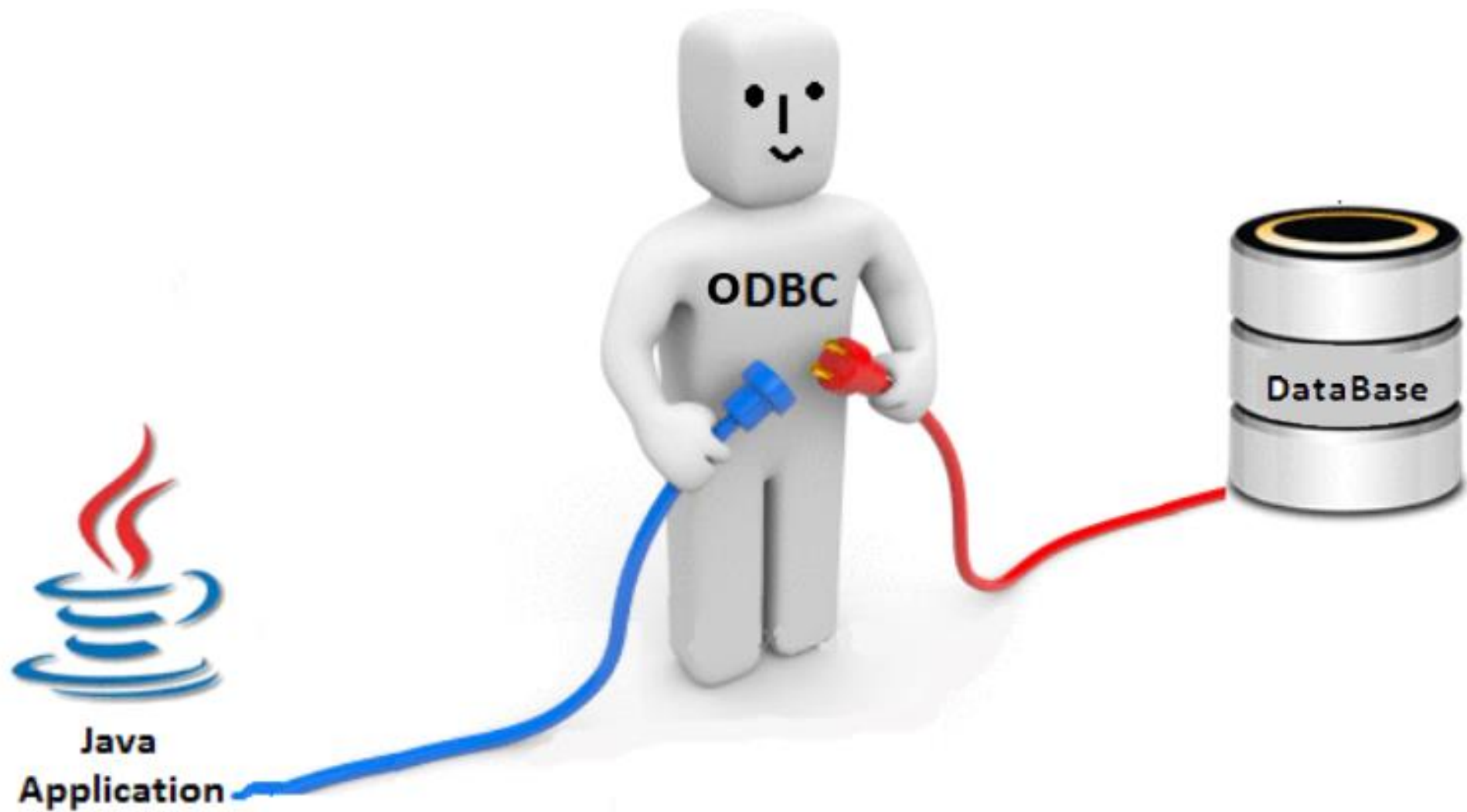
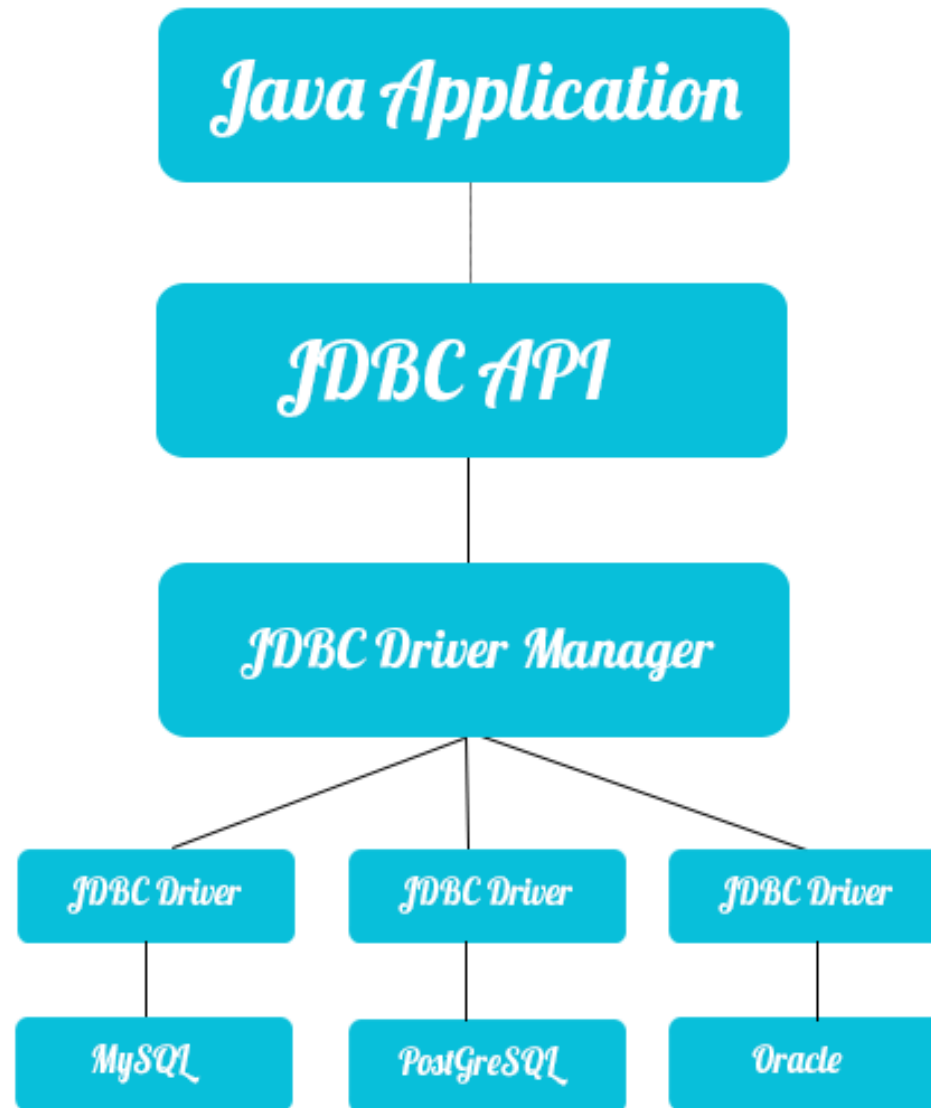


Lecture 3. Part 1. JDBC

Идея ODBC



Что такое JDBC?



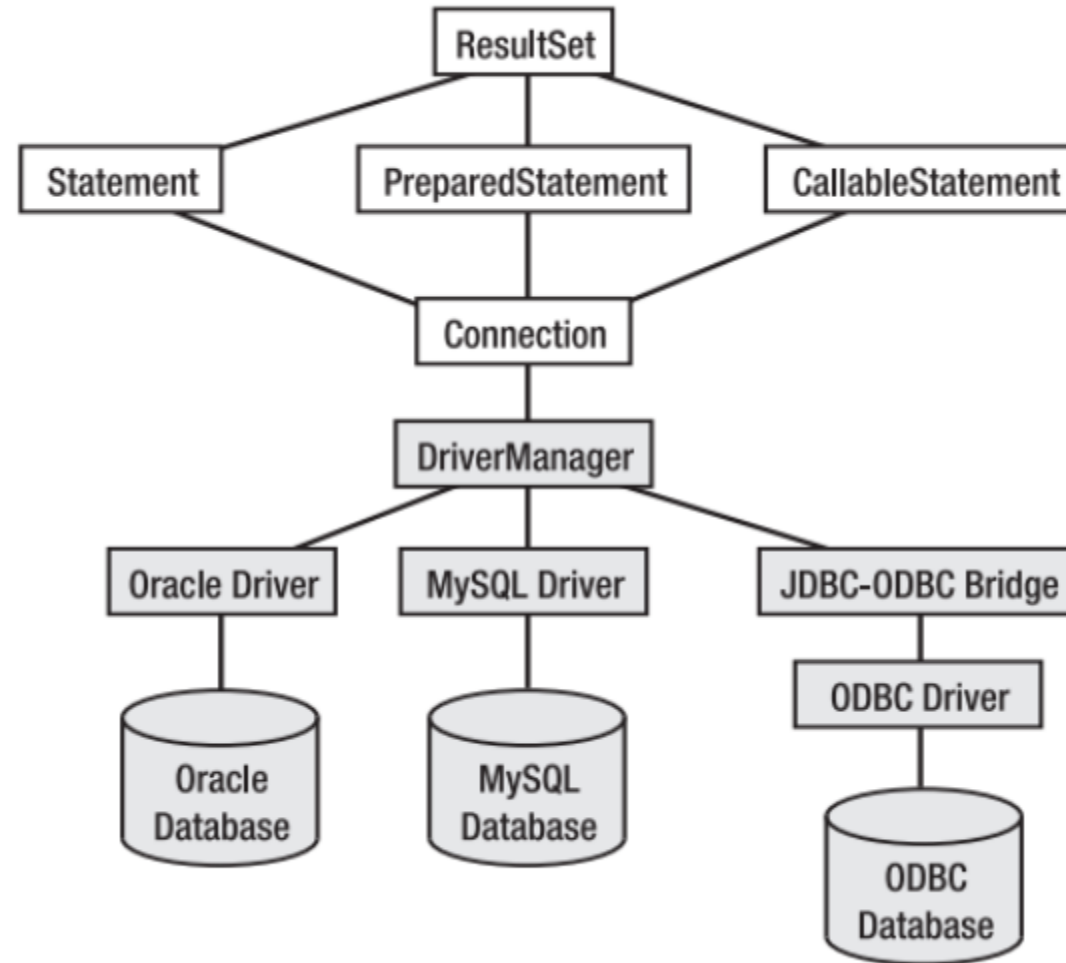
JDBC API



Типы драйверов

1. Драйвер, использующий другой прикладной интерфейс взаимодействия с СУБД, в частности, ODBC (так называемый JDBC-ODBC — мост).
2. Драйвер, работающий через внешние native библиотеки клиента СУБД.
3. Драйвер, работающий по сетевому и независимому от СУБД протоколу с промежуточным Java-сервером, который, в свою очередь, подключается к нужной СУБД.
4. Сетевой драйвер, работающий напрямую с нужной СУБД и не требующий установки native-библиотек.

Java application с использованием JDBC

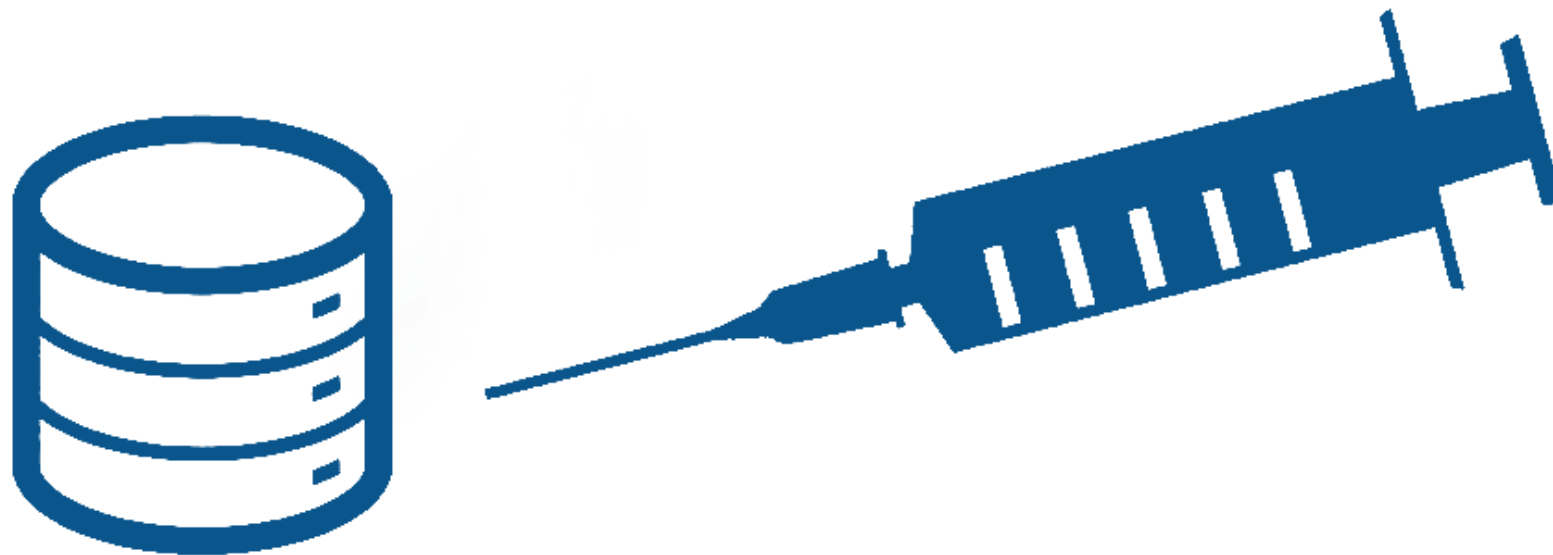


Преимущества

- Лёгкость разработки;
- Код практически не меняется;
- Не нужно устанавливать громоздкую клиентскую программу;
- К любой базе можно подсоединиться через легко описываемый URL.

Пример использования JDBC

SQL инъекция



SQL Injection

Транзакции

- Атомарность;
- Согласованность;
- Изолированность;
- Долговечность.

Уровни изоляции транзакций

- TRANSACTION_NONE;
- TRANSACTION_READ_UNCOMMITTED;
- TRANSACTION_READ_COMMITTED;
- TRANSACTION_REPEATABLE_READ;
- TRANSACTION_SERIALIZABLE.

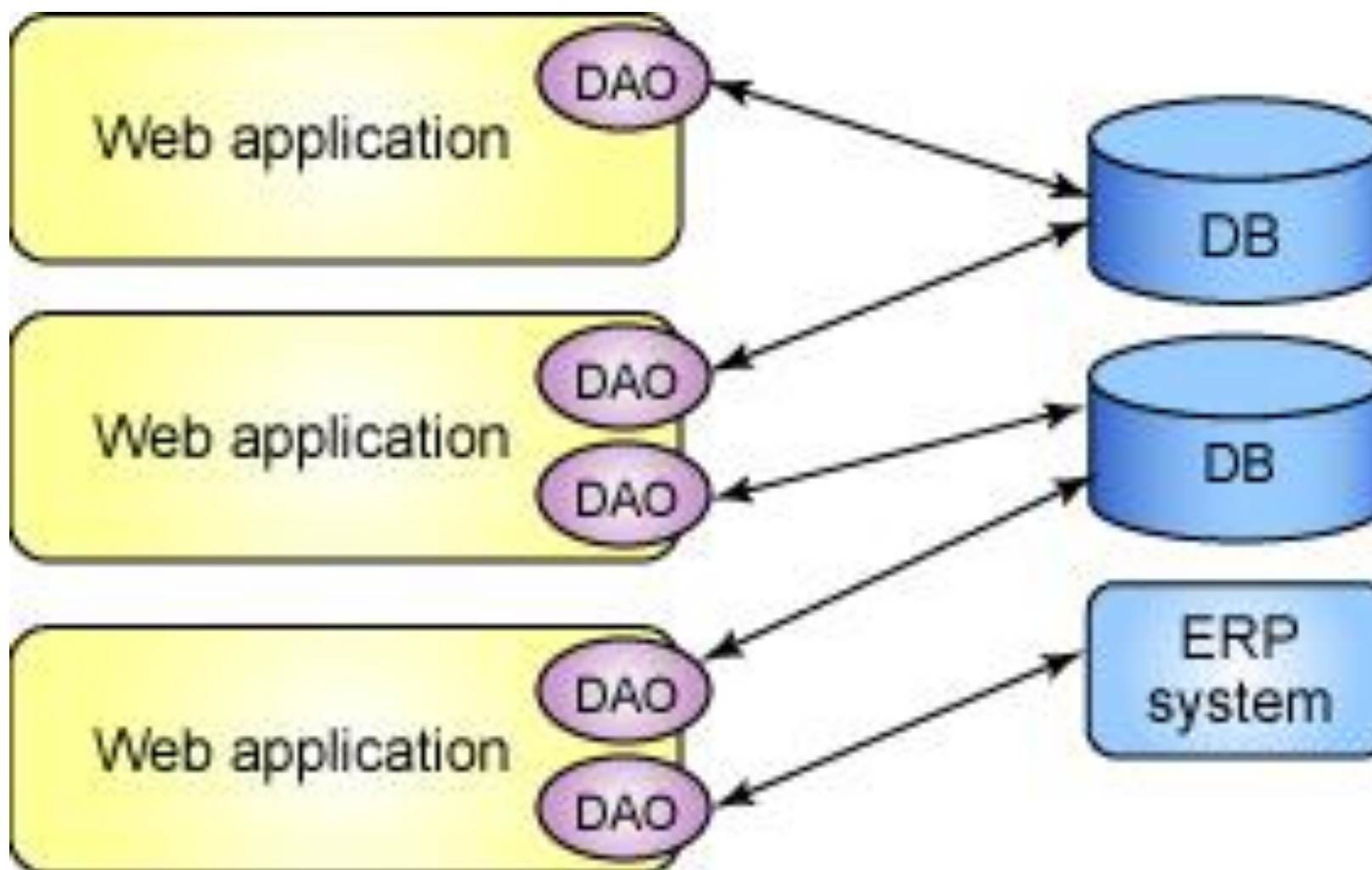
Литература

1. <https://goo.gl/ZDkxRo> (Работа с базами данных с помощью JDBC драйвера)
2. <http://www.javenue.info/post/java-jdbc-api> (JDBC API в Java - обзор и tutorial)
3. <https://o7planning.org/ru/10167/java-jdbc-tutorial> (Руководство Java JDBC)
4. <https://habr.com/post/178405/> (Правильно освобождаем ресурсы в Java)
5. <https://habr.com/post/130826/> (SQL инъекции. Проверка, взлом, защита)
6. <https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/PT-devteev-Advanced-SQL-Injection.pdf> (Advanced SQL injection)

Q&A

Lecture 3. Part 2. DAO & DTO

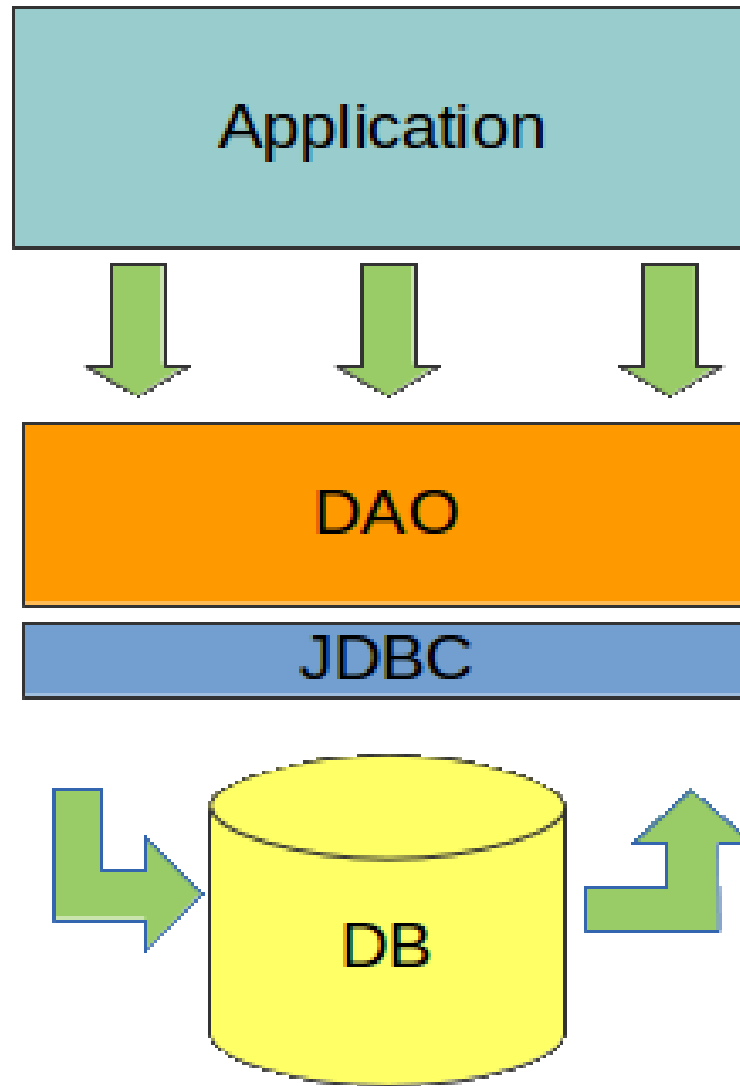
Проблема



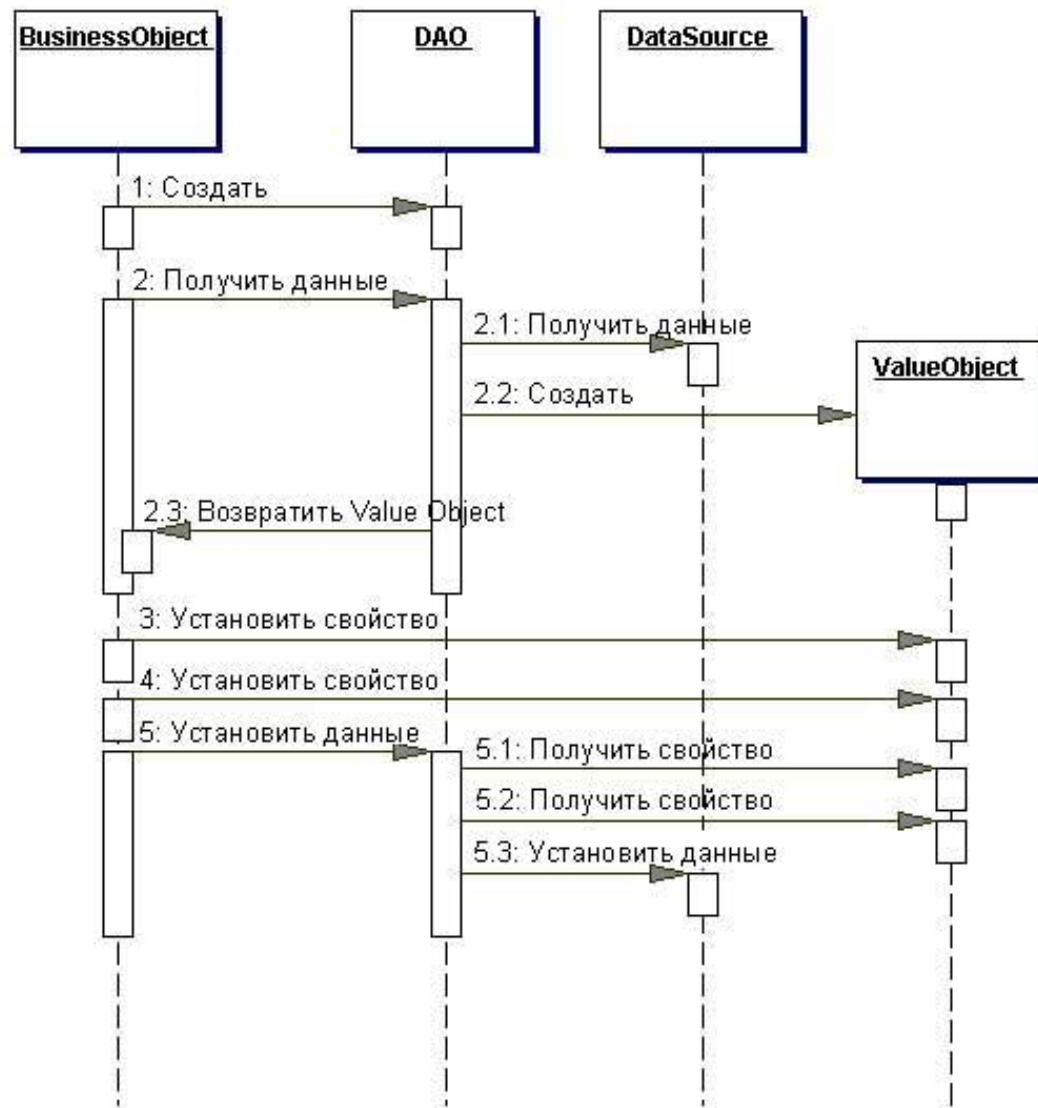
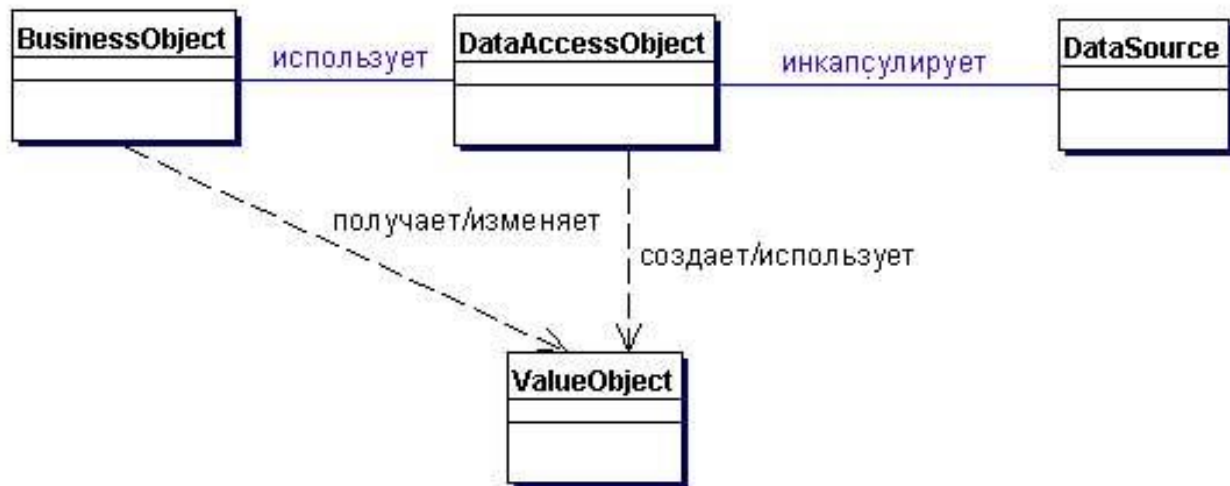
Ограничения

- Компоненты управления данными с управляемой компонентом персистенцией должны получать и сохранять информацию в персистентных хранилищах и других источниках данных.
- API доступа к персистентному хранилищу данных может зависеть от поставщика продукта. Другие источники данных могут иметь нестандартные или патентованные API.
- Для извлечения или сохранения данных во внешних и/или традиционных системах компоненты обычно используют патентованные API.
- Включение в компоненты специфических механизмов доступа и API прямо влияет на переносимость компонентов.
- Компоненты должны быть прозрачны для реальной реализации персистентного хранилища или источника данных и обеспечивать легкую миграцию на продукт другого поставщика, на другой тип хранилища и на другой тип источника данных.

Data Access Object Pattern



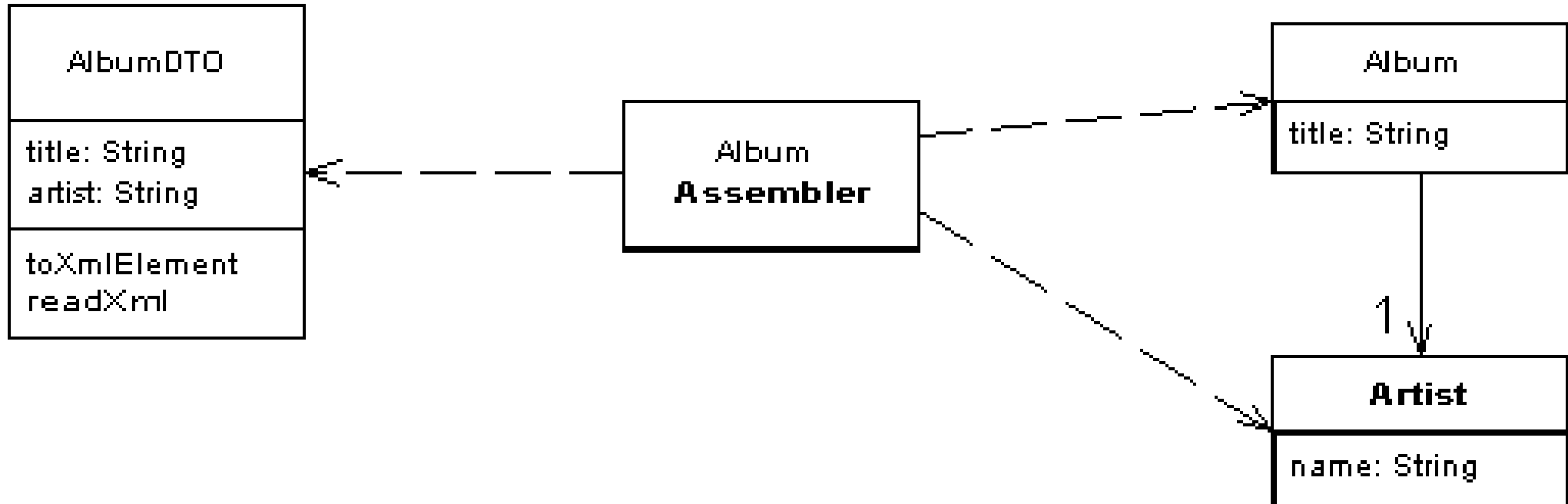
Структура



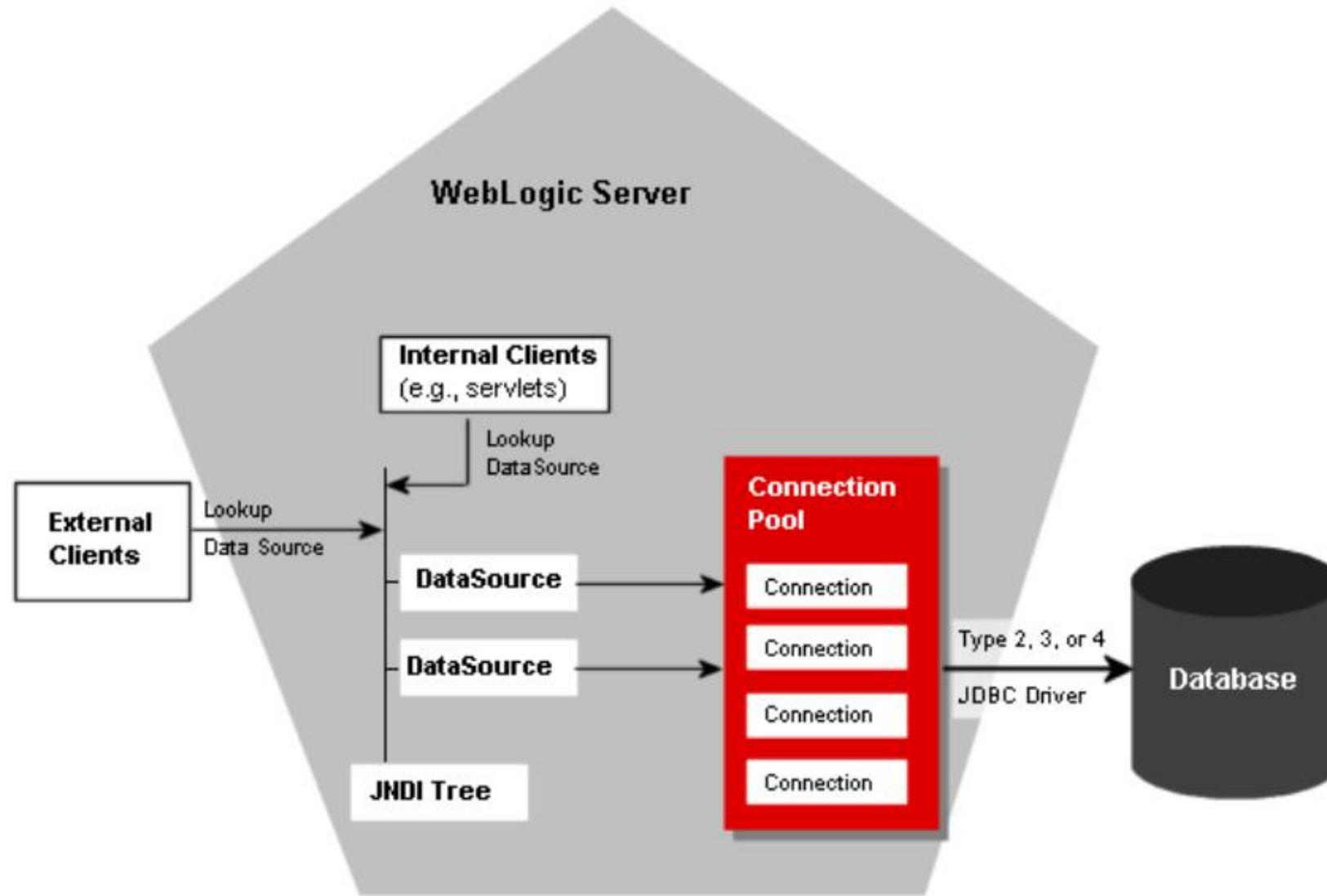
Использование DAO

- Разрешает прозрачность
- Облегчает миграцию
- Уменьшает сложность кода в бизнес-объектах
- Централизует весь доступ к данным в отдельном уровне
- Бесполезно для управляемой контейнером персистенции
- Добавляет дополнительный уровень
- Требуется разработки иерархии классов

Data Transfer Object



Connection Pool



Пример использования DAO / DTO

Литература

1. <https://goo.gl/ZDkxRo> (Работа с базами данных с помощью JDBC драйвера)
2. <https://www.dokwork.ru/2014/02/daotalk.html> (Java. Реализация шаблона DAO)
3. <https://www.ibm.com/developerworks/ru/library/j-genericdao/index.html> (Не повторяйте DAO)

Q&A

Thank You

