

# Сравнение реляционной и объектной моделей данных

## Структурные модели реляционной и объектно-ориентированной БД

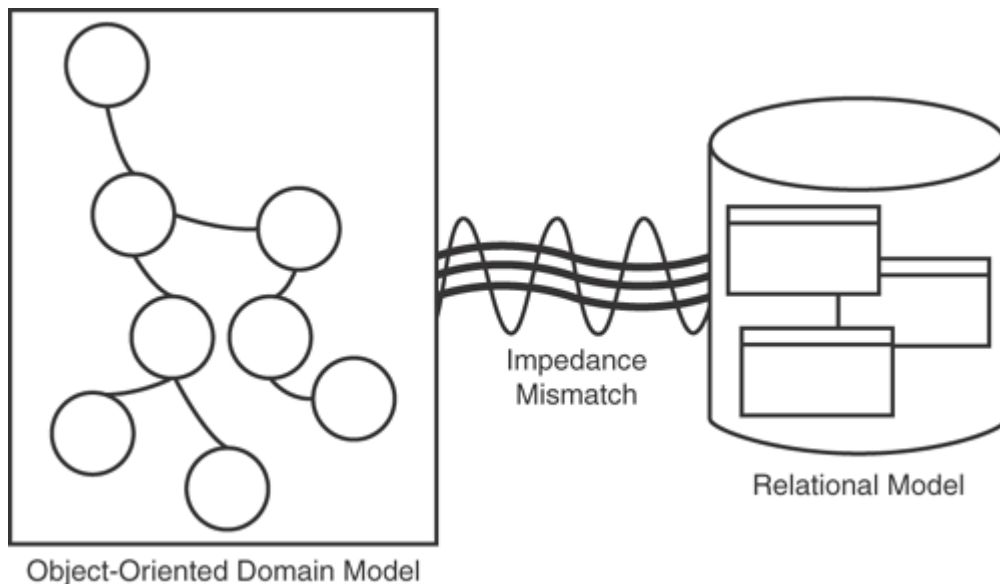


# Структурный аспект

- Объектная МД способна более адекватно реализовывать модель предметной области
  - > Модель «сущность-связь» - концептуальный уровень
  - > Фундаментальные абстракции, лежащие в основе моделей «сущность-связь» используются и в объектно-ориентированном подходе
    - > Агрегация, группирование, специализация/обобщение
- Объектная МД в большей степени соответствует ООЯП => легче разрабатывать приложения
- На структурном уровне можно осуществлять **взаимное преобразование** объектных и реляционных МД
  - > Object-Relational Mapping (ORM)

# Манипуляционный аспект

- Реляционная МД:
  - > Реляционная алгебра и реляционное исчисление
    - > Эквивалентны, математически строгим
    - > SQL комбинирует эти механизмы
  - > **Impedance Mismatch** — сочетание ООЯП и декларативного языка запросов, ориентированного на работу с множествами



# Манипуляционный аспект

- Объектная МД:
  - > 5 стандартных операций над объектами
    - > Конструктор, деструктор, модификатор, селектор, итератор
  - > Математический аппарат **отсутствует**
  - > Преобладает программная навигация по объектам
    - > Обращение по ссылке (traverse) – наиболее частая операция в ООП
    - > Явное кодирование запросов любой сложности
    - > **Сопровождаемость ухудшается**
    - > **Производительность ?**

# Аспект обеспечения целостности

- Реляционная МД:
  - > семантическая целостность значений атрибутов
  - > целостность сущностей
  - > ссылочная целостность
- Объектная МД:
  - > OID – отсутствие в БД идентичных объектов
  - > Целостность сущностей, свойств и ссылок можно поддерживать только **реализацией** соответствующего поведения

# **Устойчивость и модель персистентности**

# Устойчивость объекта

- Способность объекта существовать **во времени**, переживая породивший его процесс
- и/или **в пространстве**, перемещаясь из своего первоначального адресного пространства
  - > Группы объектов по продолжительности существования:
    - > промежуточные результаты вычисления выражений;
    - > локальные переменные;
    - > глобальные переменные и динамически создаваемые данные;
    - > данные, сохраняющиеся между сеансами выполнения программы;
    - > данные, сохраняемые при переходе на новую версию программы;
    - > данные, которые переживают программу

СУБД

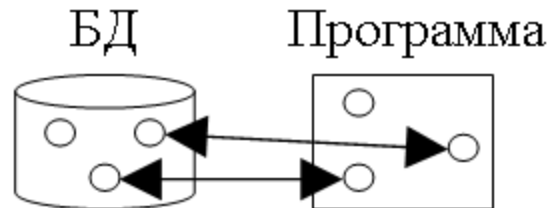


# Персистентность

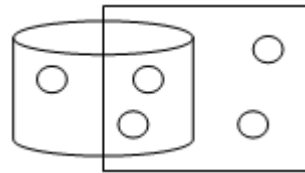
- Способность программного обеспечения создавать и поддерживать устойчивые объекты
- **Модель персистентности** – совокупность методов, обеспечивающих манипуляции с объектами, хранимыми в БД
  - > Указание устойчивости объекта
  - > Доступ к устойчивым объектам
- Идеальная модель персистентности
  - > ортогональная (к типам)
  - > прозрачная
  - > независимая (от внешней памяти)
- Бесшовная модель персистентности

# Указание устойчивости объекта

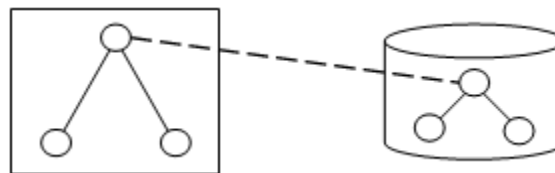
## 1) Объявление схемы базы данных



## 2) Особый механизм инициализации устойчивых объектов



## 3) Достижимость по ссылке из другого устойчивого объекта



# Доступ к устойчивым объектам

- 1) Получить ссылку на устойчивый объект и послать сообщение этому объекту для вызова операции, выполняющейся в пространстве устойчивых объектов (в процессе СУБД)
- 2) Явно считать состояние устойчивого объекта во временный объект, выполнить операции с временным объектом, обновить устойчивый объект
- 3) Прозрачный доступ по ссылке
  - Степень прозрачности зависит от способа реализации
    - указателей в приложениях и
    - объектных идентификаторов в СУБД

# Доступ по ссылке к устойчивым объектам

- В БД и приложениях используются различные форматы указателей
  - > В ООЯП – адреса виртуальной памяти
  - > В БД – генерируемый СУБД идентификатор объекта
  - > Необходимо преобразование **адрес** ↔ **OID**
- Активация / деактивация
- Абстрактные ссылки
  - > Если указываемый объект неактивен,
  - > то при разыменовании абстрактной ссылки он автоматически активизируется,
  - > информации для нахождения объекта на диске хранится в самой абстрактной ссылке
- Прямые ссылки

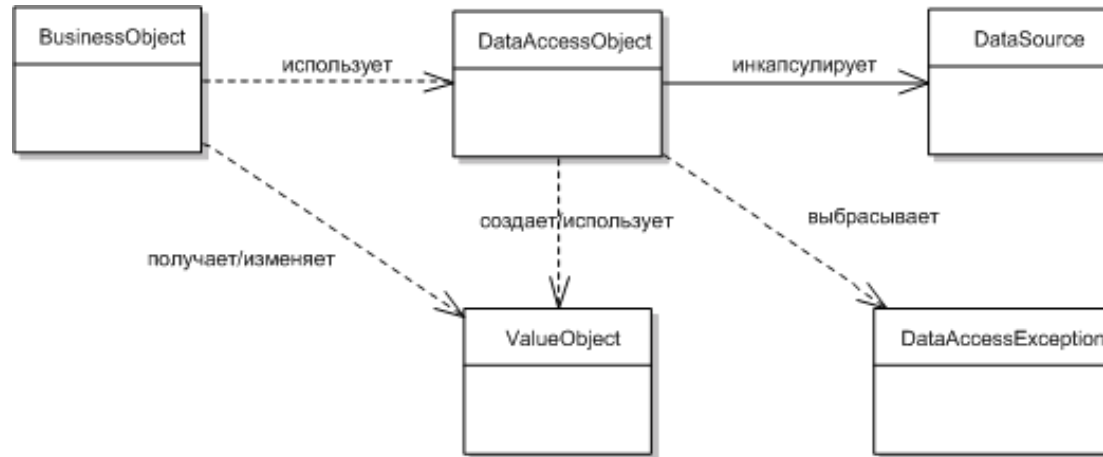
# Средства объектно- реляционного преобразования

# Сложность объектно-реляционного преобразования

- Преобразование отношений **ассоциации** и **наследования** между объектами
- в связи между таблицами, построенные на **внешних ключах**
- **Пример:** преобразование ассоциации объектов «один-ко-многим» в реляционное представление
  - > «Один» объект содержит коллекцию ссылок на «многие» объекты
  - > В РСУБД кортежи отношения «многие» включают внешний ключ на отношение «один»

# Стратегии обеспечения устойчивости объектов в РСУБД

- Шаблон DAO



- Разработать собственный слой персистентности
  - > **Риск:** затраты на поддержку кода устойчивости объектов могут превысить затраты на бизнес-логику
- Использовать стороннее средство объектно-реляционного преобразования (СОРП)

# Средства объектно-реляционного преобразования

- Состав СОРП:
  - > Библиотека времени выполнения
    - > Выполняет прямое и обратное объектно-реляционное преобразование
  - > Утилита времени разработки
    - > Для моделирования устойчивых объектов приложения и отношений между ними
    - > Выполняет объектно-реляционное преобразование на уровне схемы данных
    - > Результатом работы – карта преобразования, обычно хранится в XML-файле



# Средства объектно-реляционного преобразования

- Модель персистентности
  - > Библиотека времени выполнения создает **прокси-объекты**
    - > Объекты-заместители в оперативной памяти для объектов, находящихся в БД
    - > Классы для этих объектов могут создаваться «на лету» по карте преобразования
  - > Указание устойчивости объектов
    - > Объявление схемы данных (обязательно)
    - > + Особый механизм инициализации и/или достижимость по ссылке
  - > Доступ к устойчивым объектам
    - > При помощи абстрактных ссылок

# Средства объектно-реляционного преобразования

- API для объектно-реляционного преобразования
  - > ODMG API for Java
  - > EJB Entity Beans
  - > Java Data Objects (JDO)
  - > Java Persistence API (JPA)
  - > Собственный
- Примеры СОРП
  - > Oracle TopLink, Hibernate (NHibernate), Apache JPOX, Apache OpenJPA, EclipseLink, ...