

Объектно-  
ориентированный анализ  

---

и проектирование

# Что такое анализ и проектирование?

---

- Этап анализа состоит в исследовании системных требований и проблемы, а не в поисках путей ее решения. Например, при разработке новой информационной системы для автоматизации торговли необходимо описать способы ее использования и основные функции.
- В процессе проектирования основное внимание уделяется концептуальному решению (в виде программного обеспечения или аппаратных средств), обеспечивающему выполнение основных требований, но не вопросам его реализации.

# Требования и акценты первой итерации: основные вопросы ООА/П

---

- На первой итерации стадии развития рассматриваемых примеров основное внимание уделяется фундаментальным вопросам ООА/П и построения объектной системы. Конечно же, для построения программной системы необходимо рассмотреть и множество других вопросов, таких как разработка базы данных, исследование удобства использования и проектирование интерфейса пользователя. Однако эти вопросы выходят за рамки основ ООА/П и применения UML.

# Особенности инкрементальной разработки

---

- Требования к первой итерации являются подмножеством полного набора требований или прецедентов. Например, требования к первой итерации разработки приложения NextGen являются упрощенной версией полного прецедента оформления продажи. Они описывают простой сценарий с оплатой наличными.
- При итеративной разработке сразу не анализируются все требования к разрабатываемой системе. Детально рассматриваются только вопросы, относящиеся к реализации одного прецедента.

Инкрементальная разработка  
одного и того же прецедента в  
течение нескольких итераций

---



# Начальная фаза и стадия развития

---

- Начальная фаза — это короткий шаг к стадии развития. На этой стадии нужно лишь уяснить для себя основные риски, масштаб задачи, ее реалистичность и решить, стоит ли приступать к серьезным исследованиям, т.е. к этапу развития.

# К числу видов деятельности и артефактов начальной фазы относятся следующие:

---

- Краткий семинар по определению требований.
- Выделение основных исполнителей, задач и прецедентов.
- Описание большинства прецедентов в сжатом формате. Полное описание 10-20% прецедентов для более глубокого понимания масштаба и сложности задачи.
- Идентификация наиболее важных требований, связанных с высокими рисками.
- Разработка первой версии документов “Видение” и “Дополнительная спецификация”.
- Составление списка рисков. Например, руководство проекта очень хочет получить демонстрационную версию продукта к началу выставки POSWorld в Гамбурге, т.е. за 18 месяцев. Однако без более глубокого исследования нельзя даже примерно оценить необходимые для этого ресурсы.
- Технические прототипы, обеспечивающие обоснование концепции, а также другие исследования достижимости конкретных требований (корректно ли работает Java Swing с сенсорными экранами?).

# К числу видов деятельности и артефактов начальной фазы относятся следующие:

---

- Разработка прототипа интерфейса пользователя для выявления функциональных требований.
- Формулировка рекомендаций по использованию компонентов на стадии развития (что нужно приобрести, разработать или повторно использовать). Например, может понадобиться приобрести пакет вычисления налоговых платежей.
- Разработка высокоуровневой примерной архитектуры и предложений по использованию компонентов. Это не детальное описание архитектуры, претендующее на корректность или окончательный вариант. Это лишь краткие размышления относительно начальной точки исследований на стадии развития. Например: “Будем использовать клиентское приложение Java без сервера приложений, а в качестве базы данных выберем Oracle...”. На стадии развития эти идеи могут быть поддержаны или отвергнуты.
- Разработка плана первой итерации.
- Составление списка необходимых средств.

# Фаза развития

---

- *Фаза развития* (elaboration) — это первая последовательность итераций, в течение которых решаются следующие задачи:
- Реализуются и тестируются базовые архитектурные элементы.
- Изучается и стабилизируется большая часть требований.
- Обосновываются и устраняются основные риски.

# Основные идеи и рекомендации по реализации стадии развития

---

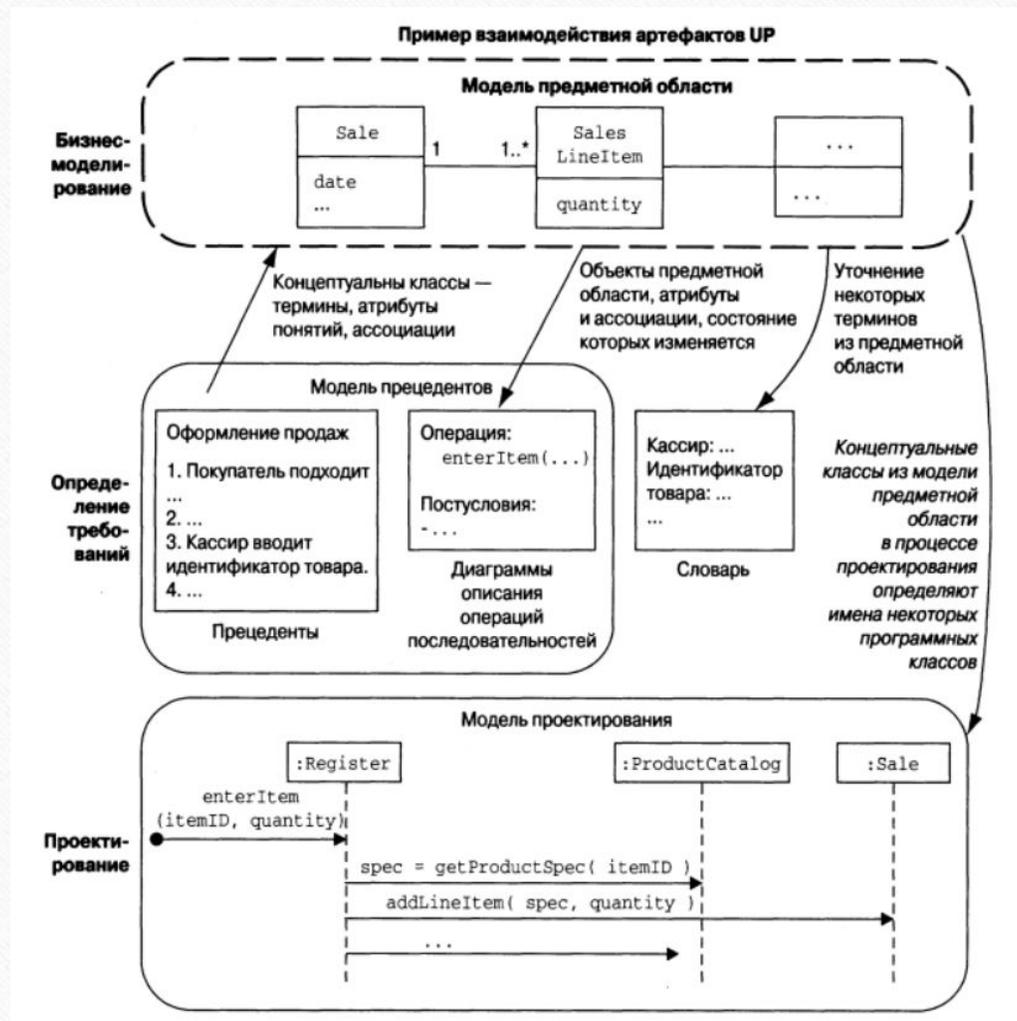
- Планирование кратких итераций, связанных с основными рисками.
- Раннее начало программирования.
- Адаптивное проектирование, реализация и тестирование основных элементов архитектуры.
- Раннее, частое и реалистичное тестирование.
- Адаптация системы на основе обратной связи от специалистов по тестированию, пользователей и разработчиков.
- Подробное описание большинства прецедентов и других требований, проведение серии семинаров, по одному на каждой итерации.

# Модели предметной области

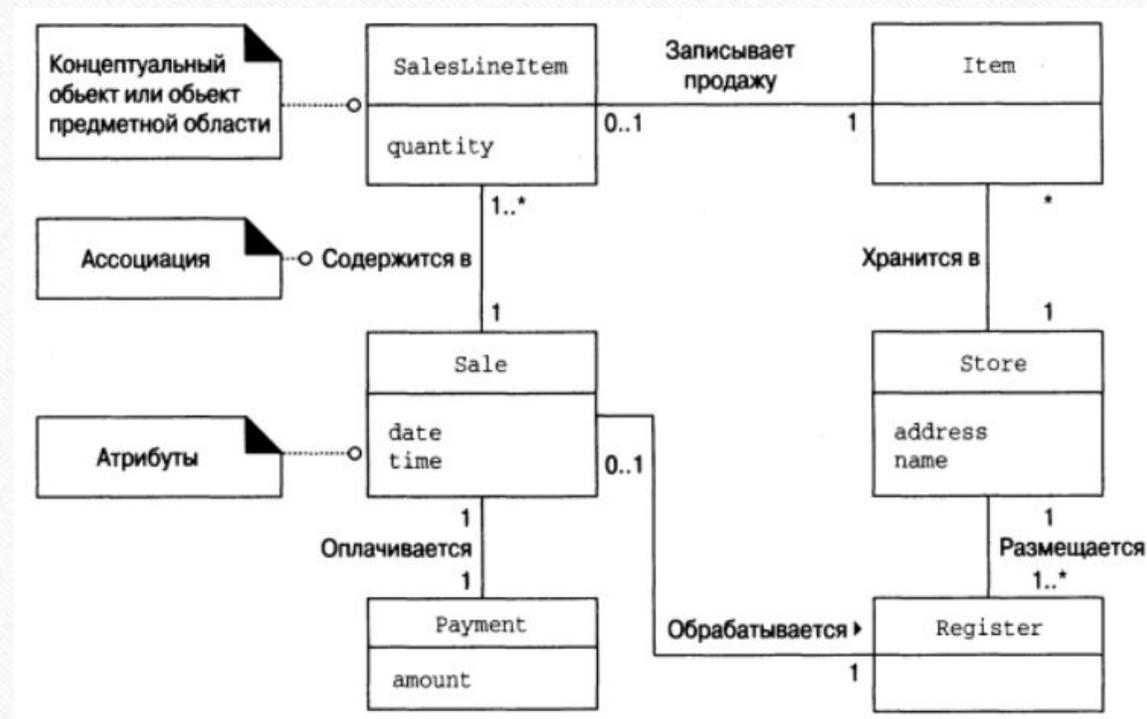
---

- Модель предметной области — это самая важная модель объектно-ориентированного анализа. Она отображает основные (с точки зрения моделирующего) классы понятий (концептуальные классы) предметной области. В этой части описывается процесс объектно-ориентированного анализа и проектирование с помощью системы обозначений UML. Базовые понятия являются тривиальными, но по ходу изложения будут даны полезные рекомендации, касающиеся построения модели, самостоятельное изучение которых может занять недели и месяцы. В этой части приводятся начальные сведения о создании моделей предметной области.

# Модели предметной области



На рис. представлен фрагмент модели предметной области в виде диаграммы классов.



# Что такое модель предметной области

---

- *Модель предметной области* — это визуальное представление концептуальных классов или объектов реального мира в терминах предметной области. Такие модели называют также концептуальными моделями (именно этот термин был принят в первом издании этой книги), моделями объектов предметной области, или объектными моделями анализа.

# Модель предметной области может отображать следующее:

---

- Объекты предметной области или концептуальные классы.
- Ассоциации между концептуальными классами.
- Атрибуты концептуальных классов.

---

Концептуальная модель — это не модель  
программных компонентов



# Концептуальные классы

---

- Модель предметной области иллюстрирует концептуальные классы или словарь предметной области. Неформально: концептуальный класс— это представление идеи или объекта. Если говорить более строго, то концептуальный класс можно рассматривать в терминах символов, содержания и расширения.
- Символы (symbol) — слова или образы, представляющие концептуальный класс.
- Содержание (intension) — определение концептуального класса.
- Расширение (extension) — набор примеров, к которым применим концептуальный класс.

# Модели предметной области и данных

---

- Модель предметной области не является моделью данных (data model), к которой по определению относятся данные, сохраняемые на постоянном носителе, поскольку неизвестно, нужно ли сохранять в базе данных информацию о классах. Концептуальные классы могут вообще не содержать атрибутов и играть в предметной области чисто поведенческую, а не информационную роль.

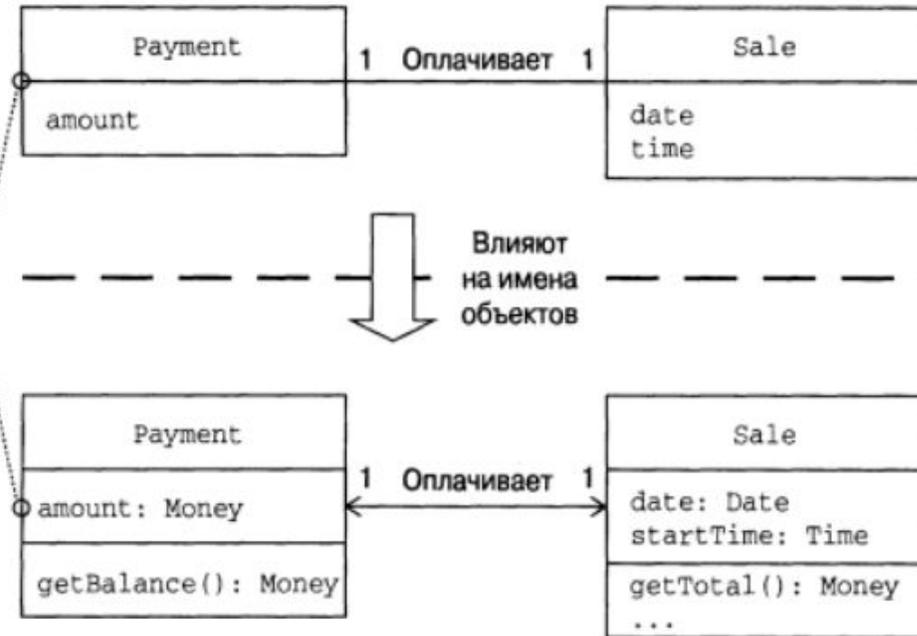
# Модели предметной области и данных

---



## Модель предметной области в рамках UP

Понятия предметной области с точки зрения заинтересованных лиц



## Модель проектирования в рамках UP

При создании программных классов разработчик объектно-ориентированной системы учитывает понятия из предметной области.

Таким образом, удается сократить разрыв между представлением заинтересованных лиц о понятиях предметной области и представлением этих понятий в программном обеспечении

В модели предметной области *Payment* — это понятие, а в модели проектирования *Payment* представляет собой программный класс. Эти сущности отличны друг от друга, однако первая из них оказывает влияние на имя и определение второй.

Такой подход позволяет сократить разрыв между ментальными и программными моделями.

Это одна из наиболее важных идей объектной технологии

# Как создать модель предметной области

---

- Для создания модели предметной области на данной итерации выполняются следующие действия.
- Выделите концептуальные классы (следуя приведенным ниже рекомендациям).
- Отобразите их в модели предметной области в виде классов на диаграмме UML.
- Добавьте необходимые ассоциации и атрибуты.