

Лабораторная работа №6

Тема: Построение диаграммы последовательности.

Цель: получить навыки построения диаграммы последовательности.

Программное обеспечение: MS Windows, MS Visio, Rational Rose, StarUML.

Оборудование: IBM PC.

Содержание работы

Задание.....	3
Теоретические сведения	4
Методические рекомендации по выполнению работы	7
Варианты заданий.....	9
Контрольные вопросы.....	12

Задание

1. Ознакомиться с теоретическими сведениями по теме лабораторной работы.
2. Ознакомиться с методическими рекомендациями и выполнить **задание в соответствии со своим вариантом.**

Номер варианта задания соответствует номеру по списку в журнале.

3. Построить диаграмму последовательности для одного варианта использования из вашей предыдущей лабораторной работы.
4. Для этого же варианта использования построить диаграмму коммуникации/кооперации.
5. Ответить на контрольные вопросы. Продемонстрировать работу преподавателю.
6. Написать отчет. В отчете должно быть размещено:
 - Титульный лист.
 - Содержание.
 - Тема, цель, ПО, Оборудование к лабораторной работе.
 - Построенные диаграммы последовательности и коммуникации.
 - Ответы на контрольные вопросы.
 - Вывод по результатам проделанной работы.

Теоретические сведения

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.

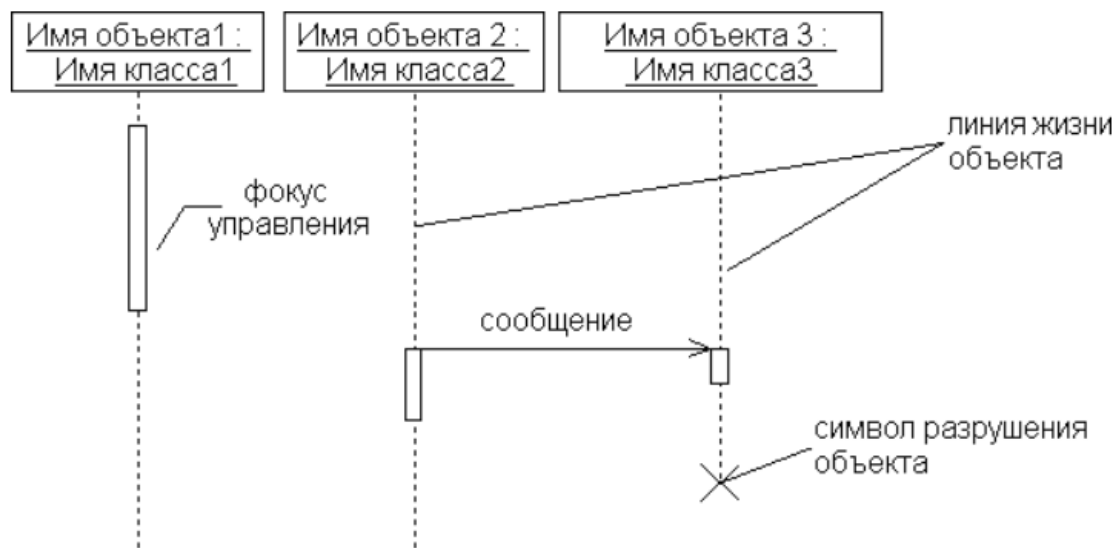


Рисунок 1. Пример типичной *Диаграммы последовательностей*

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. 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”. Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий.

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. 2.

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. 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 иницирует 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.

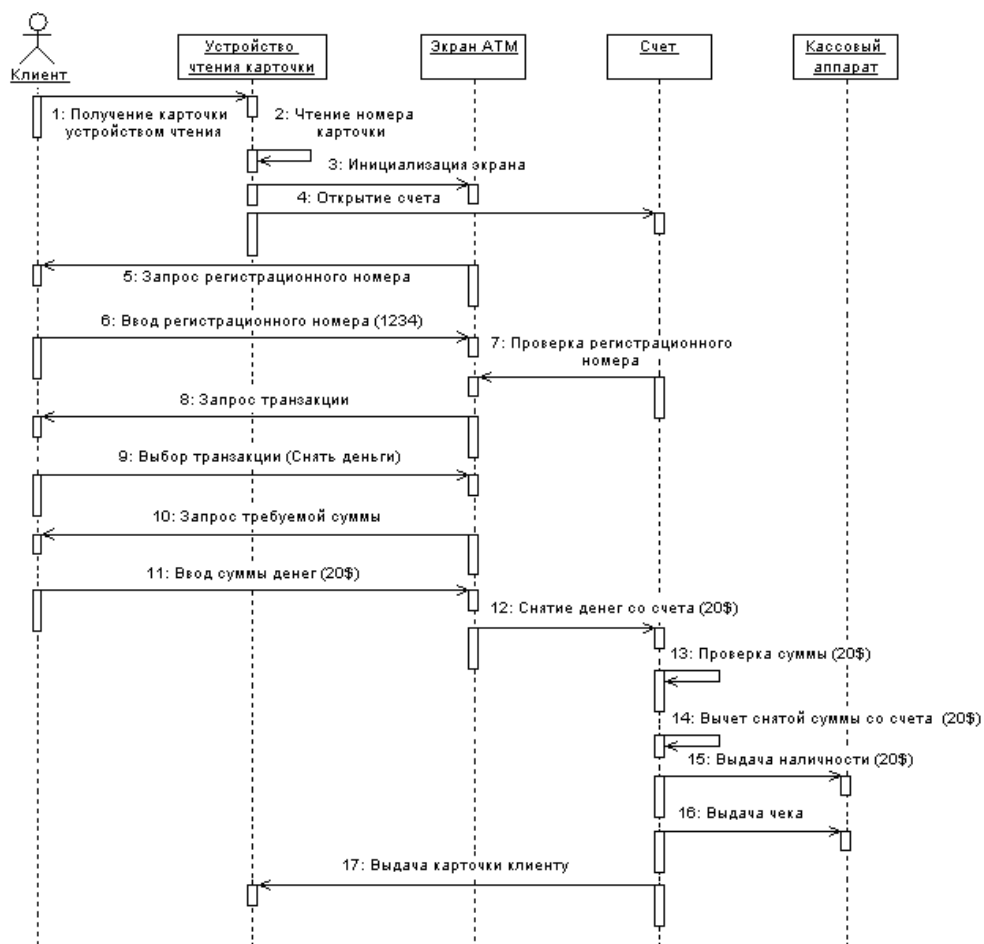
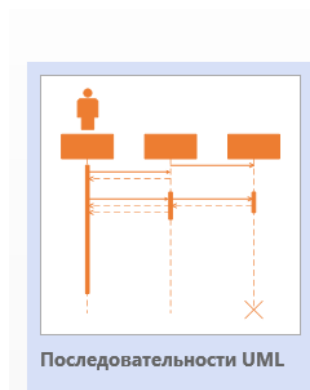


Рисунок 2. Пример построения *Диаграммы последовательностей*

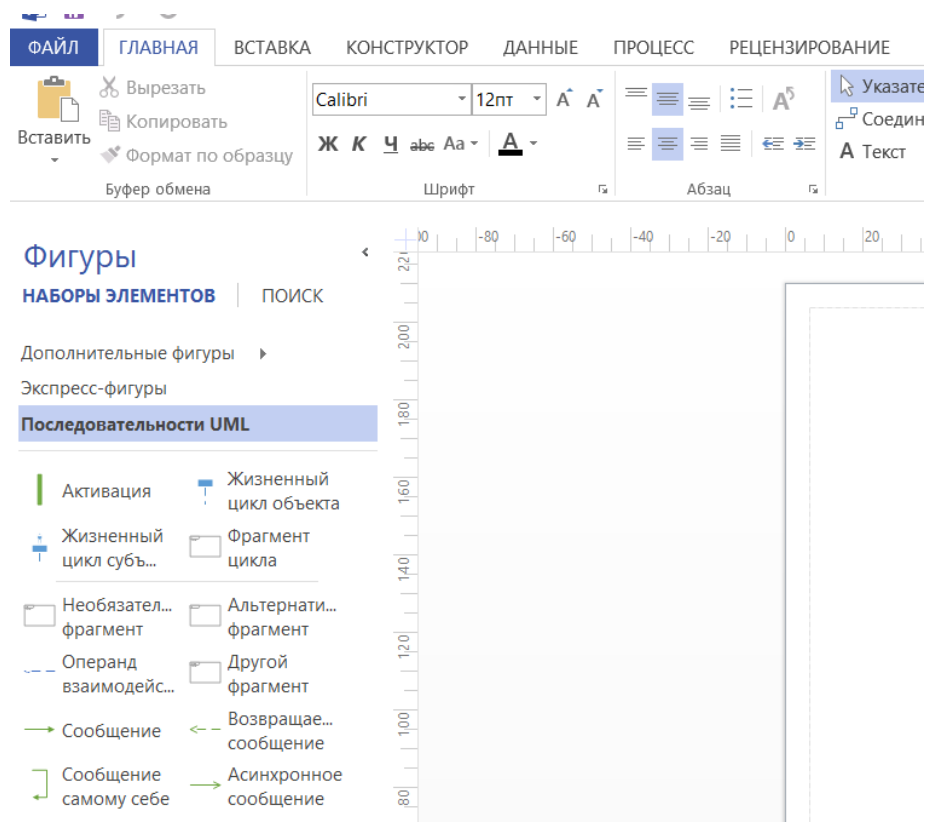
Таким образом, диаграмма последовательности иллюстрирует последовательность действий, реализующих вариант использования “Снять деньги со счета” на примере снятия клиентом 20\$. Глядя на эту диаграмму, пользователи знакомятся со спецификой своей работы. Аналитики видят последовательность (поток) действий, 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.

Методические рекомендации по выполнению работы

1. Запустить редактор MS Visio. Выбрать категорию шаблонов «Программы и базы данных», шаблон «Схема модели UML» - «Схема последовательности UML», нажать на кнопку «Создать».



2. Дать осмысленное имя созданной модели.
3. Ознакомиться с интерфейсом программы и набором команд для построения диаграммы последовательности UML.



4. Определить объекты, которые будут участвовать во взаимодействии и разместить их на листе с диаграммой.
5. Определить сообщения, которыми будут обмениваться объекты и расположить их на схеме.

6. Для построения диаграммы коммуникации необходимо разместить на рабочем листе объекты диаграммы, без учета временной шкалы и сообщения, передаваемые между этими объектами.

Варианты заданий

1. Программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
2. Программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
3. Приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.
4. Приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и несколько математических функции.
5. Программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.
6. Программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
7. Программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.

8. Программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
9. Программный модуль «Картотека агентства недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки на обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.
10. Программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.
11. Программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.
12. Программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
13. Программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.
14. Программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

15. Программный модуль «Поликлиника. Учет больных», содержащий сведения о врачах, больных, их датах и времени приема, а также назначениях. Программный модуль предназначен для организации записи пациента к определенному врачу, конкретной специализации, в отведенное время.
16. Программный модуль «Гостиница. Размещение клиентов», содержащий информацию о гостинице, ее номерах и посетителях. Посетитель может забронировать номер в отеле, соответствующего класса на определенную дату и время.
17. Программный модуль «Страховая компания. Заключение договоров», содержащий сведения о страховой компании, ее страховых агентах, условиях страхования и заключенных договорах с определенным клиентом.
18. Программный модуль «Театр. Продажа билетов.», содержащий сведения о спектаклях, проводимых в театре, его составе, и позволяющей оформить покупку и бронь билета.
19. Программный модуль «Компьютерный сервисный центр.», позволяющий вести учет сервисных работ с компьютерной и оргтехникой.
20. Программный модуль «Салон красоты. Оказание услуг», позволяющий вести запись на оказание услуг и производить учет оказанных услуг.
21. Программный модуль «Химчистка. Учет заказов», позволяющий производить учет оказанных услуг.
22. Программный модуль «Аптека. Поступление и продажа лекарств», позволяющий производить учет поступивших в аптеку лекарств, а также получать сведения о их продаже.
23. Программный модуль «Фитнес-клуб», содержащий информацию о клубе и услугах, которые он предоставляет. А также вести учет клиентов.
24. Программный модуль «Оптовый склад. Заключение договоров с поставщиками».
25. Программный модуль «Мебельный салон. Учет заказов».

Контрольные вопросы

1. Каково назначение диаграмм взаимодействия?
2. Какие диаграммы относятся к диаграммам взаимодействия?
3. Назовите основные графические элементы диаграммы последовательности и диаграммы коммуникации. Дайте им определение.
4. Какие виды сообщений могут быть изображены между объектами на диаграмме?
В чем их суть?