

Модель предметной области

Понятие, структура и основные принципы

Что было сделано

- Определены требования
 - Анализ требований может включать описание процессов функционирования системы, представленное в форме прецедентов
- Разработаны прецеденты (варианты использования)
 - Важные артефакты этапа анализа требований (они не являются объектно-ориентированными)

Что дальше?

- Требуется разработать **модель предметной области** – определить основные сущности в терминах предметной области
- **Предметная область** – это часть реального мира, которая подлежит изучению с целью автоматизации организации управления.
 - Предметной областью информационной системы является совокупность объектов, свойства которых и отношения между которыми представляют интерес для пользователей информационной системы .
- Модель предметной области отображает основные классы понятий (концептуальные классы) предметной области

Модель предметной области – концептуальная модель

Модель предметной области

- **Модель предметной области** — это система абстракций, которая описывает отдельные аспекты сферы знаний, влияния или деятельности и может быть использована для решения проблем, связанных с этой сферой
- **Модель предметной области** — это представление значимых концепций реального мира, относящихся к материальным аспектам, которые необходимо моделировать в программном обеспечении.
- Изучение предметной области состоит в идентификации *понятий, атрибутов и ассоциаций*, имеющих важное значение для решения задачи

Модель предметной области

- Понятия включают данные, используемые в бизнесе и правила, которые организация применяет в отношении этих компонентов.
- Модель предметной области обычно использует профессиональный словарь.
 - Это позволяет передавать представления заинтересованным сторонам. Он не должен ссылаться на какие-либо технические реализации.

Модель предметной области — это не описание программных компонентов, а представление понятий, выраженных в терминах предметной области задачи (понятий реального мира)

Концептуальная модель — это не модель программных компонентов

- В модели предметной области не используются следующие элементы:
 - Артефакты программирования типа окон или базы данных
 - Обязанности или методы (функции)



Реализация

- Модель предметной области представляется в виде набора диаграмм концептуальных классов, на которых не определяются никакие операции, но может отображать:
 - Объекты предметной области или концептуальные классы
 - Ассоциации между концептуальными классами
 - Атрибуты концептуальных классов
- Информацию на диаграмме (в системе обозначений UML) можно также выразить в виде словесного описания, терминов словаря

Модель предметной области можно рассматривать как визуальный словарь важных абстракций или словарь предметной области

Реализация

Концептуальный класс — это представление идеи или объекта

- 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

Концептуальные ^{[[SEP]]}классы могут вообще не содержать атрибутов и играть в предметной области чисто ^{[[SEP]]}поведенческую, а не информационную роль

Реализация

Пример. Событие **Выполнение покупки**

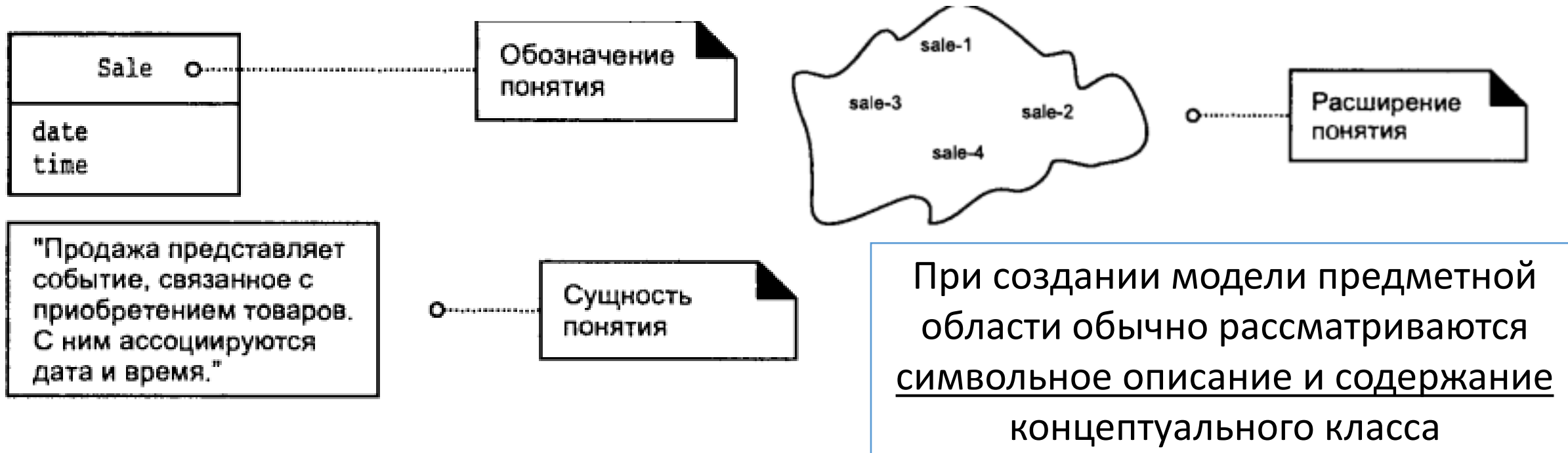
Концептуальный класс для этого события:

Продажа

Символы: Sale

Содержание: представление события *Выполнение покупки* в определенный день и определенное время

Расширения: все примеры покупок (все множество покупок)



Выявление концептуальных классов

Основная задача — идентифицировать концептуальные классы, связанные с разрабатываемым сценарием

- Модель предметной области строится в течение нескольких итераций
- На каждой итерации к модели добавляются концептуальные классы рассматриваемых сценариев, а не делается попытка “объять необъятное” и сразу же построить исчерпывающую модель всех возможных концептуальных классов и их взаимоотношений
- Если на текущей итерации рассматривается упрощенный сценарий, описывающий конкретное действие, то в разрабатываемом фрагменте модели предметной области нужно отобразить понятия, относящиеся только к этому сценарию
- Концептуальные классы без атрибутов вполне допустимы

Стратегии идентификации концептуальных классов

1. С использованием списка категорий концептуальных классов
2. На основе выделения существительных

Категория концептуальных классов	Примеры
Спецификации, элементы проектных решений или описания объектов	ProductSpecification (Спецификация товара), FlightDescription (Описание полета)
Места	store (Магазин), Airport (Аэропорт)
Транзакции	Sale (Продажа), Payment (Платеж), Reservation (Резервирование)
Элементы транзакций	SalesLineItem (Элемент продажи)
Роли людей	Cashier (Кассир), Pilot (Пилот)
Контейнеры других объектов	Store (Магазин), Bin (Бункер), Airplane (Самолет)
Содержимое контейнеров	item (Элемент), Passenger (Пассажир)
Другие компьютеры или электромеханические системы, внешние по отношению к данной системе	CreditPaymentAuthorizationSystem (Система авторизации кредитных платежей), AirTrafficControl (Система управления движением)
Организации	SalesDepartment (Отдел продаж), objectAirline (Авиалинии)
События	Sale (Продажа), Payment (Платеж), Meeting (Встреча), Flight (Полет), Crash (Крушение), Landing (Приземление)
Процессы	sellingAProduct (Продажа продукта), BookingASeat (Бронирование места)
Каталоги	Productcatalog (Каталог товаров), PartsCatalog (Каталог частей)
Записи финансовой, трудовой, юридической и другой деятельности	Receipt (Чек), Ledger (Гроссбух), EmploymentContract (Трудовой контракт), MaintenanceLog (Журнал обслуживания)
Финансовые инструменты и службы	LineOfcredit (Кредитная линия), stock (Акция)
Руководства, документы, статьи, книги	DailyPriceChangeList (Бюллетень ежедневного изменения цен), RepairManual (Руководство по восстановлению)

Стратегии идентификации концептуальных классов

1. С использованием списка категорий концептуальных классов
2. **На основе выделения существительных (лингвистического анализа)**

1. Выделить существительные из текстовых описаний предметной области
2. Отобрать их в качестве кандидатов в концептуальные классы или атрибуты
 - Для реализации подобного подхода удобно использовать развернутые описания прецедентов

Проблема: между существительными и концептуальными классами нет взаимно однозначного соответствия, а слова естественного языка могут иметь по несколько значений

Пример. Лифт

- Предложение в техническом описании:
 - Лифт закрывает дверь, прежде чем двигаться к следующему этажу
The elevator will close its door before it moves to another floor
- Функционально-ориентированный разработчик
 - Извлечет необходимость создания функции "move "
- ОО-разработчик
 - Увидит необходимость создания объектов трех типов: ELEVATOR, DOOR, FLOOR, приводящих к классам

Проблема: Но необходим ли класс DOOR?

Может быть, да, а может быть и нет

Свойство класса ELEVATOR?

Отдельный класс?

Решение проблемы: абстрактные типы данных (АТД)

- Абстрактный тип данных (АТД):
 - математическая модель и операции, определенные в рамках этой модели

Для решения проблемы выбора (пример про дверь) вот вопрос, на который следует ответить:

Является ли " DOOR " независимым типом данных с собственными четко определенными операциями или все они уже включены в операции других типов данных, таких как, например, ELEVATOR?

Последовательность создания модели

Для создания модели предметной области выполните следующие действия:

1. Составьте список кандидатов на роль концептуальных классов на основе списка категорий и метода анализа текстового описания для текущей итерации разработки
2. Отобразите их в модели предметной области
3. Добавьте необходимые ассоциации, отражающие связи, для которых требуется выделение памяти
4. Добавьте атрибуты, необходимые для выполнения информационных требований

Выявление класса – это двойственный процесс: генерирование кандидатов, их отбраковка

Типичная ошибка при создании модели предметной области

Отнесение некоторого объекта к атрибутам, в то время как он должен относиться к концептуальным классам

Чтобы избежать этой ошибки, следует придерживаться правила:

Если некоторый объект X в реальном мире не является числом или текстом, значит, это скорее концептуальный класс, чем атрибут

Пример. Является ли store (магазин) атрибутом объекта Sale (Продажа) или отдельным концептуальным классом Store?

- В реальном мире магазин не является числом или текстом, он представляет реальную сущность, организацию, занимающую некоторое место. Следовательно, Store нужно рассматривать в качестве концептуального класса.

Типичная ошибка при создании модели предметной области

Проектирование класса, которого нет – называя классом то, что в действительности является функцией (подпрограммой)

В одном классе смешиваются две или более абстракций

Чтобы избежать этих ошибок, следует придерживаться правила:

Следует убедиться, что каждый класс соответствует осмысленной абстракции данных

Рекомендации при обосновании типа

- Цель анализа системы не в том, чтобы "моделировать мир"
- Задачей создателей ПО является моделирование мира лишь в той мере, которая касается создаваемого ПО.
- Подход АТД и ОО-метода основан на том, что объекты определяются только тем, что можно с ними делать.
- Если все операции и свойства некоторого типа не связаны с целями системы или покрываются другими классами, то сам тип не должен рассматриваться как самостоятельный класс.

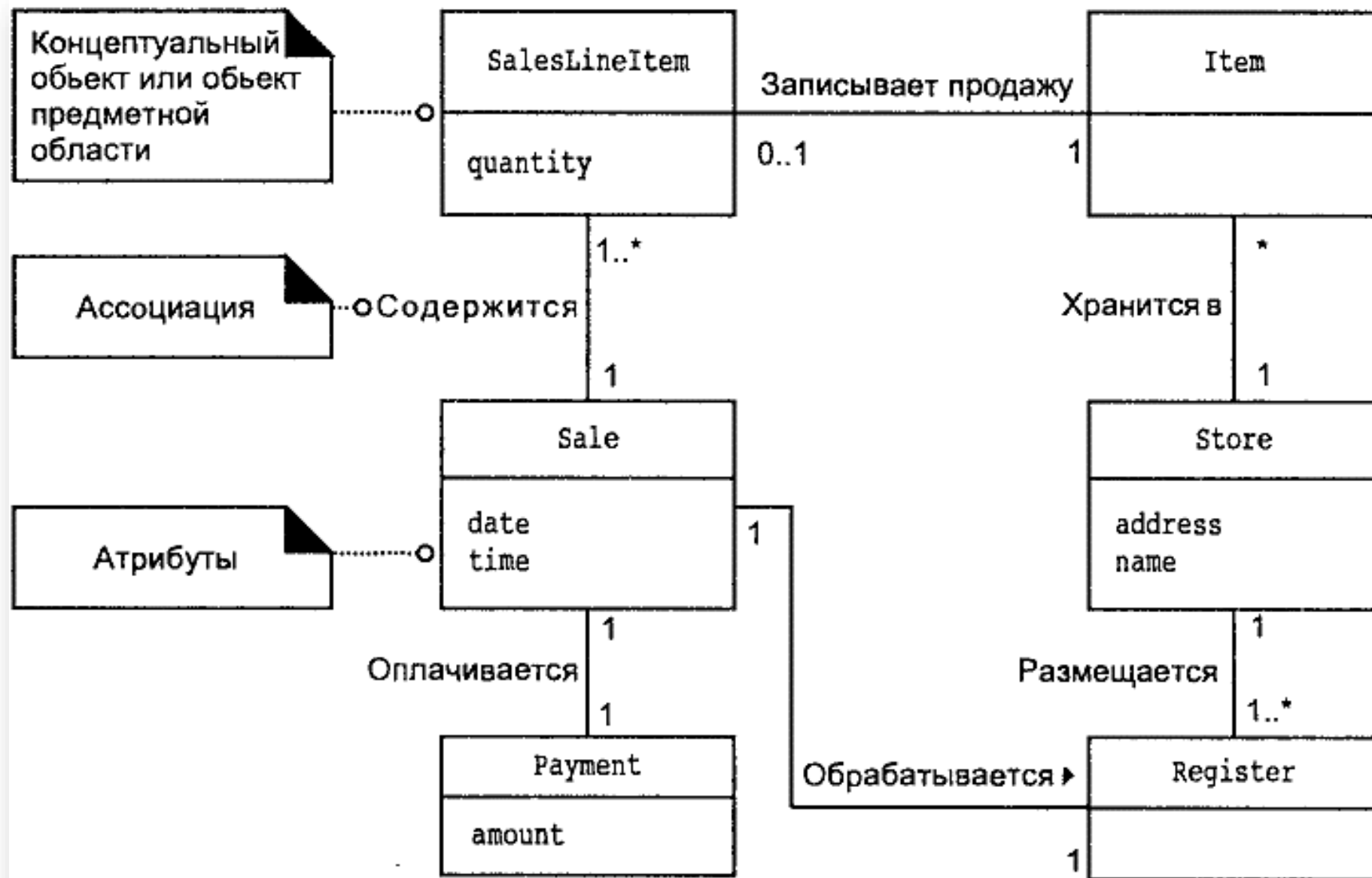
Пример. Лифт. Проблема выявления класса.

- Документ с требованиями к лифтовой системе содержит предложение:
 - Запись базы данных должна создаваться всякий раз, когда лифт перемещается от одного этажа к другому (A database record must be created every time the elevator moves from one floor to another).
- Существительное "record" предполагает класс DATABASE_RECORD
- Вопрос. Абстракция данных: понятие move, определяющее перемещение между этажами?
 - Из смысла данного предложения скорее следует необходимость класса MOVE
- Другой вариант того же требования:
 - Каждое перемещение лифта приводит к созданию записи в базе данных (A database record must be created for every move of the elevator from one floor to another).
- Из этого видно, что "move" из глагола переходит в разряд существительных, претендуя на класс

Фрагмент модели предметной области. Пример

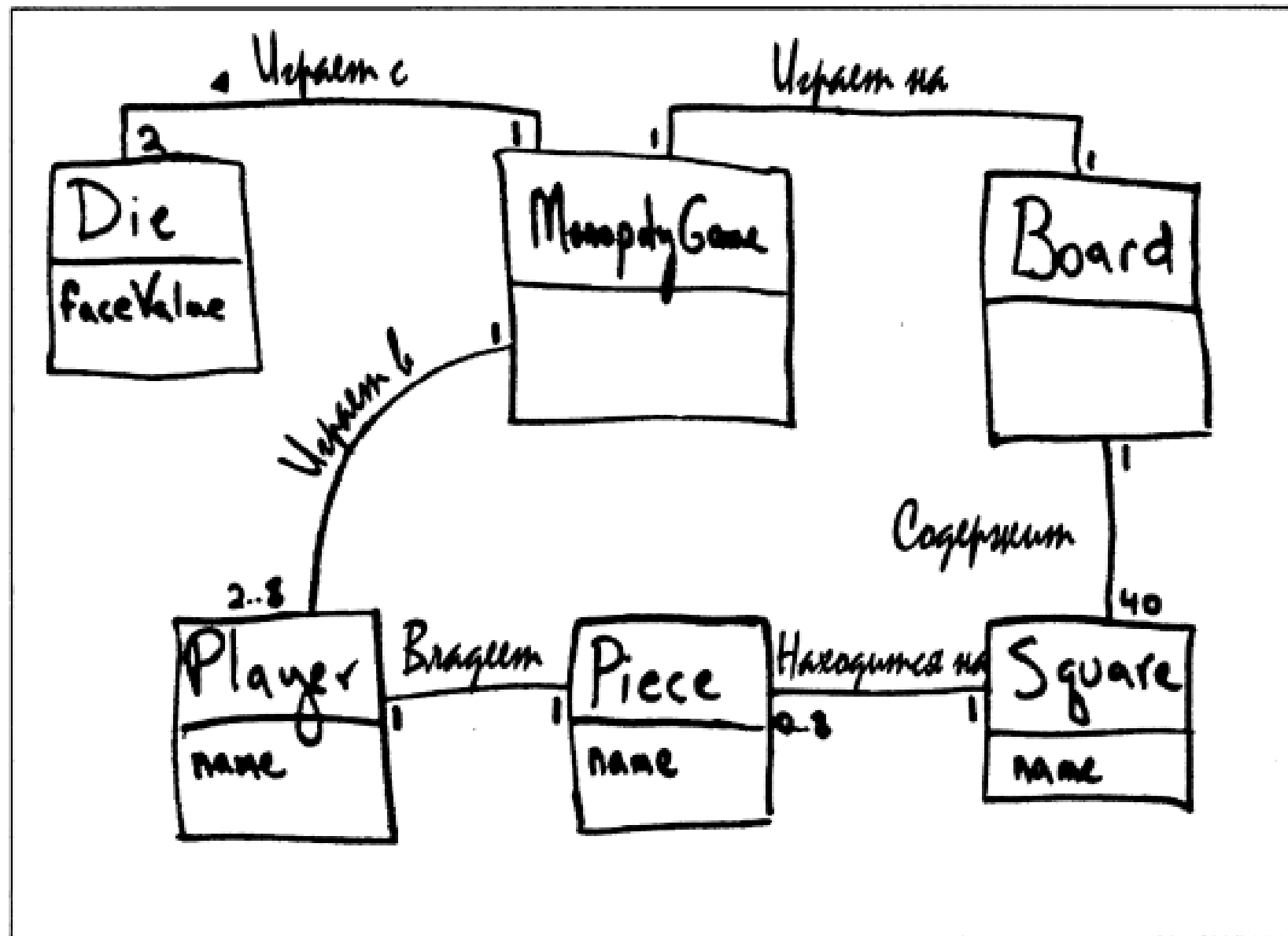
Концептуальные классы

- SalesLineItem (Элемент продажи)
- Item (Элемент)
- Sale (Продажа)
- Payment (Платеж)
- Store (Магазин)
- Register (Реестр)



Модель предметной области игры «Монополия». Пример

Концептуальные
классы



Что дальше?

- Требуется добавить **методы программных классов**, описывающие передачу сообщений между объектами для удовлетворения требованиям
 - ✓ Вопрос определения способов взаимодействия объектов и принадлежности методов важен и не тривиален
- Разработать диаграммы взаимодействий
- Принципы объектного проектирования отражены в шаблонах проектирования GRASP – шаблонах распределения обязанностей

Модель проектирования

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

