

## **Лабораторная работа №5. Моделирование поведения системы**

### **1. Цель работы**

Научиться моделировать поведение системы на основе диаграммы классов, CRC-карт и диаграммы последовательности с использованием MS Visual Studio 2010.

### **2. Порядок выполнения работы**

1. Изучение методологии моделирования диаграмм классов на основе языка UML.
2. Построение диаграммы классов для выбранной системы.
3. Оформление CRC-карт.
4. Изучение методологии моделирования диаграмм взаимодействия (последовательности) на основе языка UML.
5. Построение диаграмм взаимодействия (последовательности) для проверки избыточности CRC-карт и корректности диаграммы классов системы.
6. Оформление и сдача отчета преподавателю.

### **3. Краткая теория**

#### **3.1. Методология моделирования диаграммы классов на основе языка UML**

*Диаграмма классов* служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования.

Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Таким образом, сообщения не только передают некоторую информацию, но и требуют от принимающего объекта выполнения ожидаемых действий. Сообщения могут инициировать выполнение операций объектом соответствующего класса, а параметры этих операций передаются вместе с сообщением. На диаграмме последовательности все сообщения упорядочены по времени своего возникновения в моделируемой системе.

Предполагается, что окончательный вариант диаграммы содержит наиболее полное описание классов, которые состоят из трех разделов или секций. Иногда в обозначениях классов используется дополнительный четвертый раздел, в котором приводится семантическая информация справочного характера или явно указываются исключительные ситуации.

Даже если секция атрибутов и операций является пустой, в обозначении класса она выделяется горизонтальной линией, чтобы сразу отличить класс от других элементов языка UML.

Объект (object) является отдельным экземпляром класса, который создается на этапе выполнения программы. Он имеет свое собственное имя и конкретные значения атрибутов. В силу самых различных причин может возникнуть необходимость показать взаимосвязи не только между классами модели, но и между отдельными объектами, реализующими эти классы.

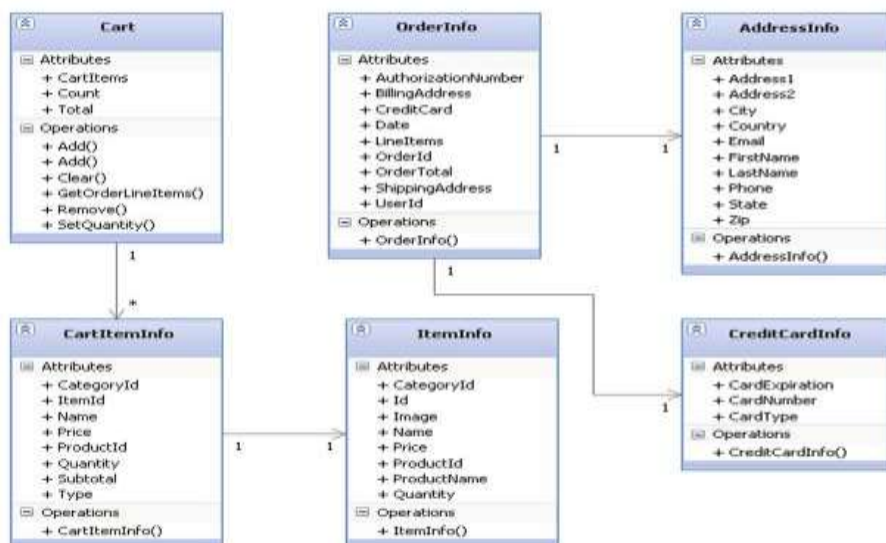
Для графического изображения объектов используется такой же символ прямоугольника, что и для классов. Отличия проявляются при указании имен объектов, которые в случае объектов обязательно подчеркиваются (рис. 1). При этом запись имени объекта представляет собой строку текста "имя объекта:имя класса", разделенную двоеточием (рис. 1 а, б). Имя объекта может отсутствовать, в этом случае предполагается, что объект является анонимным, и двоеточие указывает на данное обстоятельство (рис. 1, г). Отсутствовать может и имя класса. Тогда указывается просто имя объекта (рис. 1, в). Атрибуты объектов принимают конкретные значения.

При изображении диаграммы объектов нужно помнить, что каждый объект представляет собой экземпляр соответствующего класса, а отношения между объектами описываются с помощью связей, которые являются экземплярами соответствующих отношений. При этом все связи изображаются сплошными линиями.

#### Примеры построения диаграммы классов:

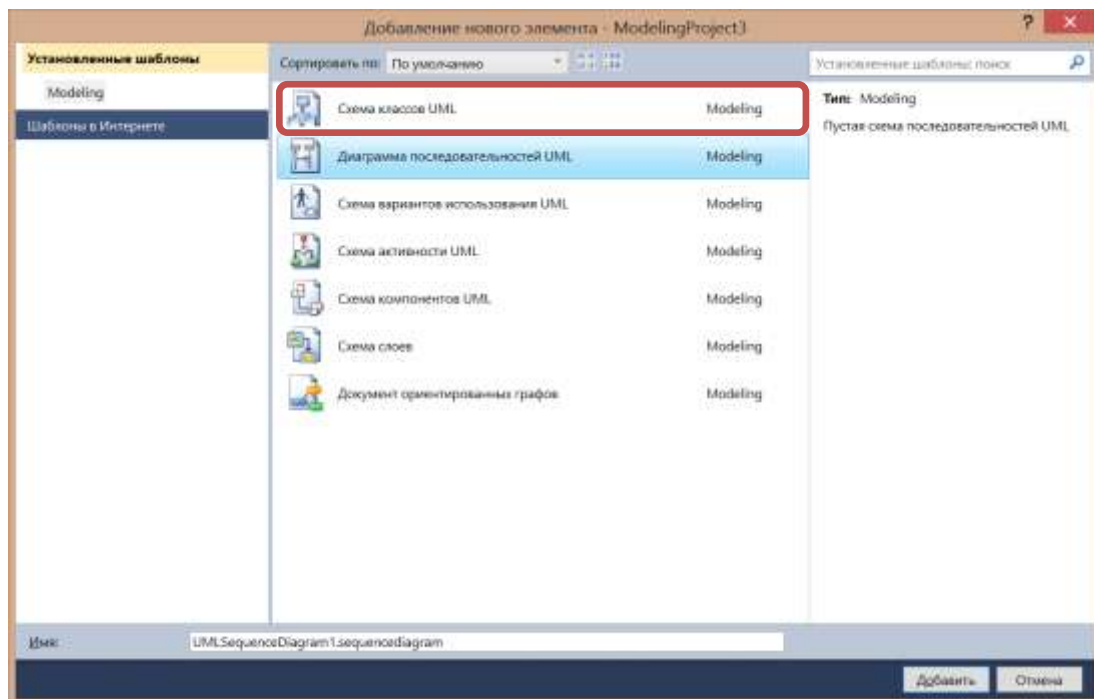


cd Order System



## 3.2. Построение диаграммы классов

Добавить новую диаграмму в свой проект через *Проект - Добавить новый элемент*:



Информация о правилах построения диаграмм последовательностей в Visual Studio.

<http://msdn.microsoft.com/ru-ru/library/dd409437.aspx>

msdn

Главная Новостная Библиотека Обучение Ссылки Microsoft Водные Поддержка

Решение (Русский) | Ввод | Поиск MSDN и помощник Bing

Данная статья переведена автоматически. Поставьте указатель мыши на предложенные статьи, чтобы просмотреть исходный текст. [Дополнительные сведения](#)

Перевод: Текст оригинала

## UML-схемы классов: справочные материалы

Visual Studio 2012 | Другие версии: 4 из 4 оценили этот материал как полезный | [Оценить эту тему](#)

UML-схема классов описывает структуры объектов и сведений, используемые для внутренней организации приложения и для взаимодействия с пользователем. Кроме того, схема предоставляет сведения об этих структурах независимо от конкретной реализации. Ее классы и отношения могут реализовываться несколькими способами, например в таблице базы данных, XML-узлы или сочетаниями программных объектов.

**Примечание**

Этот раздел посвящен UML-схемам классов. Существует другой вид схем классов, схема классов .NET, которая используется для визуализации программного кода. Дополнительные сведения см. в разделе [Разработка и просмотр классов и типов](#).

Чтобы создать UML-схему классов, в меню **Архитектура** выберите пункт **Создать схему**. Дополнительные сведения о создании UML-схем классов см. в разделе [UML-схемы классов: правила работы](#). Дополнительные сведения об общих принципах создания схем моделирования см. в разделе [Практическое руководство. Изменение моделей и схем UML](#).

### Чтение схем классов

В этом разделе в таблице описаны элементы, которые можно увидеть на UML-схеме классов. Сведения о свойствах этих элементов см. в следующем разделе.

- Свойства типов на UML-схемах классов
- Свойства атрибутов на UML-схемах классов
- Свойства операций на UML-схемах классов
- Свойства ассоциаций на UML-схемах классов

### 3.3. CRC-карты

Это листки бумаги 10х15. Они разделены на 3 части и выглядят следующим образом:

Название класса	
Действия, которые он выполняет (всегда начинаются с глагола)	Классы, с которыми данный класс обменивается информацией

CRC-карты акцентируют внимание дизайнера на сущности класса и скрывают от него детали, рассмотрение которых на данном этапе будет контрпродуктивным. CRC-карты также заставляют дизайнера воздержаться от назначения классу слишком многих обязанностей.

### 3.4. Методология моделирования диаграмм последовательности на основе языка UML

*Диаграммы последовательности* являются моделями, описывающими поведение взаимодействующих групп объектов.

В языке UML взаимодействие элементов рассматривается в информационном аспекте их коммуникации, т.е. взаимодействующие объекты обмениваются между собой некоторой информацией. При этом информация принимает форму законченных сообщений.

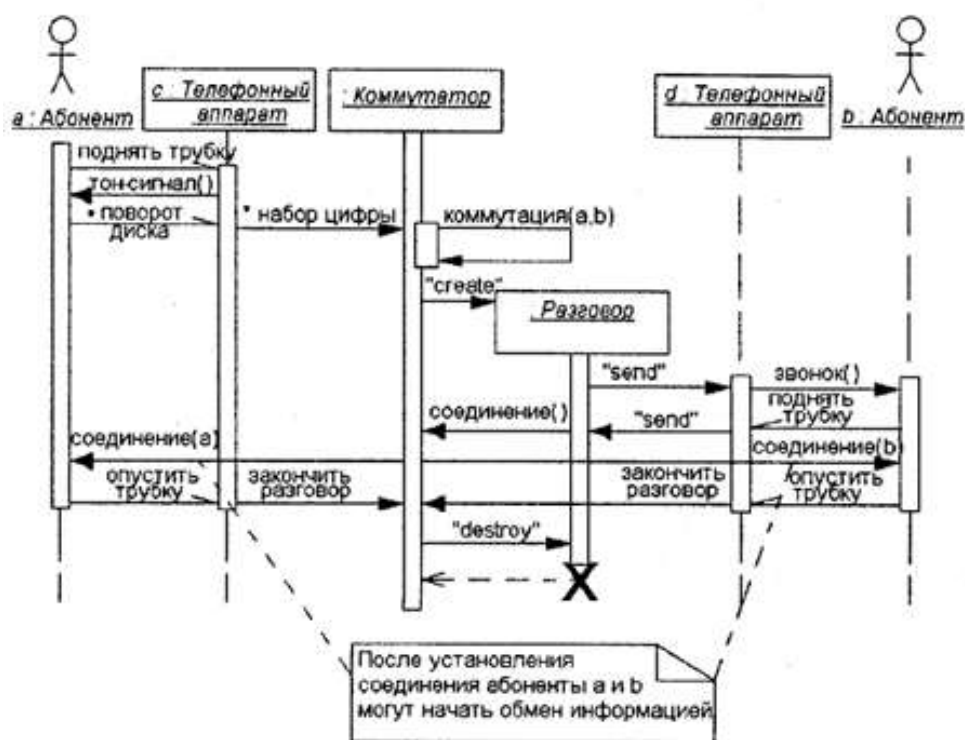
На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами.

Цель взаимодействия в контексте языка UML заключается в том, чтобы описать связи между множеством взаимодействующих объектов. Каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой. В этом смысле сообщение (message) представляет собой законченный фрагмент информации, который отправляется одним объектом другому. При этом прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

Таким образом, сообщения не только передают некоторую информацию, но и требуют от принимающего объекта выполнения ожидаемых действий. Сообщения могут инициировать выполнение операций объектом соответствующего класса, а параметры этих операций передаются вместе с сообщением. На диаграмме последовательности все сообщения упорядочены по времени своего возникновения в моделируемой системе.

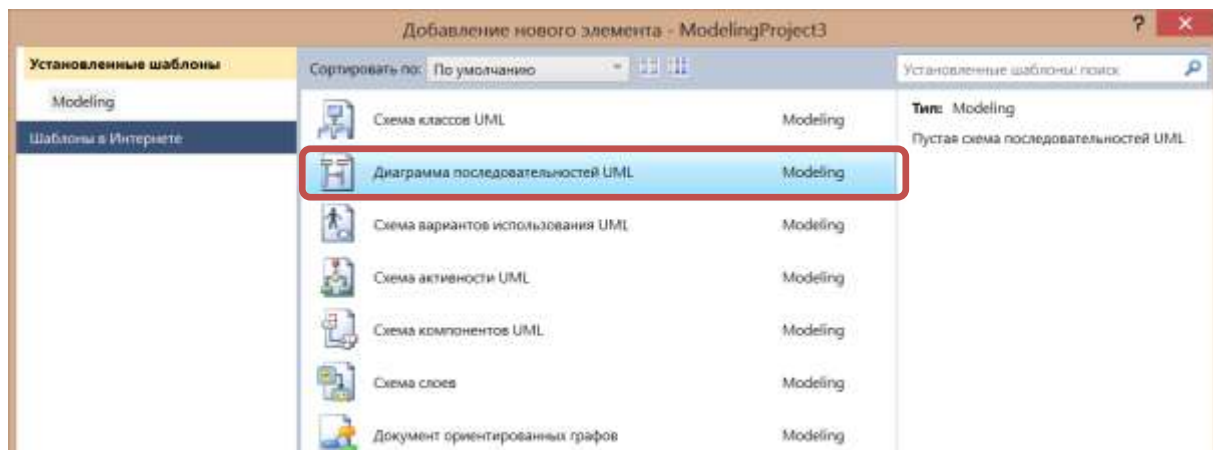
В таком контексте каждое сообщение имеет направление от объекта, который инициирует и отправляет сообщение, к объекту, который его получает.

Пример построения диаграммы последовательности: диаграмма последовательности для моделирования телефонного разговора



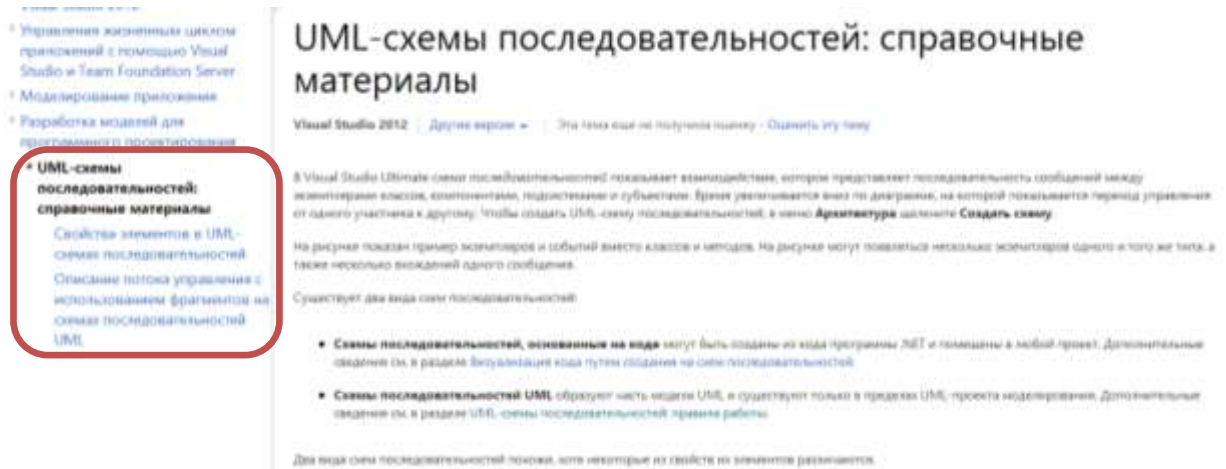
### 3.5. Построение диаграммы последовательностей

Добавить новую диаграмму в свой проект через *Проект - Добавить новый элемент*:



Информация о правилах построения диаграмм последовательностей в Visual Studio.

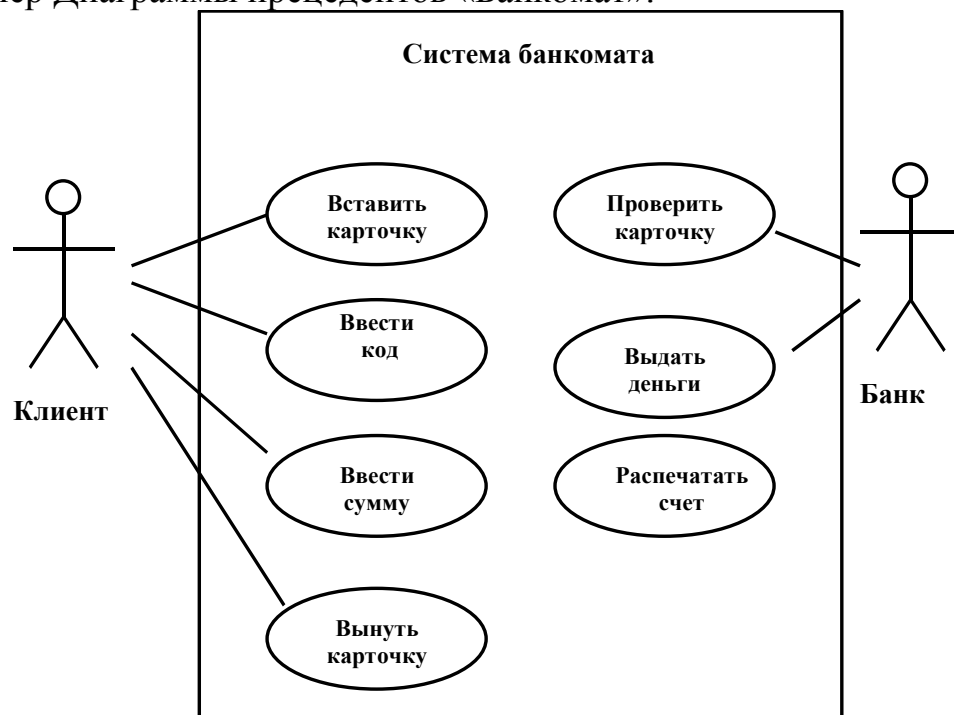
<http://msdn.microsoft.com/ru-ru/library/dd409377.aspx>



### 3.6 Пример: система банкомата

1. Определить варианты использования системы и описать их (см. лабораторные работы №2 и №3).
2. Определить классы проектируемой системы.
3. Создать CRC-карты для всех классов системы.
4. Построить диаграмму взаимодействия (последовательности).

Пример Диаграммы прецедентов «Банкомат»:



Эта диаграмма дает понять, что будет делать система, как она будет функционировать.

Далее выявляют *классы*, которые необходимо будет создать в программе для реализации системы. В случае банкомата это: *клиент*, *банк*, *служба безопасности банка*, сам *банкомат* и т.д. Строится диаграмма классов банкомата.

Далее оформляются CRC-карты.

На примере банкомата:

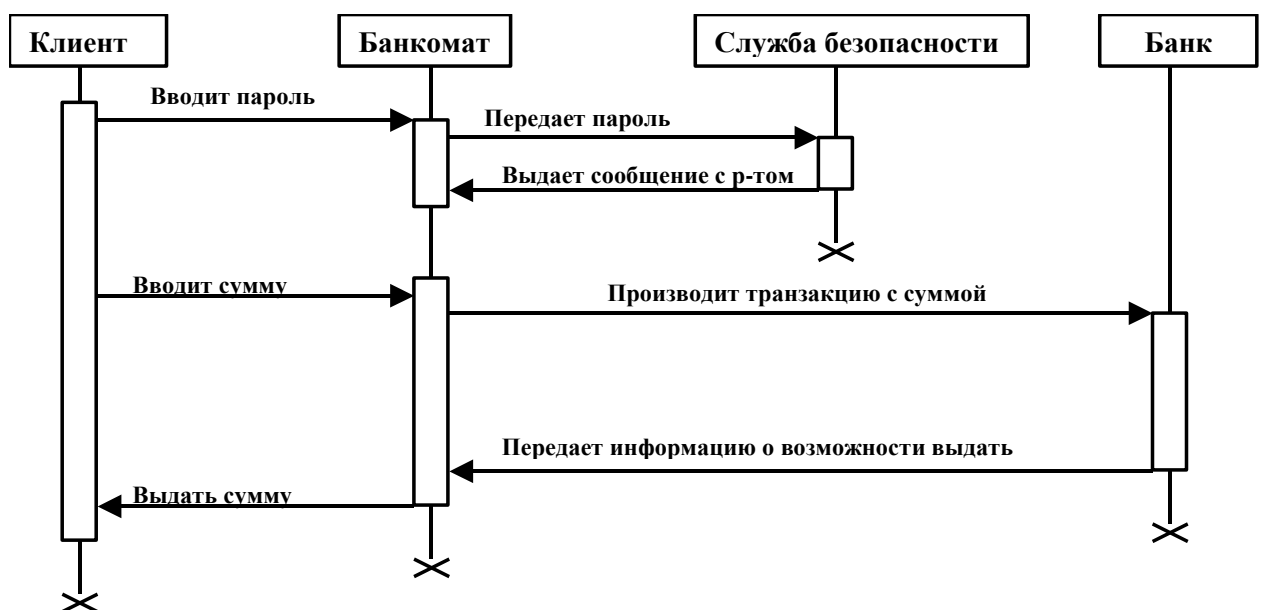
Клиент	
<ol style="list-style-type: none"><li>1. Вставляет карточку в банкомат</li><li>2. Вводит пароль</li><li>3. Указывает тип операции (снять деньги, просмотреть остаток)</li><li>4. Вводит сумму</li><li>5. Получает деньги</li><li>6. Вынимает карточку</li></ol>	Банкомат

Банкомат	
<ol style="list-style-type: none"><li>1. Отображает информацию для клиента</li><li>2. Передает информацию в банк</li><li>3. Отсчитывает купюры</li><li>4. Распечатывает счет</li></ol>	Клиент Банк Служба безопасности банка

Служба безопасности банка	
<ol style="list-style-type: none"><li>1. Проверяет пароль</li><li>2. Проверяет подлинность карточки</li><li>3. Идентифицирует клиента</li><li>4. Следит за правильностью транзакций операций с деньгами</li></ol>	Банк Банкомат

Банк	
<ol style="list-style-type: none"> <li>1. Проверяет возможность выдачи средств</li> <li>2. Сообщает о наличии денег</li> <li>3. Выдает информацию об остатке</li> <li>4. Хранит информацию о счете клиента</li> </ol>	Банкомат Служба безопасности банка

Для проверки достаточности или избыточности придуманных классов, а также корректности их взаимодействия строится диаграмма взаимодействия:



### 3.3 Сдача отчета преподавателю

1. Отчет сдается в электронном виде: каталог созданного в Visual Studio проекта моделирования архивируется и высылается по адресу [typer2010@mail.ru](mailto:typer2010@mail.ru). В теме письма указать ФИО, группа, номер и тема лабораторной работы, название проектируемой системы.
2. Отчет также содержит текстовое описание (файл в формате MS Word) процесса создания диаграммы классов, CRC-карт и диаграммы взаимодействия для выбранной предметной области, также отчет содержит соответствующие выводы по результатам моделирования.