Содержание

Глава

3

Конкретные примеры

Существует совсем немного вещей, которые придумать труднее, чем хороший пример.
Марк Твен (Mark Twain)

Основная задача

- Ознакомиться с конкретными примерами, рассматриваемыми в этой книге.

Введение

Рассматриваемые в этой книге конкретные примеры выбраны именно потому, что многие знакомы с подобными задачами. В книге рассматриваются интересные и сложные проектные решения, позволяющие сконцентрироваться на фундаментальных принципах ООА/П, анализе требований, UML и шаблонах, а не на постановке задач и предметной области.

Что дальше? После ознакомления с итеративным процессом в данной главе рассматриваются конкретные примеры и основные элементы уровня логики приложения, которые будут рассматриваться в книге. Следующая глава посвящена стадии начала разработки учебных приложений. При этом подчеркивается, что начало *не* является фазой каскадного процесса, на которой проводится ранний анализ всех требований

3.1. Какие вопросы рассматриваются и *не* рассматриваются при изучении конкретных примеров

Архитектура типичной информационной системы включает графический интерфейс пользователя, взаимодействие с базой данных, уровень логики приложения и взаимосвязь с другими программными системами или архитектурными компонентами (рис. 3.1).

> Хотя объектно-ориентированную технологию можно применять ко всем уровням, в данной книге основное внимание уделяется вопросам анализа и проектирования *базового уровня логики приложения*, а другие уровни рассматриваются только поверхностно.

При рассмотрении других уровней (например, интерфейса пользователя) мы, в основном, сосредоточимся на разработке их интерфейса с уровнем логики приложения.

Почему основное внимание в процессе ООА/П уделяется базовому уровню логики приложения? На этот вопрос можно ответить следующим образом.

- Другие уровни обычно в значительной мере зависят от платформы или от реализации. Например, чтобы изучить вопросы объектно-ориентированного проектирования Web-интерфейса или интерфейса пользователя на Java, необходимо детально разобраться с такими каркасами, как Struts или Swing. Если же в качестве платформы разработки выбрана .NET или Python, базовые знания должны быть совершенно другими.
- В отличие от этого объектно-ориентированное проектирование уровня логики приложения одинаково для всех технологий.
- Основные знания и навыки, приобретенные в процессе изучения ООП в контексте уровня логики приложения, применимы ко всем другим уровням или компонентам.
- Подход к проектированию (в том числе на основе шаблонов) быстро изменяется с появлением новых каркасов или технологий. Например, в середине 1990-х разработчикам приходилось создавать свои собственные доморощенные объектно-реляционные уровни доступа к базам данных. А спустя несколько лет они предпочли использовать свободно распространяемые решения с открытым кодом, такие как Hibernate (для технологии Java).

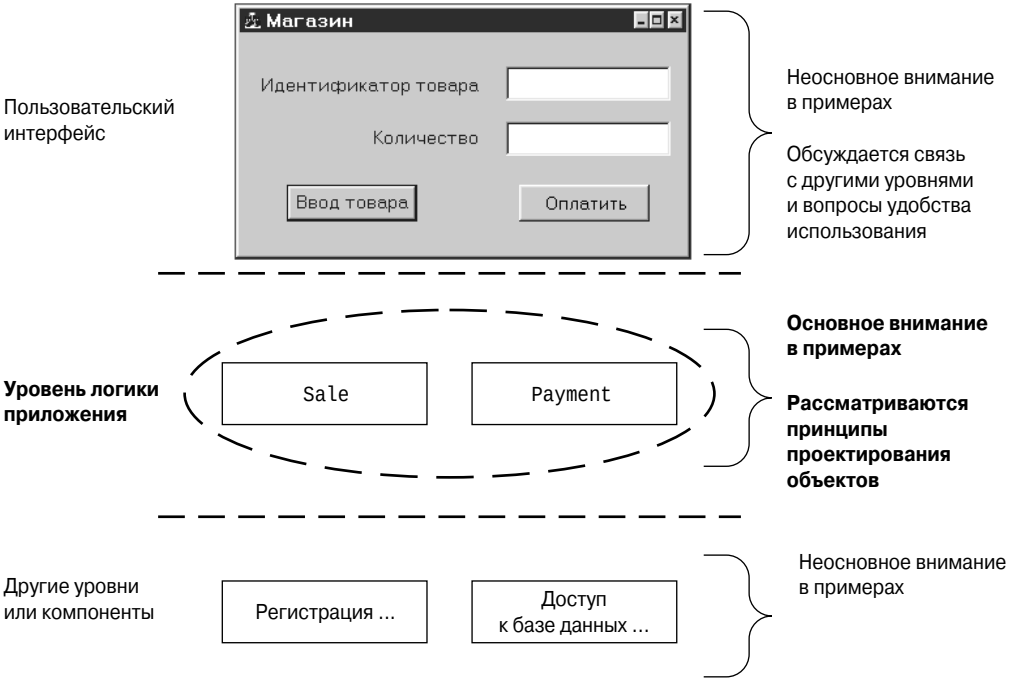

Рис. 3.1. Уровни и объекты типичной объектно-ориентированной системы с указанием степени детализации их описания в данной книге

3.2. Стратегия описания конкретных примеров: итеративная разработка + итеративное изучение

Структура материала в этой книге основывается на итеративной стратегии разработки. Объектно-ориентированный анализ и проектирование применяются к разработке конкретных систем в процессе реализации нескольких итераций. На первой итерации обеспечивается выполнение некоторых базовых функций, а на последующих расширяется функциональность систем (рис. 3.2).

В соответствии с итерационной стратегией разработки в книге будут последовательно представлены главы, посвященные объектно-ориентированному анализу и проектированию, системе обозначений языка UML и шаблонам. В главах, посвященных первой итерации, будут рассмотрены базовые вопросы анализа и проектирования, а также основные обозначения, принятые в UML. В главах, посвященных второй итерации, будут описаны новые идеи, новые обозначения UML, шаблоны и т.д. Это же касается и третьей итерации.

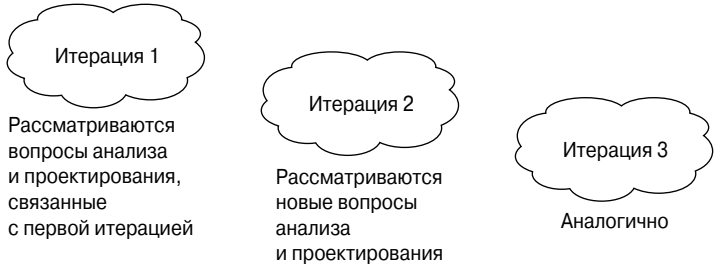

Рис. 3.2. Последовательность изучения материала соответствует итеративному процессу разработки приложения

3.3. Пример 1: POS-система NextGen

Первым примером, рассматриваемым в этой книге, является система автоматизации торговой точки — POS-система NextGen. Рассматриваемая задача очень интересна своими требованиями и проблемами, возникающими в процессе проектирования. Кроме того, она очень реалистична. Многие организации действительно разрабатывают POS-системы (point-of-sale system) на основе объектной технологии.

POS-система — это компьютеризированное приложение, предназначенное для организации товарооборота и обработки платежей в обычных магазинах. Система автоматизации торговли включает аппаратные компоненты (компьютер и устройство считывания штрихкода), а также программное обеспечение, выполняющее основные задачи системы. Это приложение связано с различными служебными программами, например с программой вычисления налогов, разработанной сторонними производителями, или с системой складского учета товаров. Подобные системы должны быть устойчивыми к сбоям, т.е. работоспособными при временном выходе из строя удаленных служб (например, системы складского учета товаров). В критических ситуациях они должны обслуживать продажу товаров и обеспечивать обработку хотя бы платежей наличными (чтобы хозяин магазина не обанкротился).

POS-система должна поддерживать различные типы клиентских терминалов и интерфейсов, в том числе клиентский терминал с Web-браузером ("тонкого" клиента), обычный персональный компьютер с графическим интерфейсом пользователя типа Java Swing, сенсорный ввод информации, беспроводный интерфейс и т.п.

Более того, коммерческие POS-системы должны уметь работать с различными категориями потребителей, для которых определены отдельные бизнес-правила. Для каждого потребителя может быть предусмотрена своя логика выполнения отдельных операций в рамках сценария использования системы, например, специфические действия при добавлении нового товара или создании новой продажи. Следовательно, необходимо предусмотреть механизм обеспечения этой гибкости и настройки системы.

На основе итеративной стратегии разработки мы выполним все необходимые этапы создания системы: формулировку требований, объектно-ориентированный анализ, проектирование и реализацию.

3.4. Пример 2: игра “Монополия”

Чтобы показать применимость основных приемов ООА/П к различным задачам, в качестве второго примера рассмотрим программную версию игры “Монополия”. Хотя предметная область и требования к этой системе не имеют ничего общего с коммерческими системами типа POS-системы NextGen, при разработке программы применяются те же методы моделирования предметной области, объектно-ориентированного проектирования, шаблоны и обозначения UML. Как и система автоматизации розничной торговли, программная версия игры “Монополия” реально существует и продается. Она также поддерживает разнообразное клиентское программное обеспечение и Web-интерфейс.

Мы не станем здесь описывать правила игры “Монополия”. Наверное, каждый человек в любой стране, будучи ребенком или подростком, играл в эту игру. Если вас миновала эта участь, правила игры можно найти на многих Web-узлах.

Программная версия игры работает в режиме моделирования. Один человек начинает игру и указывает количество моделируемых игроков, а затем наблюдает за ходом игры, отслеживая действия своих соперников и выполняя собственные.