**Диаграмма состояний.**

Диаграмма состояний покажет нам все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния.

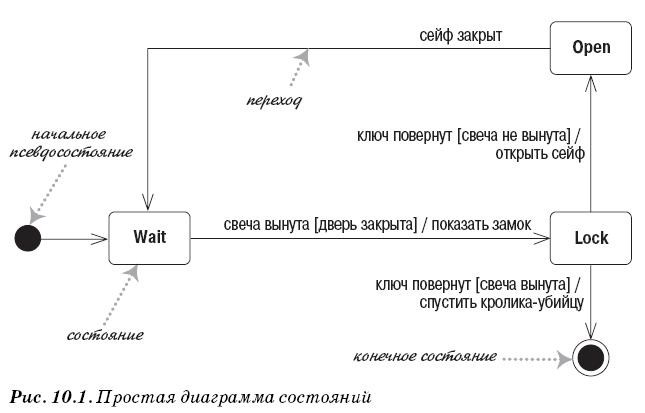


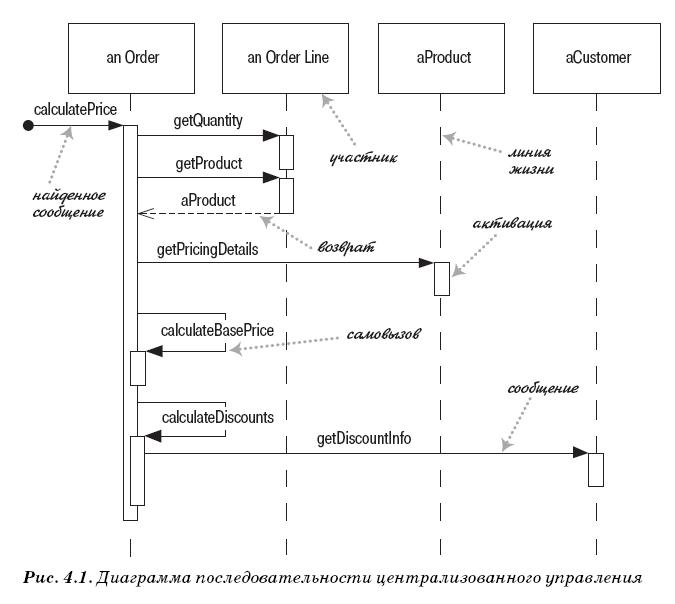
Диаграмма состояний должна создаваться только для объектов, обладающих каким-либо поведением.

**Целесообразно выбрать такой объект, который существенен для понимания описываемой предметной области.**

**Количество состояний объекта: минимум 5 шт.**

**Диаграмма последовательности.**

Диаграммы последовательности показывают взаимодействие, представляя каждого участника вместе с его **линией жизни**, которая идет вертикально вниз и упорядочивает сообщения на странице.



**На ваших диаграммах должно быть:**

*Участник* – это объект из вашей программы (Класс, страница, БД и т.д.)

*Сообщение* – это метод, который обращается к следующему участнику.

**Как необходимо выбрать процесс:** необходимо описать один из Use case, наиболее важный.

*Напоминаю, что в папке EXAMPLES есть пример.*

**Диаграмма компонентов.**

Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

**Тут все просто: открываем дерево проектов в вашей IDE и переносим его структуру на диаграмму.**

*Напоминаю, что в папке EXAMPLES есть пример.*

**Диаграммы классов.**

Диаграмма классов описывает структуру системы, классы системы, их атрибуты, методы и зависимости между классами.

**Здесь тоже все достаточно просто: открываем проект в IDE и генерируем диаграммы классов по пакетам.**

**Диаграмма развертывания.**

Диаграмма развертывания представляет физическое расположение системы, показывая, на каком физическом оборудовании запускается та или иная составляющая программного обеспечения.

**Ну я не знаю, что тут можно прокомментировать. На диаграмме нужно отобразить взаимодействие всех элементов: не забыть про Web Services.**

