

JDBC & Hibernate

Введение в Hibernate

JDBC & Hibernate

Автор курса



Бондаренко Андрей

JDBC & Hibernate

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Введение в Hibernate

What is ORM

Что такое ORM?

ORM(Object Relation Mapping) – технология, которая работает с таблицами, как с объектами, которая напрямую к БД не обращается (хоть и это возможно).



What is JPA

Что такое JPA?

JPA (Java Persistence API) – спецификация, документ, в котором описаны правила и API для реализации принципов ORM для Java (аннотации, настройки, подход).

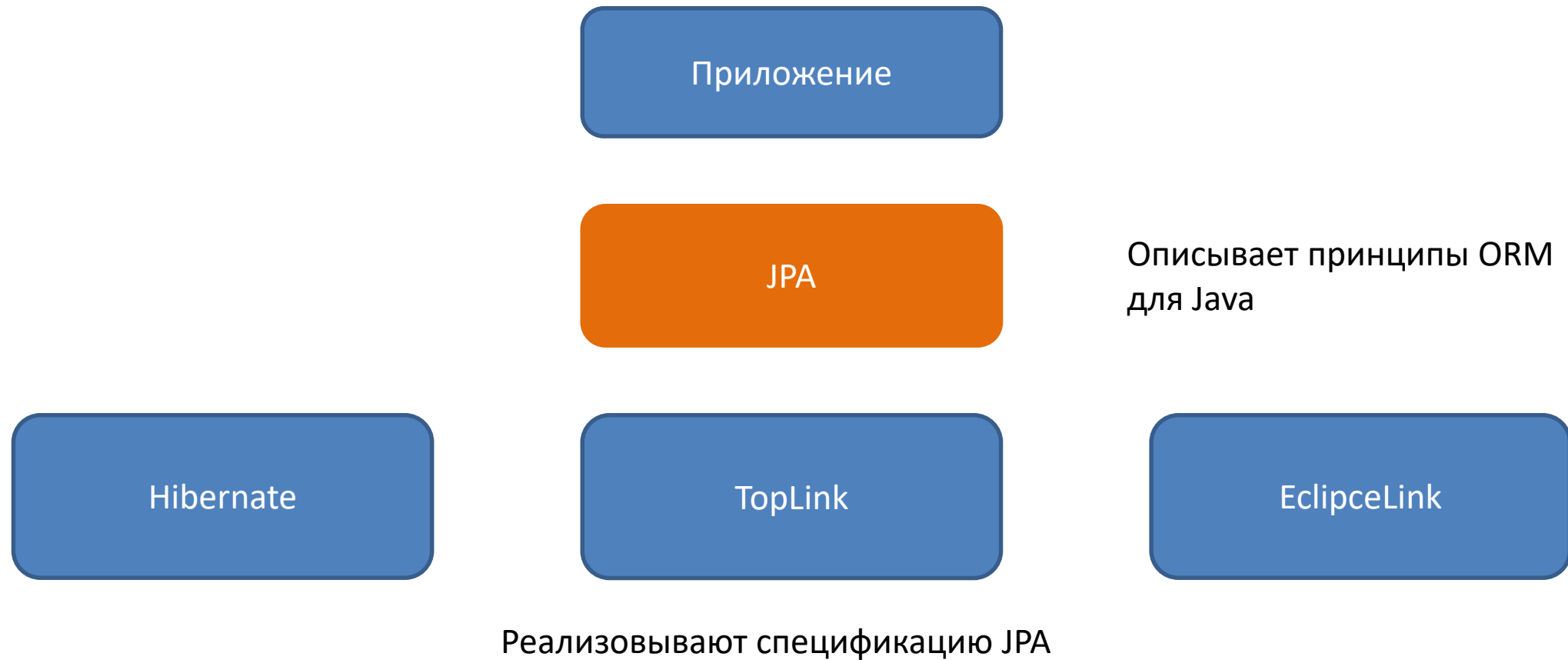
JPA имеет множество реализаций:

- OpenJPA (Apache)
- Hibernate
- TopLink (Oracle)
- EclipseLink

Часто используется запись JPA API.

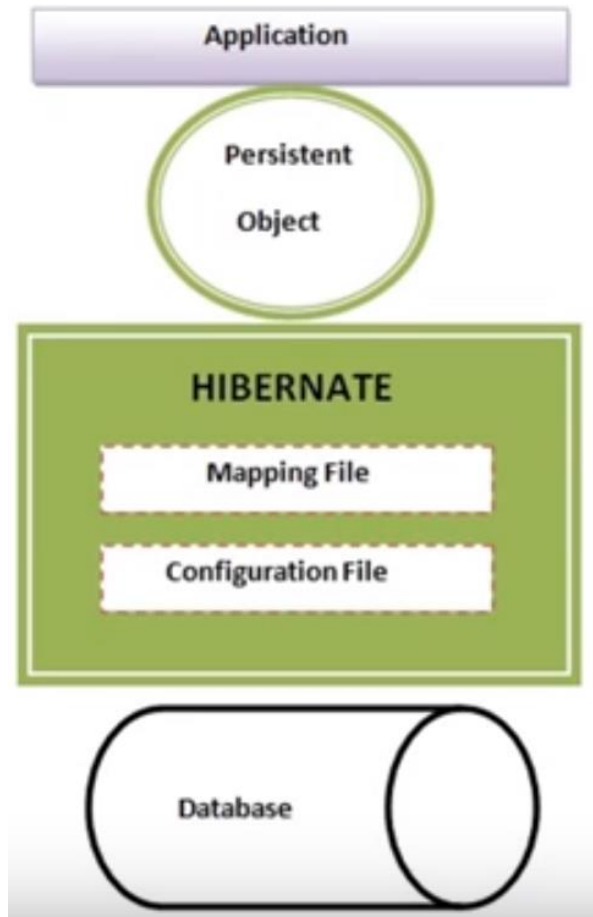
ORM

Приложение с использованием ORM



Hibernate

Hibernate



Hibernate – фреймворк между БД и приложением, в котором не нужно создавать прямые SQL запросы через JDBC.

Может применяться в любом типе приложения (web/desktop/Spring и т.д.).

Основная цель – работа с таблицами БД как с объектами.

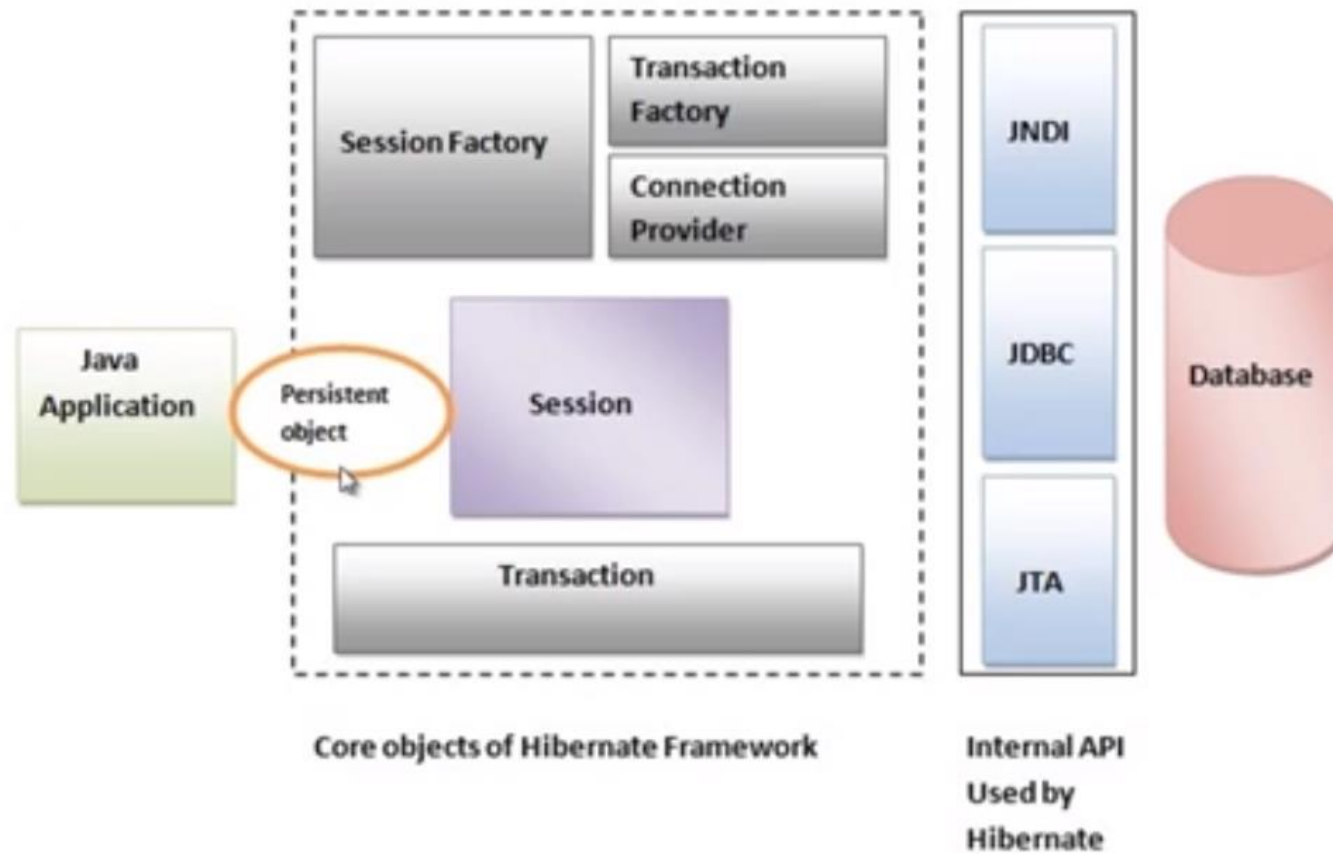
Легче создавать правильный абстрактный уровень для приложения (применять принципы ООП).

Hibernate & JDBC

- На низком уровне Hibernate выполняет запросы через JDBC
- JDBC обязательно нужно знать

Hibernate scheme

Расширенная схема



Advantages

Достоинства Hibernate

- Более высокоуровневый (ООП-ориентированный) код без ручной работы JDBC API
- Переносимость (перенос на другую БД в основном требует небольших изменений).
- Много примеров и документации.
- Работа со всеми популярными СУБД.
- Работа с NoSQL решениями.
- Интеграция с фреймворками, IDE
- Удобство (для тех, кто разобрался с основами)
- Можно не углубляться в SQL
- Множество настроек (транзакции, кеширование, генерация таблиц и пр.).
- Обновления, улучшения.

Disadvantages

Недостатки Hibernate

- В крупных проектах возможна потеря производительности (тяжеловесность), при неправильных настройках.
- Дополнительно нужно изучать API для Hibernate и JPA.
- Сложность для новичков, множество настроек, специфики, аннотаций.
- Меньше контроля (доступ к БД на низком уровне).
- Привязка к своему API (все меньше и меньше).

Hibernate

Hibernate API

- В ранних версиях Hibernate имел сильную привязку к своему API из-за чего переход на другую JPA реализации переходил не так быстро.
- В данный момент с каждой версией происходит уклон в сторону JPA API.

Configuration

Hibernate Configuration

Для работы с Hibernate необходимо создать файл конфигурации hibernate.cfg.xml

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE hibernate-configuration SYSTEM
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>

        <property name = "hibernate.dialect"> org.hibernate.dialect.MySQLDialect</property>
        <property name = "hibernate.connection.driver_class">com.mysql.jdbc.Driver </property>
        <property name =
            "hibernate.connection.url">jdbc:mysql://localhost:3306/carsshop</property>
        <property name = "hibernate.connection.username">root</property>
        <property name = "hibernate.connection.password">root</property>
        <mapping class="ex_001_hibernate_configuration.ClientsDetails" />

    </session-factory>

</hibernate-configuration>
```

Entity

Hibernate Entity with Annotations

Описываем сущность таблицы через класс

```
@Entity
public class ClientsDetails {
    @Id
    private int id;
    private String name;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

Inserting operation

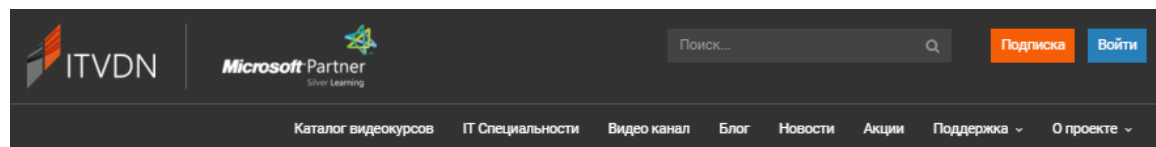
Insert or create table

Работаем с таблицами как с объектами

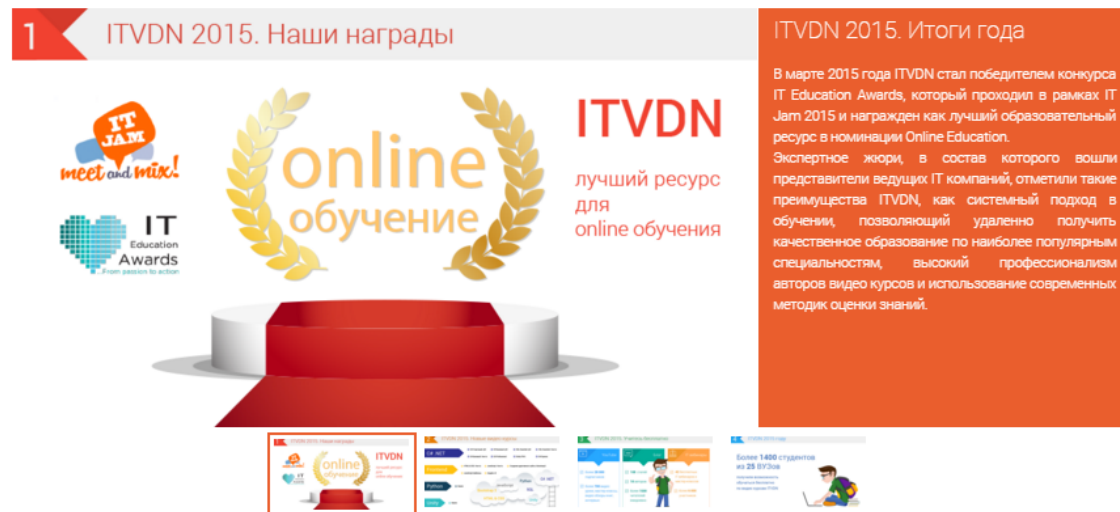
```
public class Main {  
    public static void main(String[] args) {  
        ClientsDetails details = new ClientsDetails();  
        details.setAge(20);  
        details.setId(1);  
        details.setName("Vasiliy");  
        details.setPhone("+380950750703");  
  
        SessionFactory factory = new Configuration()  
            .configure()  
            .buildSessionFactory();  
  
        Session session = factory.openSession();  
        session.beginTransaction();  
        session.save(details);  
        session.getTransaction().commit();  
        session.close();  
    }  
}
```


Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

Новые видео

Исключения	0
Итераторы и генераторы	0

Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

Теги

.NET Developer
Frontend Developer



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

