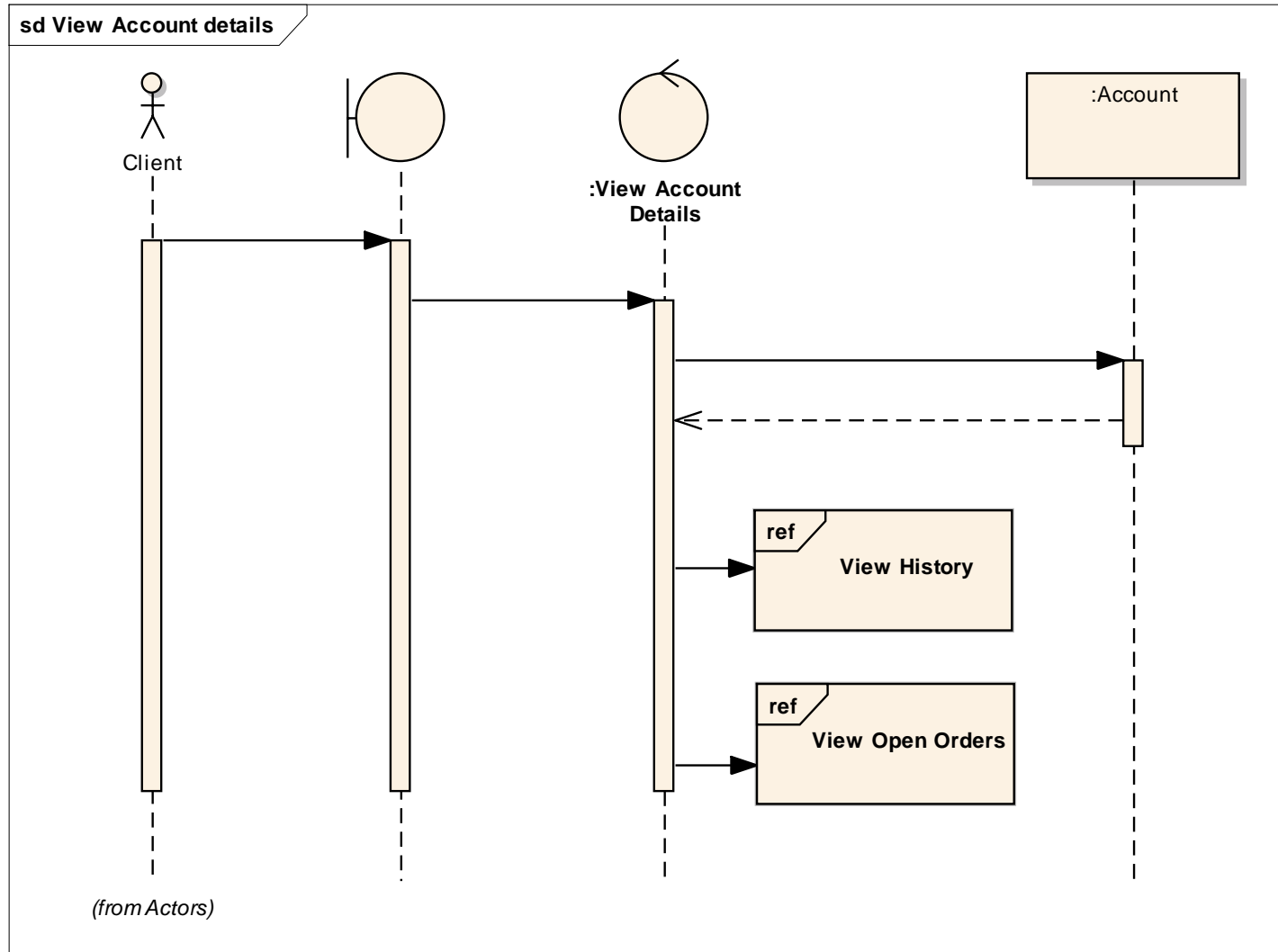
Временные ограничения

- 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.
- Временные ограничения могут относиться как к выполнению определенных действий объектами, так и к самим сообщениям.
- Важно понимать, что в отличие от условий ветвления, которые должны выполняться альтернативно, временные ограничения имеют обязательный или директивный характер для ассоциированных с ними объектов.
- {время_ожидания_ответа < 5 сек.}
- {время_передачи_пакета < 10 сек.}

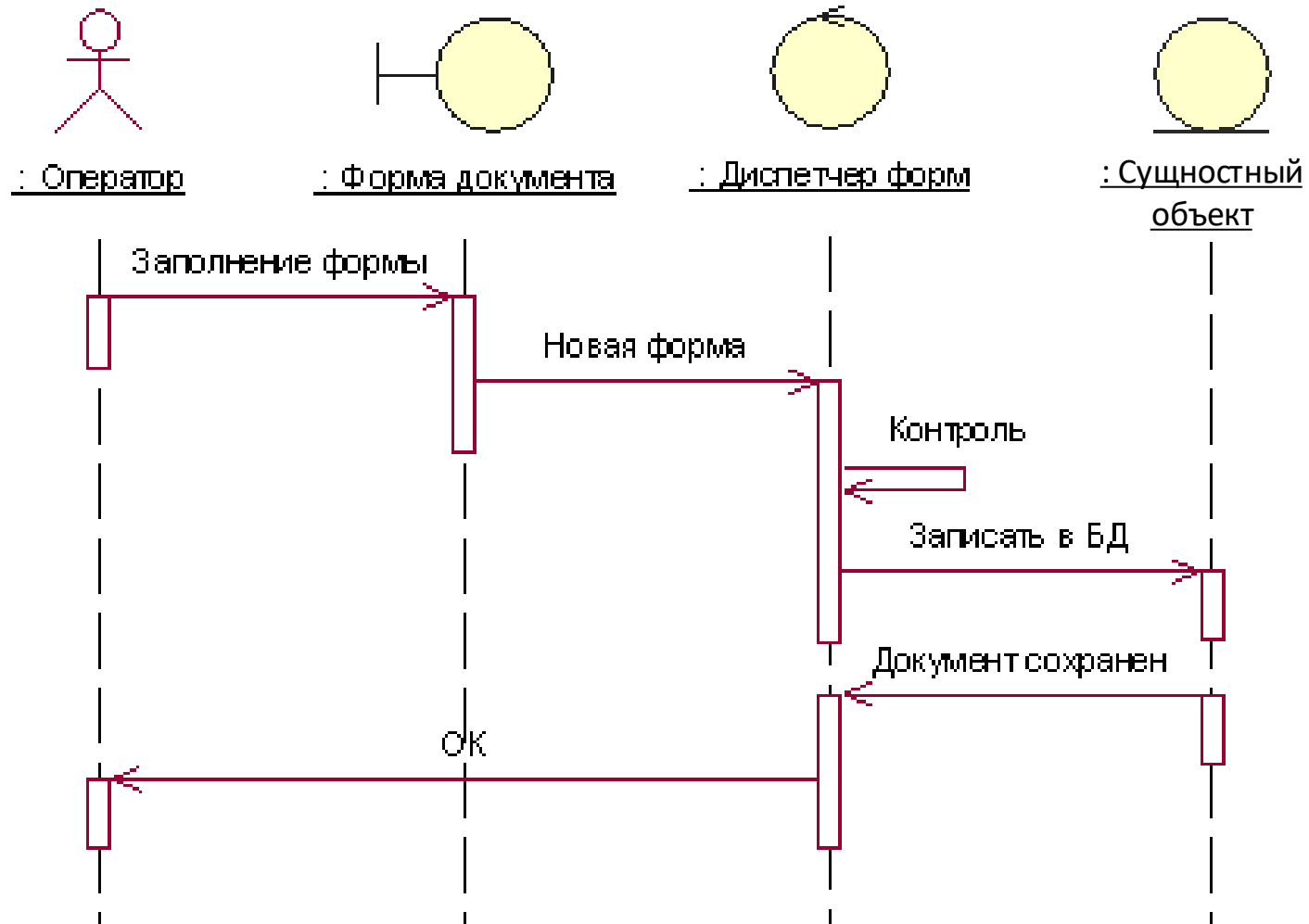
Пример



Пример



Пример



Виды диаграмм последовательностей

- Диаграммы бывают *концептуальные (системные)* и *детализированные*.
- *Концептуальные диаграммы последовательностей отображают взаимодействия во времени внешних сущностей и системы. Строятся на этапе анализа. При этом система рассматривается как единое целое (без детализации на подсистемы) – по своей сути являются контекстными диаграммами.*
- Для построения диаграмм используются обозначения, используемые при построении диаграмм классов концептуального уровня и уровня проектирования.
- Система – отображается в виде абстрактного класса ***mupа System***.
- Линия жизни – явное отображение времени существования объектов.
- Системные события – это события, которые генерируются для системы действующими лицами (внешними сущностями). Системные события инициируют выполнение соответствующего множества операций, также называемых системными. Множество всех системных операций определяют, идентифицируя системные события всех вариантов использования. Для наглядности системные операции изображают в виде операций ***абстрактного класса (mupа) System***. Если необходимо разделить множество операций на подмножества, инициируемые разными пользователями, то используют ***несколько абстрактных классов***: System 1, System 2 и т. д.

Рекомендации

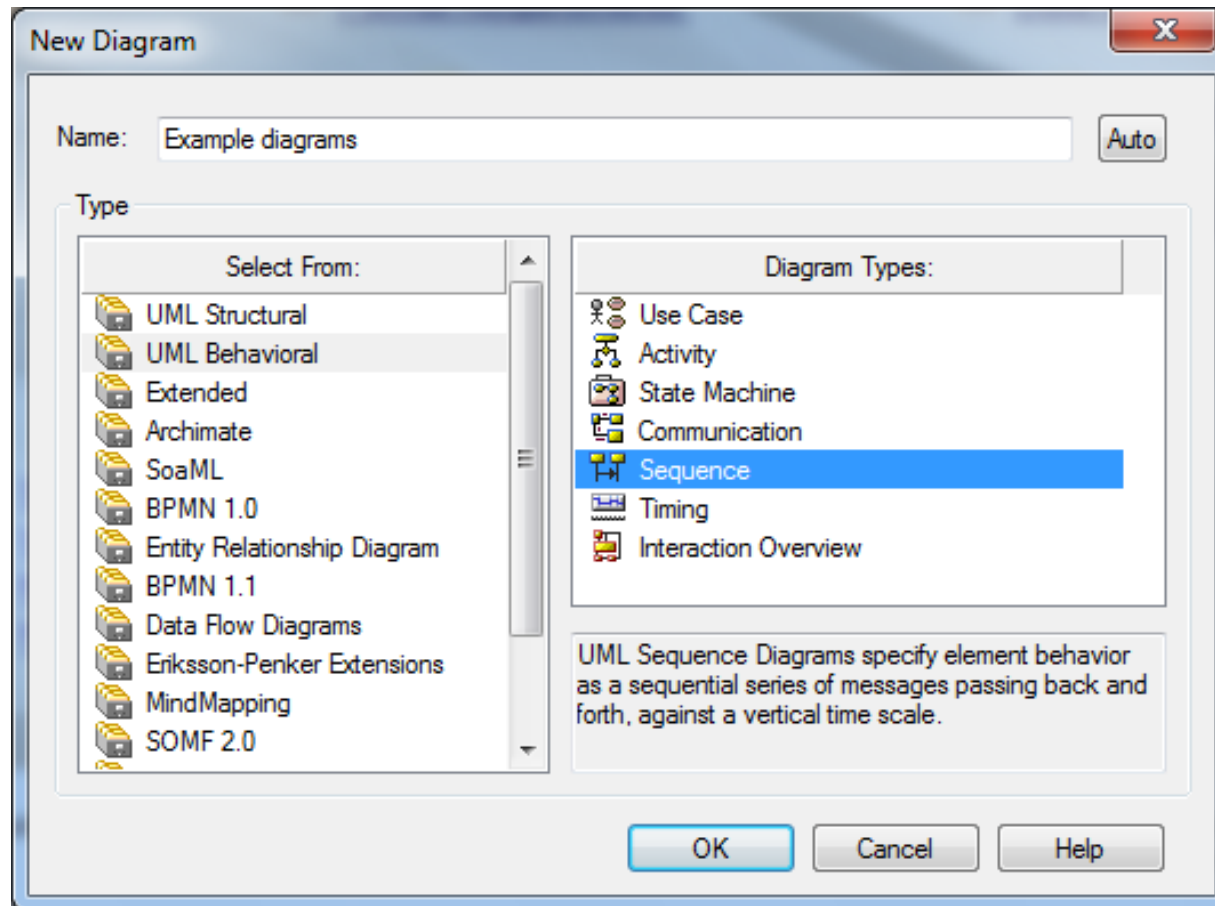
- Как уже отмечалось, построение диаграммы последовательности целесообразно начинать с выделения из всей совокупности тех и только тех классов, объекты которых участвуют в моделируемом взаимодействии.
- После этого все объекты наносятся на диаграмму с соблюдением некоторого порядка инициализации сообщений. Здесь необходимо установить, какие объекты будут существовать постоянно, а какие временно — только на период выполнения ими требуемых действий.
- Когда объекты визуализированы, можно приступить к спецификации сообщений.
- При этом следует учитывать те роли, которые играют сообщения в системе. При необходимости уточнения этих ролей надо использовать их разновидности и стереотипы. Для уничтожения объектов, которые создаются на время выполнения своих действий, нужно предусмотреть явное сообщение.

Рекомендации

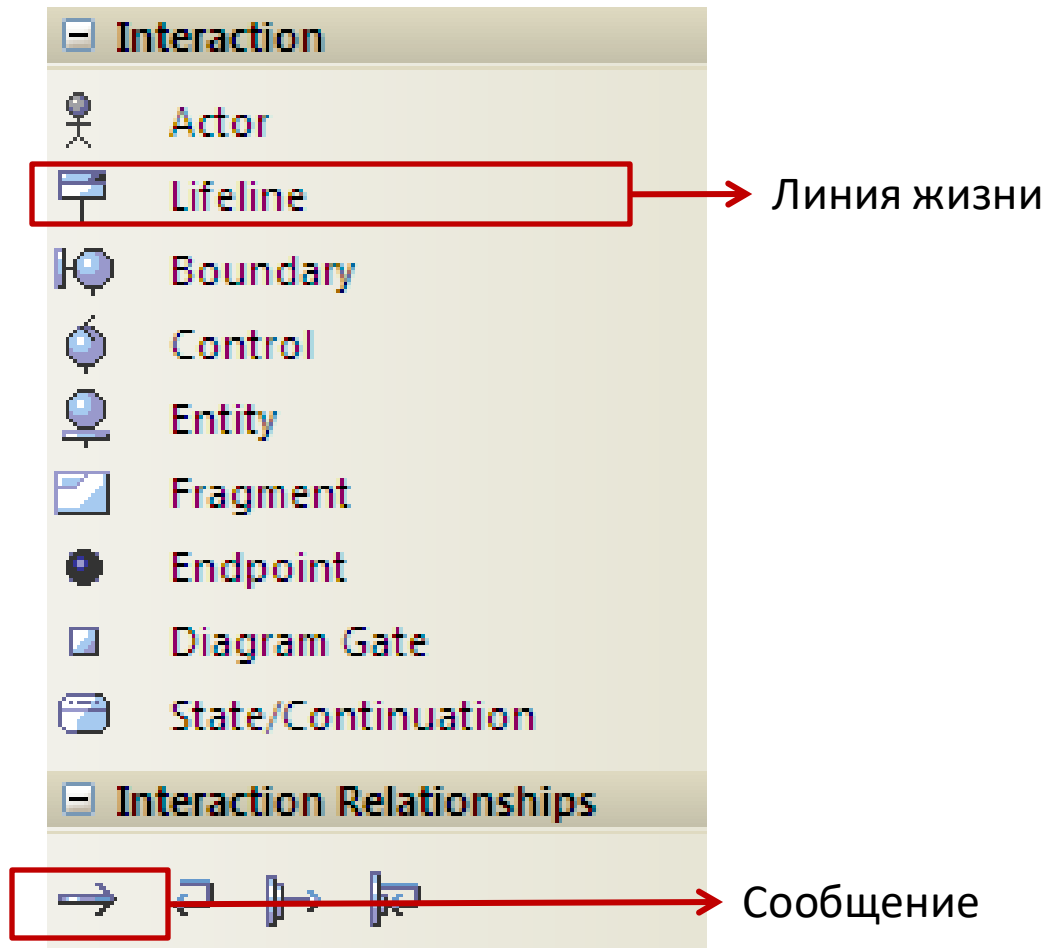
- Наиболее простые случаи ветвления процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. Однако следует помнить, что каждый альтернативный поток управления может существенно затруднить понимание построенной модели.
- Поэтому общим правилом является визуализация каждого потока управления на отдельной диаграмме последовательности. В этой ситуации такие отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия.
- Дальнейшая детализация диаграммы последовательности связана с введением временных ограничений на выполнение отдельных действий в системе. Для простых асинхронных сообщений временные ограничения могут отсутствовать. Однако необходимость синхронизировать сложные потоки управления, как правило, требуют введение в модель таких ограничений. Общая их запись должна следовать семантике языка объектных ограничений, который рассмотрен в приложении.

Создание диаграммы последовательности

- New Diagram-> UML Behavioral-> Sequence

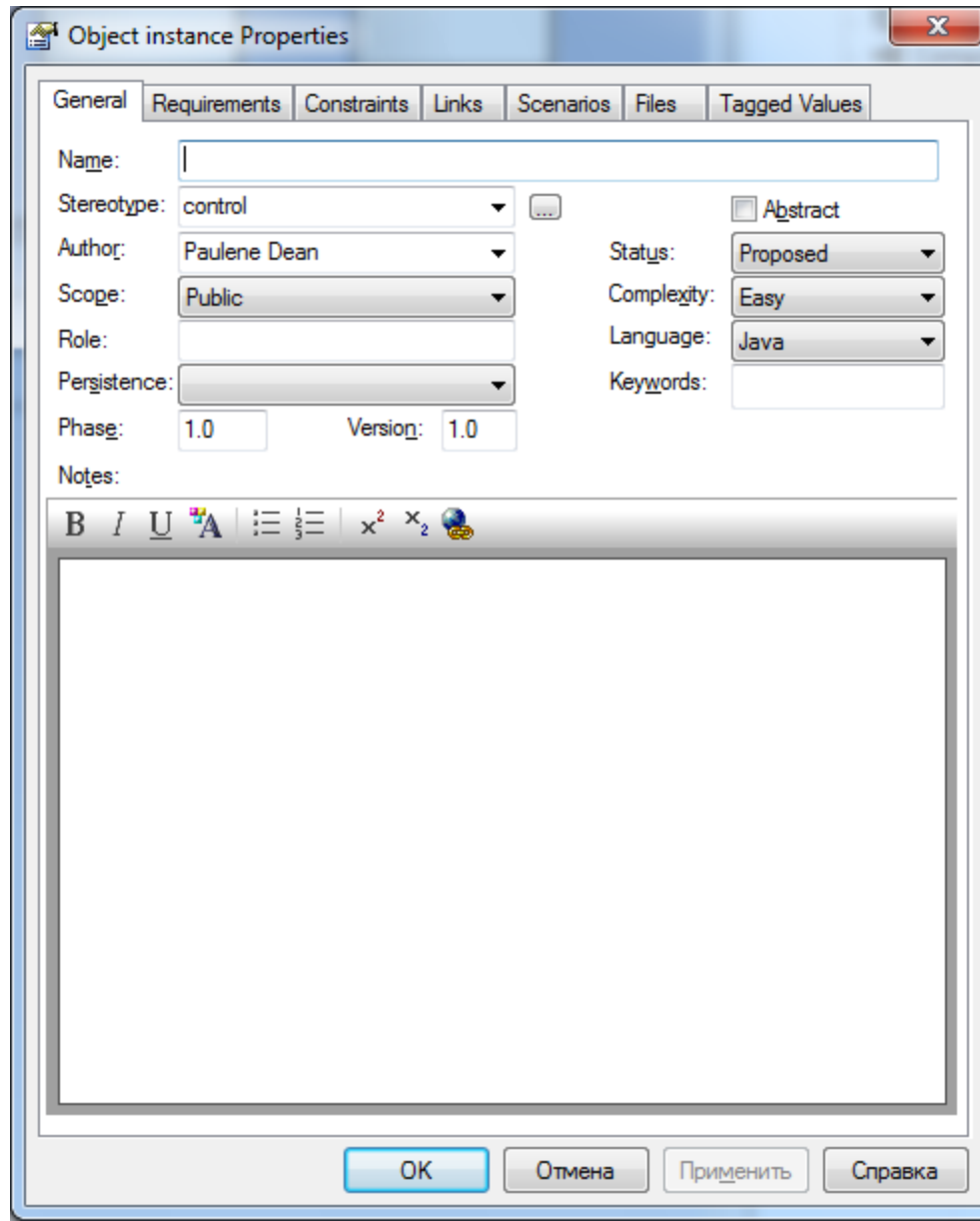


Создание диаграммы последовательности



- Для создания диаграммы используется инструмент «Interaction».

Спецификация объекта



The image shows a software dialog box titled "Object instance Properties". It features a tabbed interface with the following tabs: General, Requirements, Constraints, Links, Scenarios, Files, and Tagged Values. The "General" tab is currently selected. The dialog contains several input fields and dropdown menus for specifying object properties. At the bottom, there is a "Notes" section with a rich text editor toolbar and a large text area. The bottom of the dialog has four buttons: "OK", "Отмена" (Cancel), "Применить" (Apply), and "Справка" (Help).

Object instance Properties

General Requirements Constraints Links Scenarios Files Tagged Values

Name:

Stereotype: control ☐ Abstract

Author: Paulene Dean Status: Proposed

Scope: Public Complexity: Easy

Role: Language: Java

Persistence: Keywords:

Phase: 1.0 Version: 1.0

Notes:

B I U A |

OK Отмена Применить Справка

Спецификация связи Message

Message Properties

Signature

Message: createNewAccount() Operations

Parameters

Argument(s):

Return Value: void ☒ Show Inherited Methods

Assign To:

Stereotype:

Alias:

Sequence Expression

Condition:

Constraint:

☐ Is Iteration

Control Flow Type:

Synch: Synchronous Lifecycle:

Kind: Call ☐ Is Return

Notes:

OK Cancel Help

Имя сообщения

Условия и
ограничения