Лабораторная работа 3 Структурный подход к проектированию ИС. Создание объектной модели предметной области

# Цель работы

Получить опыт практической работы в создании иерархий классов предметной области с помощью UML диаграмм.

### Введение

**UML** (Unified Modeling Language– унифицированный язык моделирования) – [язык](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [графического](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](http://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F). UML является языком широкого профиля, это [открытый стандарт](http://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82), использующий графические обозначения для создания [абстрактной модели](http://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)[системы](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна [генерация кода](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%BE%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F).

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для [моделирования бизнес-процессов](http://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [системного проектирования](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и отображения [организационных структур](http://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0).

Средством визуализации моделей может быть пакет MS Visio, являющийся одной из составных частей MS Office.

3.1. Элементы нотации UML

* актор (actor) – (действующее лицо) объект, который взаимодействует с системой, но не является ее частью;
* диаграммы сценариев (use-case);
* диаграммы классов (class);
* диаграммы состояния (statechart);
* диаграммы активности (activity);
* диаграммы последовательности (sequence);
* диаграммы взаимодействия (collaboration);
* диаграммы компонентов (component);
* диаграммы топологий (deployment).

*Диаграмма вариантов использования* (use case diagram)(диаграмма прецедентов, диаграмма сценариев) показывает набор сценариев, актеров и их отношений. *Диаграмма вариантов использования* представляет собой наиболее общую концептуальную модель системы, которая является исходной для построения всех остальных диаграмм. Эти диаграммы особенно важны при орга­низации и моделировании поведения системы, задании требований заказчика к системе.

*Диаграмма классов* (class diagram) показывает набор классов, интерфейсов, коопераций и ихотношений. При моделировании объектно-ориентированных систем диаграммы классов используются наиболее часто. Диаграммы классов можно отнести к логической модели, отражающей статические аспекты структурного построения системы. Диаграммы классов, включающие активные классы, обеспечивают статическое представление процессов системы.

*Диаграмма объектов* (object diagram)показывает набор объектов и их отношения. Диаграмма объек­тов представляет статический «моментальный снимок» с экземпляров предметов, которые находятся на диаграммах классов. Как и диаграммы классов, эти диаграммы обеспечивают статическое проектное представление или статическое представ­ление процессов системы (с точки зрения реальных случаев).

*Диаграмма последовательности и диаграмма сотрудничества (кооперации)* – это разновиднос­ти *диаграмм взаимодействия* (interaction diagrams). Диаграмма взаимодействияпоказывает взаимодействие, включающее набор объек­тов и их отношений, а также пересылаемые между объектами сообщения. Диа­граммы взаимодействия обеспечивают динамическое представление системы.

*Диаграмма последовательности* (sequence diagram) – это диаграмма взаимодействия, которая выде­ляет упорядочение сообщений по времени.

*Диаграмма сотрудничества* (диаграмма кооперации) (collaboration diagram) – это диаграмма взаимодей­ствия, которая выделяет структурную организацию объектов, посылающих и при­нимающих сообщения.

Диаграммы последовательности и диаграммы кооперации изоморфны. Это означает, что одну диаграмму можно трансформировать в другую диаграмму.

*Диаграмма схем состояний* (statechart diagram)показывает конечный автомат, представляет состоя­ния, переходы, события и действия. Диаграммы схем состояний обеспечивают ди­намическое представление системы. Они особенно важны при моделировании по­ведения интерфейса, класса или кооперации. Эти диаграммы выделяют такое поведение объекта, которое управляется событиями.

*Диаграмма деятельности* (activity diagram) – специальная разновидность диаграммы схем состоя­ний, которая показывает поток от действия к действию внутри системы. Диаграм­мы деятельности обеспечивают динамическое представление системы. Они осо­бенно важны при моделировании функциональности системы и выделяют поток управления между объектами.

Диаграммы взаимодействия, диаграммы состояний и диаграммы деятельности относятся к *диаграммам поведения* (behavior diagrams). Диаграммы поведения представляют логическую модель системы и отражают динамические аспекты ее функционирования.

Диаграмма компонентов и диаграмма размещения – это разновидности *диаграмм реализации* (implementation diagrams). Эти диаграммы служат для представления физических компонентов системы и относятся к ее физической модели.

*Компонентная диаграмма* (component diagram) показывает организацию набора компонентов и зависи­мости между компонентами. Компонентные диаграммы обеспечивают статичес­кое представление реализации системы. Они связаны с диаграммами классов в том смысле, что в компонент обычно отображается один или несколько классов, ин­терфейсов или коопераций.

*Диаграмма размещения* (диаграмма развертывания, диаграмма топологии) (deployment diagram) показывает конфигурацию физических элементов системы, а также компоненты, размещенные в них. Диаграммы развертывания обеспечивают статическое представление размещения системы. Они связаны с компонентными диаграммами в том смысле, что узел обыч­но включает один или несколько компонентов

3. 2. Правила разработки диаграмм вариантов использования

Не моделируйте связи между действующими лицами. По определению акторы находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к ее компетенции.

Не соединяйте стрелкой непосредственно два варианта использования, кроме случаев отношений между ними (обобщение, включение и расширение). Диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения.

Каждый *конкретный* вариант использования должен быть инициирован действующим лицом. Это означает, что всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования. Исключением являются отношения между вариантами использования (обобщение, включение и расширение).

База данных – это слой, находящийся под диаграммой. С помощью одного варианта использования можно вводить данные в базу,   
а получать их с помощью другого. Для изображения потока информации не нужно рисовать стрелки от одного варианта использования   
к другому.

3.3. Порядок выполнения лабораторной работы

1. Запустите MS Office Visio 2007.

MS Visio позволяет строить различные диаграммы. Поэтому  перед началом работы необходимо выбрать соответствующий тип диаграммы :

* на панели Категория выбрать SoftWare;
* на панели Шаблоны выбрать UML Model Diagram.

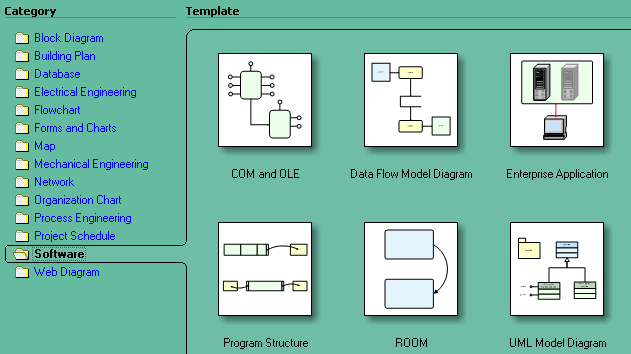


Рисунок 3.1. Выбор средств моделирования

***Масштаб.*** На панели Меню выберите **Вид**, затем измените масштаб изображения для активной диаграммы или для всех диаграмм в модели на тот, который Вам нужен.

**Образцы объектов модели** (рис. 2.4) обеспечивают стандартизацию как общего вида модели, так и каждого ее элемента.

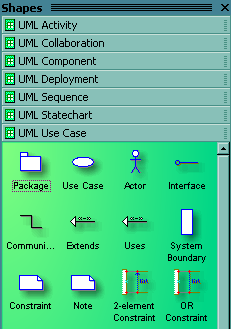


Рисунок 3.2. Образцы объектов модели

**Область для рисования**  (рис. 3.3) предназначена собственно для построения модели.

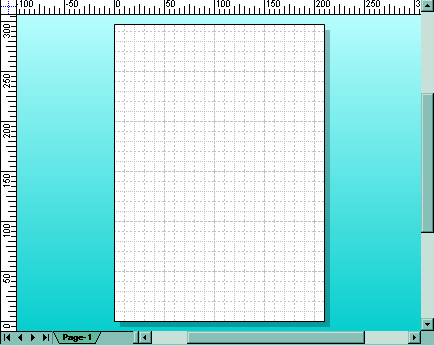


Рисунок 3.3. Область рисования модели

2. Для выбранной предметной области:

* постройте две диаграммы использования (UseCase) : общей функциональности и раскрытие одного из сценариев;
* для выбранного сценария постройте диаграмму последовательностей.

3. Оформить отчёт, который должен содержать:

* + титульный лист;
  + название лабораторной работы, цель;
  + построенные UML-диаграммы.