

Лабораторная работа № 6

Построение диаграммы последовательности (sequence diagram)

Цель: изобразить участвующие во взаимодействии объекты и последовательность сообщений, которыми они обмениваются.

Задание 1. Разработать диаграммы последовательности для вашего проекта (для прецедентов описанных в диаграммах деятельности (activity)).

Объекты диаграммы последовательности

Диаграммы последовательности отражают поток событий, происходящих в рамках варианта использования. На этих диаграммах изображаются только те объекты, которые непосредственно участвуют во взаимодействии т.к. ключевым моментом является именно динамика взаимодействия объектов во времени и не используются возможные статические ассоциации с другими объектами. При этом диаграмма последовательности имеет два измерения (рис. 1). Одно – слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Второе измерение – вертикальная временная ось, направленная сверху вниз. При этом взаимодействия объектов реализуются посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения. Другими словами, сообщения, расположенные на диаграмме последовательности выше, инициируются раньше тех, которые расположены ниже.

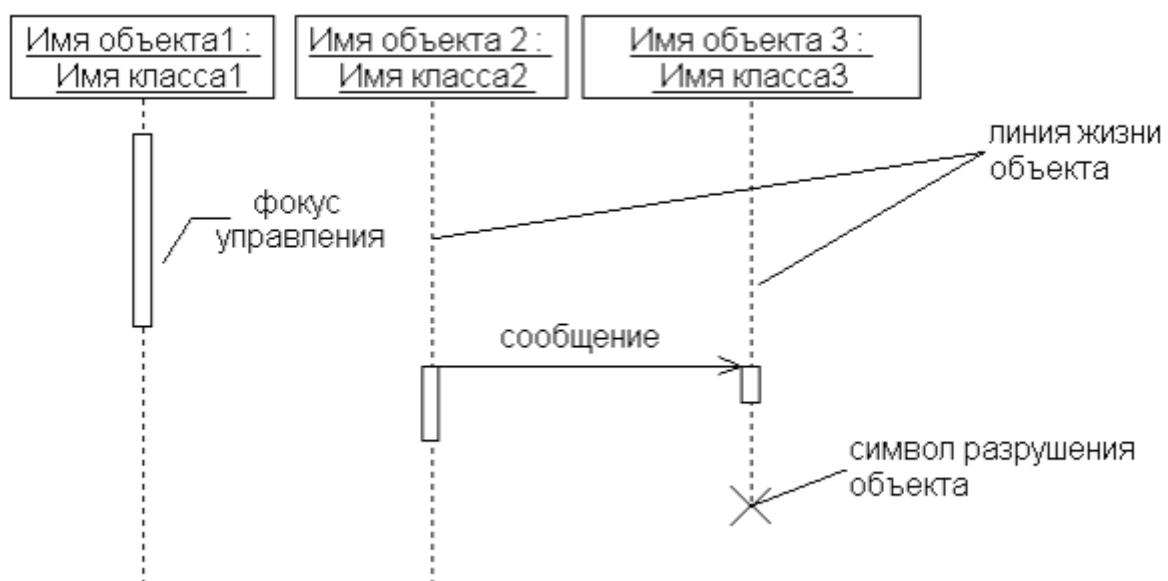


Рис. 1 Графические примитивы диаграммы последовательности

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 и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то его линия жизни должна начинаться в верхней части диаграммы и заканчиваться в нижней части (объекты 1 и 2 на рис. 1). Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены, чтобы освободить занимаемые ими ресурсы. Для обозначения момента уничтожения объекта в языке UML используется специальный символ в форме латинской буквы “X” (объект 3 на рис. 1). Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий.

Отдельные объекты в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее производительность (объект 6 на рис. 2).

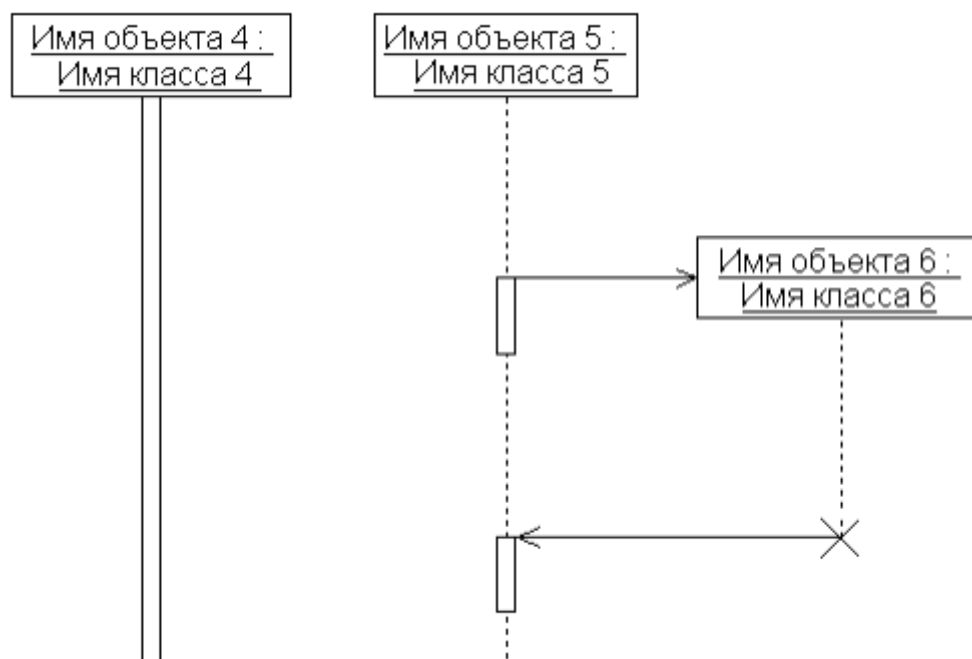


Рис. 2 Варианты линий жизни и фокусов управления объектов

Комментарии или примечания уже рассматривались ранее при изучении других видов диаграмм. Они могут включаться и в диаграммы последовательности, ассоциируясь с отдельными объектами или сообщениями.

Как уже отмечалось выше, взаимодействия объектов реализуются с помощью сообщений. У каждого сообщения должно быть имя, соответствующее его цели. Существует несколько видов сообщений: простое, синхронное, с отказом становиться в очередь и др. (рис. 3).



Рис. 3 Примеры сообщений

Простое сообщение используется по умолчанию. Означает, что все сообщения выполняются в одном потоке управления (рис. 3, 1).

Синхронное (synchronous) применяется, когда клиент посылает сообщение и ждет ответа пользователя (рис. 3, 2).

Сообщение с отказом становиться в очередь (balking): клиент посылает сообщение серверу и, если сервер не может немедленно принять сообщение, оно отменяется (рис. 3, 3).

Сообщение с лимитированным временем ожидания (timeout): клиент посылает сообщение серверу, а затем ждет указанное время; если в течение этого времени сервер не принимает сообщение, оно отменяется (рис. 3, 4).

Асинхронное сообщение (asynchronous): клиент посылает сообщение серверу и продолжает свою работу, не ожидая подтверждения о получении (рис. 3, 5).

Пример диаграммы последовательности

Пример сценария снятия 20\$ со счета (при отсутствии таких проблем, как неправильный идентификационный номер или недостаток денег на счету) показан на рис. 4.

Эта диаграмма последовательности отображает поток событий в рамках варианта использования «Снять деньги». В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Снять деньги». Стрелки соответствуют сообщениям, передаваемым между действующим лицом и объектом или между объектами для выполнения требуемых функций. Следует отметить также, что на диаграмме Последовательности показаны именно объекты, а не классы. Классы представляют собой типы объектов. Объекты конкретны; вместо класса Клиент на диаграмме

Последовательности представлен конкретный клиент Джо.

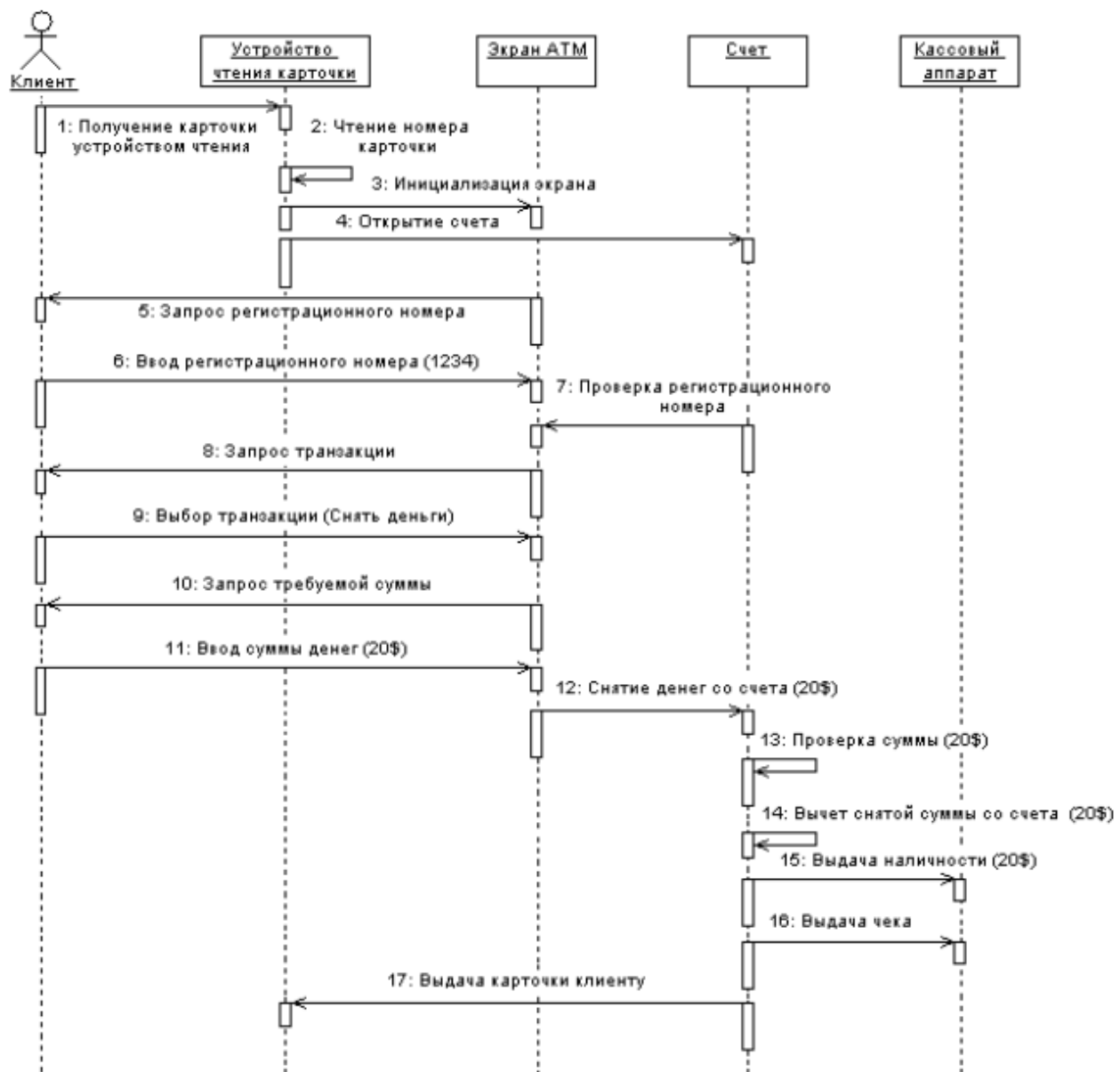


Рис. 4 Диаграмма последовательности для снятия клиентом 20\$

Вариант использования начинается, когда клиент вставляет свою карточку в устройство для чтения – этот объект показан в прямоугольнике в верхней части диаграммы. Он считывает номер карточки, открывает объект «счет» (account) и инициализирует экран АТМ. Экран запрашивает у клиента его регистрационный номер. Клиент вводит число 1234. Экран проверяет номер у объекта «счет» и обнаруживает, что он правильный. Затем экран предоставляет клиенту меню для выбора, и тот выбирает пункт «Снять деньги». Экран запрашивает, сколько он хочет снять, и клиент указывает 20\$. Экран снимает деньги со счета. При этом он инициирует серию процессов, выполняемых объектом «счет». Во-первых, осуществляется проверка, что на этом счету лежат, по

крайней мере, 20\$. Во-вторых, из счета вычитается требуемая сумма. Затем кассовый аппарат получает инструкцию выдать чек и \$20 наличными. Наконец все тот же объект «счет» дает устройству для чтения карточек инструкцию вернуть карточку.

Таким образом, диаграмма последовательности иллюстрирует последовательность действий, реализующих вариант использования «Снять деньги со счета» на примере снятия клиентом 20\$. Глядя на эту диаграмму, пользователи знакомятся со спецификой своей работы. Аналитики видят последовательность (поток) действий, разработчики – объекты, которые надо создать, и их операции. Специалисты по контролю качества поймут детали процесса и смогут разработать тесты для их проверки. Таким образом, диаграммы последовательности полезны всем участникам проекта.