

# Начало работы StarUML

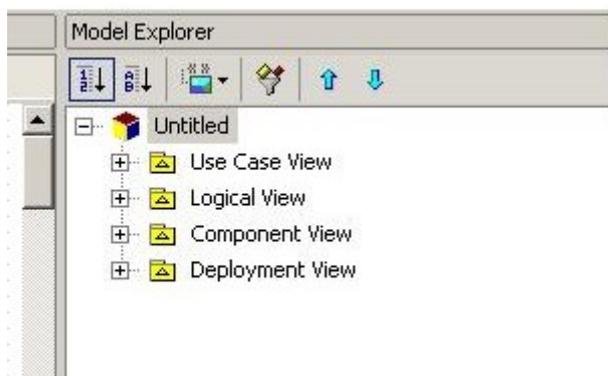
Запустите StarUML

В диалоговом окне выберите Rose Approach



Изучите структуру окна

Найдите окно Model Explorer



- Папка или пакет **Use Case View** соответствует этапу анализа
- Пакет **Logical View** - логическое проектирование
- Пакет **Component View** - компоненты системы
- Пакет **Deployment View** - размещение

Средняя часть окна представляет собой окно диаграммы, с левой стороны расположена панель инструментов. Для каждой диаграммы панель своя.

Центральной диаграммой Use Case View является диаграмма прецедентов.

Центральной диаграммой Logical View является диаграмма классов.

## Создание диаграммы прецедентов

 Actor **Действующее лицо (actor)** - это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ.

Несмотря на то, что на диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок, действующее лицо может также быть внешней системой, которой необходима информация от данной системы. Показывать на диаграмме действующих лиц следует только в том случае, когда им действительно необходимы некоторые варианты использования.

 UseCase **Вариант использования** представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать.

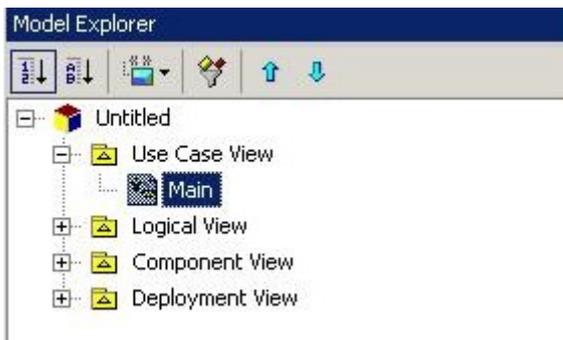
 DirectedAssociation Устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

 Include связь включения (**include**) между двумя вариантами использования, которая указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательности поведения другого варианта использования.

 Extend связь расширения (**extend**) отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования.

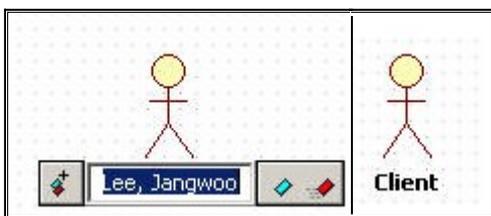
Перед созданием диаграммы выделите пользователей, разрабатываемой системы.

Нажмите на "+" около пакета Use Case View и выполните двойной щелчок мыши по диаграмме main (главная)

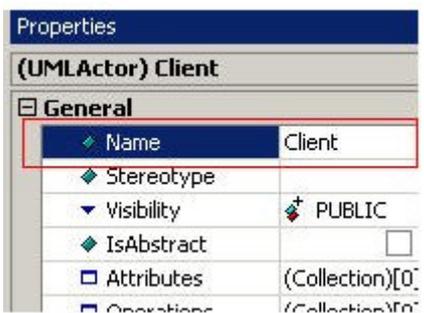


Используя инструменты диаграммы Use Case создайте диаграмму. На диаграмме расположите всех пользователей и укажите основные прецеденты. Сделайте акцент на общих прецедентах.

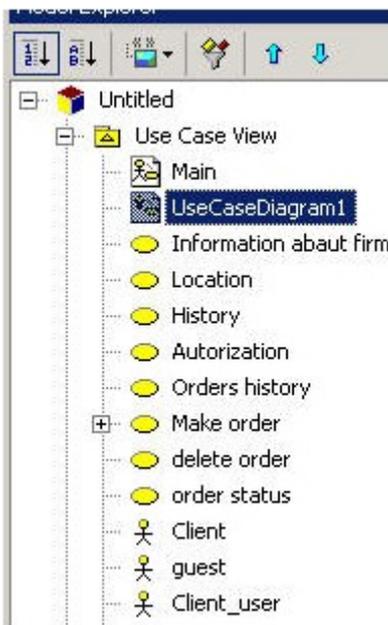
При создании актера или прецедента необходимо ввести его имя в появившемся окне.



Изменить имя любого объекта можно при помощи окна Properties (свойства), указав его имя

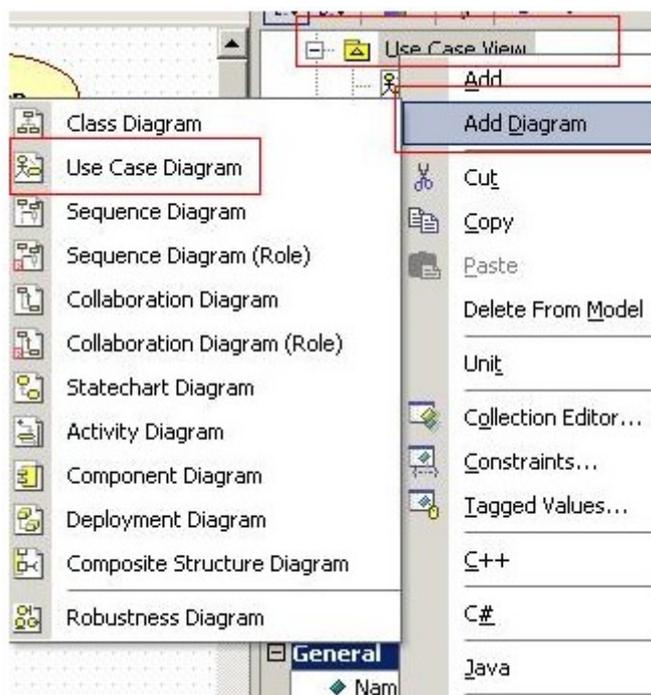


Обратите внимание, что в браузере появляются, созданные Вами объекты.



Далее для каждого актера создайте новую диаграмму.

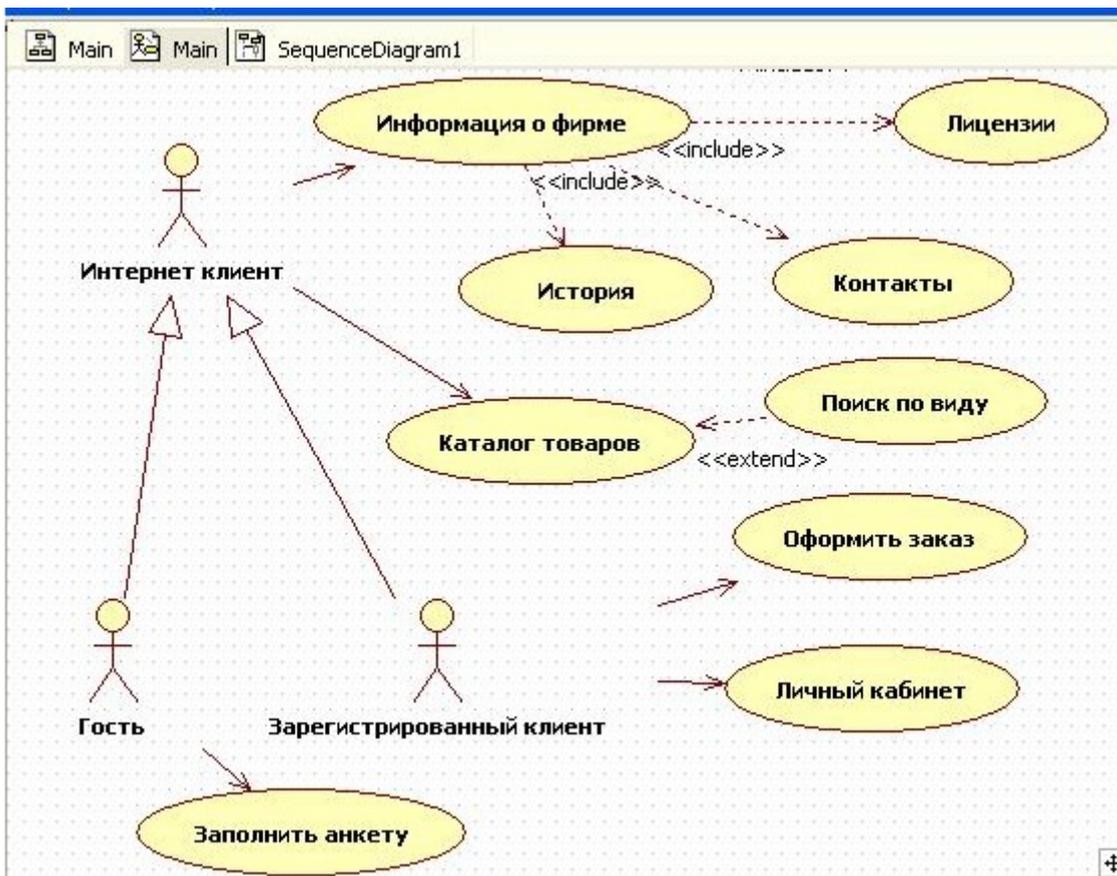
Для создания новой диаграммы выделите пакет Use Case View и воспользуйтесь контекстным меню, выберите Add diagram (создание диаграммы) и Use Case Diagram



Перетащите на диаграмму, созданные ранее прецеденты и добавьте новые. При необходимости воспользуйтесь связями include и extend.

Для наиболее сложных прецедентов создайте текстовый документ поток событий. К сожалению StarUml не позволяет привязать его к диаграмме, но этот документ поможет создать следующую диаграмму

Пример фрагмента use case diagram (диаграммы прецедентов или вариантов использования)



## Диаграмма активности

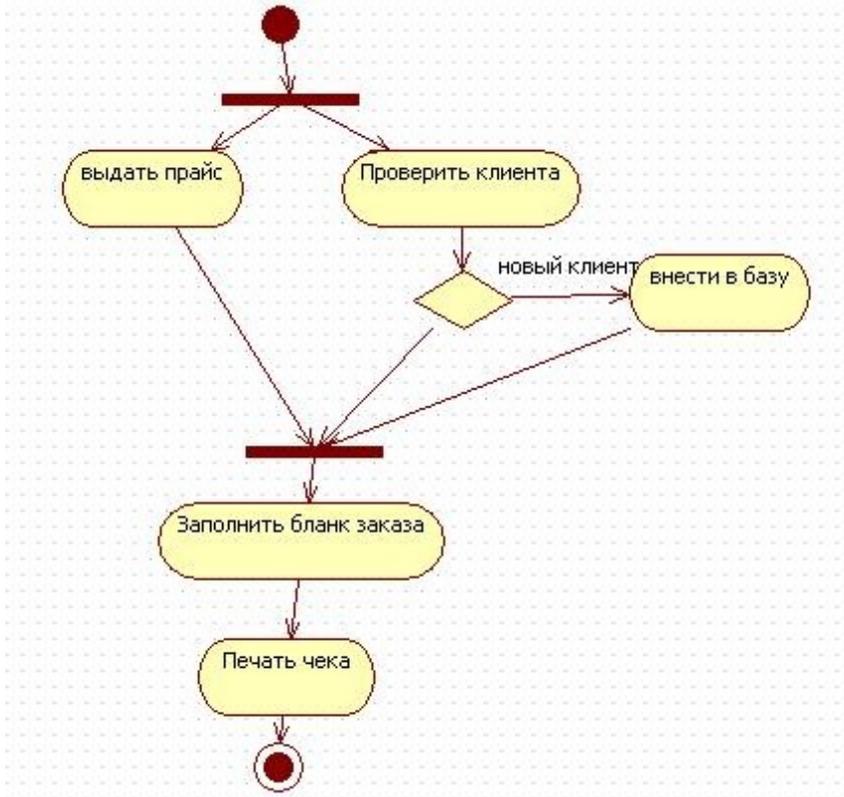
Диаграммы активности (деятельности) частный случай диаграмм состояний. Каждое состояние есть выполнение некоторой операции и переход в следующее состояние

<b>ActionState</b>	Состояние действий с некоторым входным действием и по крайней мере одним выходящим из состояния перехода.
<b>Decision</b>	Ветвление, применяемое в случаях, когда последовательно выполняемая деятельность должна разделиться на альтернативные ветви в зависимости от значения некоторого промежуточного результата. В этот ромб может входить только одна стрелка от того состояния, после выполнения которого, поток управления должен быть продолжен по одной из взаимно исключающих ветвей. Выходящих стрелок может быть две или более, но для каждой из них явно указывается соответствующее условие
<b>Transition</b>	Переход, т. е. отношение между двумя последовательными действиями.
<b>Synchronization</b>	Распараллеливание вычислений. Может иметь один входящий переход и несколько выходящих, либо несколько входящих и один выходящий
<b>InitialState</b> <b>FinalState</b>	Начальное и конечное состояние

Swimlane(Vertical) — Swimlane(Horizontal)	Дорожки
--	---------

Выделите пакет use case view и при помощи контекстного меню создайте диаграмму активности или действия (Activity Diagram)

Пример диаграммы - оформление заказа в офисе



## Дорожки

Одна из важных областей применения диаграмм активности связана с моделированием бизнес процессов. Деятельность любой компании, также представляет совокупность отдельных действий, направленных на достижения отдельного результата. Однако применительно к бизнес процессам желательно выполнение каждого действия ассоциировать с конкретным подразделением. В этом случае подразделение несет ответственность за реализацию отдельных действий, а сам бизнес процесс представляется в виде переходов действий из одного подразделения к другому.

Все действия делятся на отдельные группы, которые отделяются друг от друга вертикальными линиями. Группа состояний между этими линиями выполняется отдельным подразделением (отделом, группой, филиалом) компании.

Названия подразделений явно указываются в верхней части дорожки. Пересекать линию дорожки могут только переходы, которые в этом случае обозначают выход или вход потока управления в соответствующих подразделениях компании

Пример диаграммы активности с дорожками

