ЛЕКЦИЯ 8-10

Тема: Диаграмма последовательностей. Основные части диаграммы. Принцип работы и основные параметры.

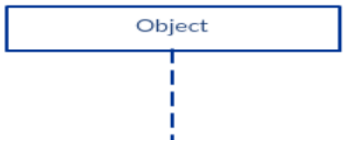
Диаграммы последовательностей, обычно используемые разработчиками, моделируют взаимодействия между объектами в едином сценарии использования. Они иллюстрируют, как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая использования.

Проще говоря, диаграмма последовательности показывает различные части работы системы в “последовательности”, чтобы что-то сделать.

Обозначения диаграмм последовательности

Принцип построения: Схема последовательности построена таким образом, что она представляет собой временную шкалу, которая начинается сверху и постепенно опускается, чтобы отметить последовательность взаимодействий. Каждый объект имеет колонку, а сообщения, которыми обмениваются между собой, представлены стрелками.

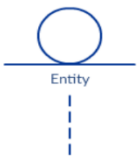
Части диаграммы последовательности



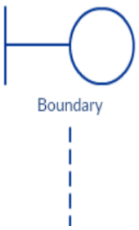
1.Нотация линии жизни обьект последовательность состоит из нескольких таких обозначений линии жизнеобеспечения, которые должны быть расположены горизонтально в верхней части диаграммы.

Никакие две нотации страховочной линии не должны перекрывать друг друга. Они представляют собой различные объекты или части, которые взаимодействуют друг с другом в системе во время последовательности.

2. линия жизни Actor – экземпляр участника процесса (роль на диаграмме прецедентов)



2.линия жизни Entity – Класс-сущность - обычно применяется для обозначения классов, которые хранят некую информацию о бизнес-объектах (соответствует таблице или элементу БД)



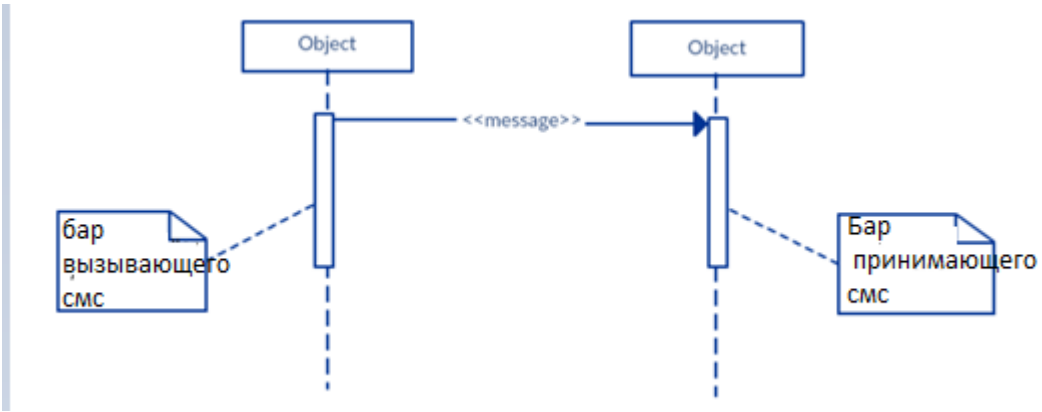
3. линия жизни Boundary – Класс-Разграничитель - используется для классов, отделяющих внутреннюю структуру системы от внешней среды (экранная форма, пользовательский интерфейс, устройство ввода-вывода). Объект большей предназначен для вызова методов класса, с которым он связан. Объект boundary показывает именно экранную форму, которая принимает и передает данные обработчику

4. линия жизни Control – Класс-контроллер - активный элемент, который используются для выполнения некоторых операций над объектами (программный компонент, модуль, обработчик)

5. Активация – прямоугольник на линии жизни, который показывающий на период времени , в течении которого элемент выполняет операцию

взаимодействия между двумя объектами. Длина прямоугольника указывает на продолжительность пребывания объектов в активном состоянии.

На диаграмме последовательности взаимодействия между двумя объектами происходит, когда один объект посылает сообщение другому.



Использование строки активации на спасательных линиях вызывающего сообщения (объекта, отправляющего сообщение) и получателя сообщения (объекта, принимающего сообщение) указывает на то, что оба они активны/осуществляются во время обмена сообщением.

8.Стрелки -*используются для вызова процедур выполнения операций или обозначают отдельно вложенных потоков управления. Начало стрелки всегда соприкасается с линией жизни или блоком активации.*

Стрелка от одного к другому приемнику указывает сообщение на схеме последовательности. Сообщение может идти в любом направлении: слева направо, справа налево или обратно к самому вызывающему сообщение абоненту. В то время как вы можете описать сообщение, отправляемое с одного объекта на другой, на стрелке, с разными заголовками стрелок, вы можете указать тип отправляемого или получаемого сообщения.

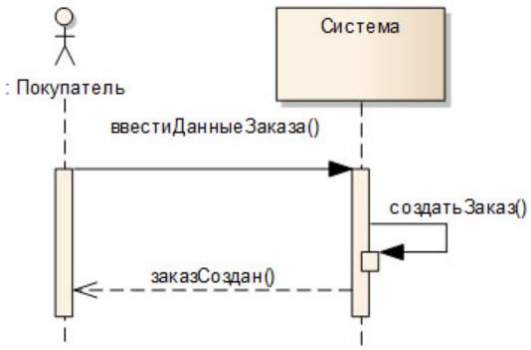
Стрелка сообщения содержит описание, известное как подпись сообщения.

Атрибут = имя_сообщения (аргументы): return_type

- **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

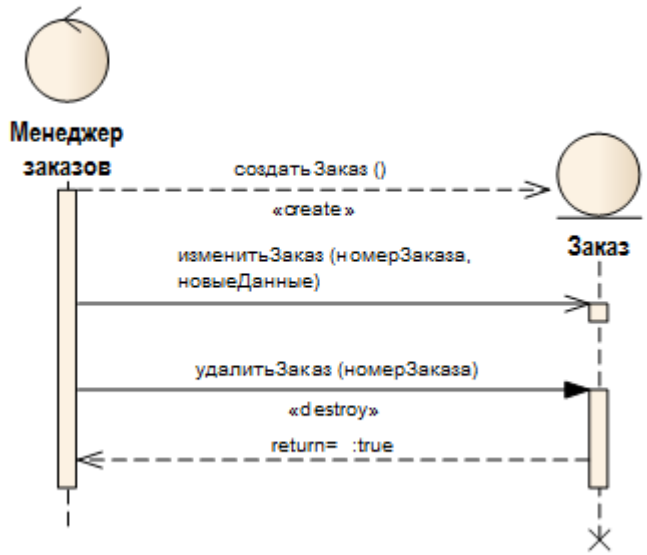


На диаграмме состояний частично показан обмен сообщениями в рамках сообщений инициирующих изменение состояния объекта., если есть действие стрелка выбирается с темным наконечником

Диаграмма последовательности объединяет диаграмму деятельности, диаграмму состояний и диаграмму классов.

диаграмме последовательности мы можем увидеть следующие аспекты:

- **Сообщения**, побуждающие объект к действию
- **Действия**, которые вызываются сообщениями (методы) – зачастую это передача сообщения следующему объекты или возвращение определенных данных объекта
- **Последовательность** обмена сообщениями между объектами
- **Рекурсивный вызов-** действие отправляющий запрос само себе, в данном случае она говорит себе создать заказ



Если нам нужно создать участника(в данном случае новый заказ), мы ведем стрелку к этому элементу, если он не был создан.

После этого всегда должен быть блок активации на этой линии жизни, в данном случае это активатор для изменения заказа, так как он был создан пустым.

Сообщение в большинстве случаев (за исключением диаграмм, описывающих концептуальный уровень Системы) это вызов методов отдельных объектов, поэтому для корректного исполнения метода в сообщении необходимо передать какие-то данные и определить, что мы хотим видеть в ответ. При именовании сообщения на уровне проектирования реализации системы в качестве имени сообщения следует использовать имя метода.

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

НАШ ОПЕРАТОР if-then-else; case; switch

- **Opt -** Необязательный (optional) фрагмент; выполняется, только если условие истинно. Эквивалентно **alt** с одной веткой

if-then, без ветки иначе

- **Par -** Параллельный (parallel); все фрагменты выполняются параллельно

Это наши разграничители на схеме отделает наш if от else

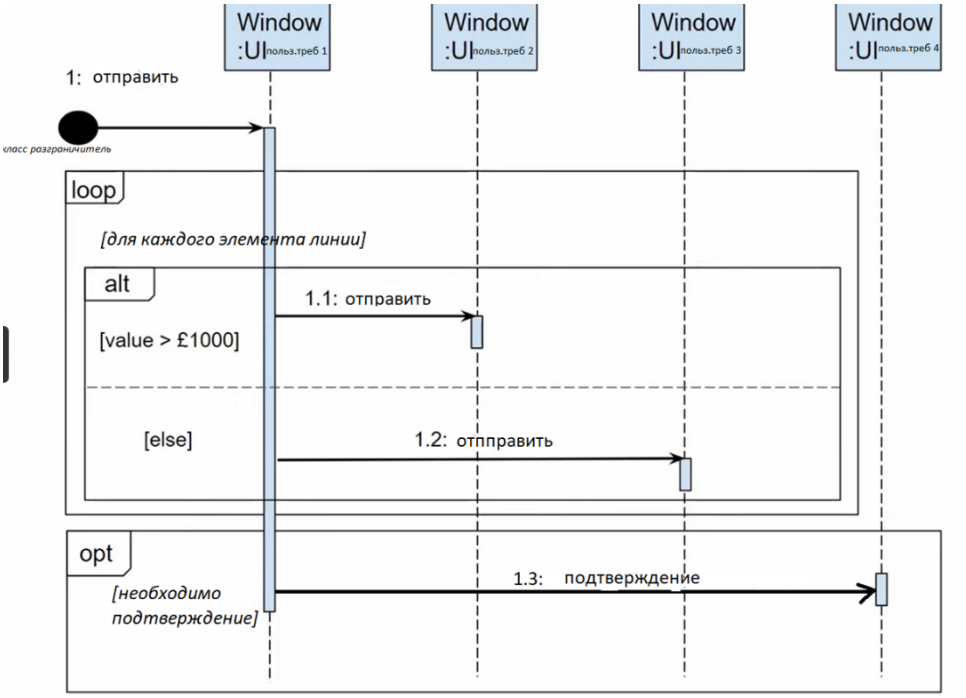
- **loop -** Цикл (loop); фрагмент может выполняться несколько раз, а защита обозначает тело итерации
- **Neg -** Отрицательный (negative) фрагмент; обозначает неверное взаимодействие

К примеру, блок ждет пароля от пользователя, время ожидания вышло и ему выведется сообщение Время вышло

- **ref -** Ссылка (reference); ссылается на взаимодействие, определенное на другой диаграмме. Фрейм рисуется, чтобы охватить линии жизни, вовлеченные во взаимодействие. Можно определять параметры и возвращать значение

ссылка на другой блок или диаграмму

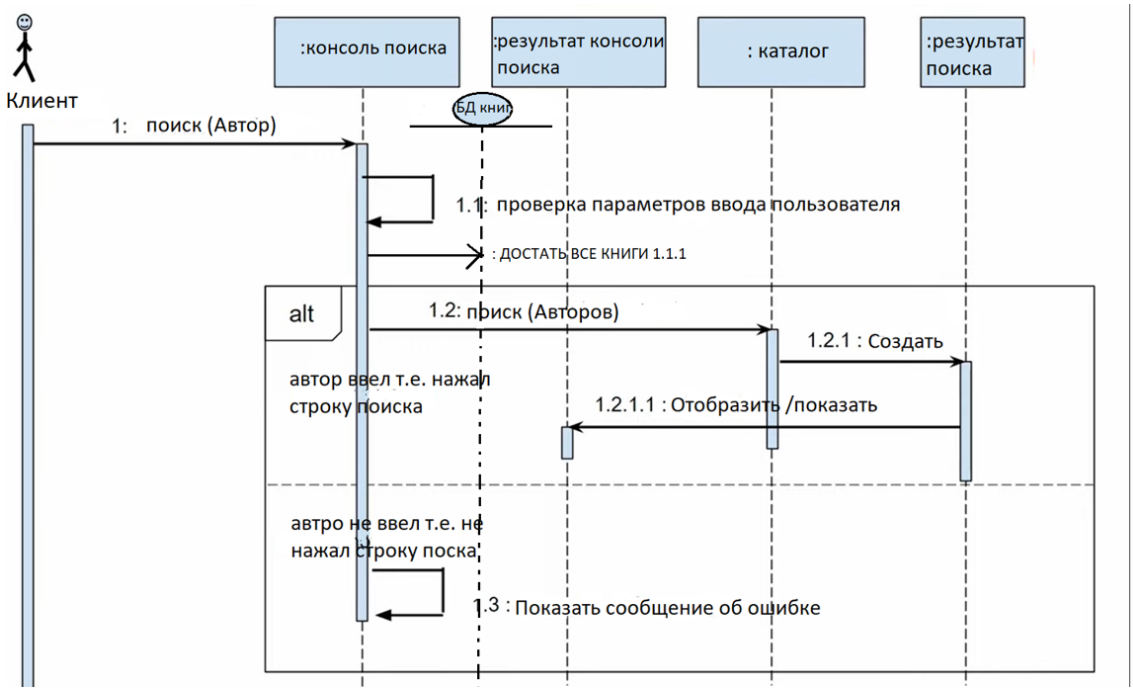
- **Sd -** (sequence diagram); используется для очерчивания всей диаграммы последовательности, если это необходимо.



Как показать цикличность

В данном случае используется класс разграничитель(черный кружек)и 4 линии жизни (пользовательские инерфейсы 4 шт). Прямоукольник loop выделил область которая указывет на цикл, а прямоуголник с alt указывает на условие(пока пользователь не достиг просмотра 1000минут -он видит сообщение 1.1, если время становится больше 1000-он увидит сообщение 1.2. Далее Блок блок ALT,он показывет истенное выражение- в данном случае пользователю нравится интерфейс и он нажимает на лайк-это и есть подтверждение.

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.



есть два сценария:

1. Основной
2. Альтернативный

1.Основной: КЛИЕНТ вводит ФИО автора и нажимает кнопку поиска. После чего система проверяет параметры ввода пользователя, если валидация пройдена-система ищет Автора по каталогам, когда поиск окончен -система выдает результат

2. Если КЛИЕНТ не ввел имя автора, но нажал на кнопку поиска, то система выводит сообщение об ошибке.