

传智播客

《SSH 框架整合实战教程》

教学设计

课程名称： SSH 框架整合实战教程

授课年级： 2015 年级

授课学期： 2015 学年第二学期

教师姓名： 某某老师

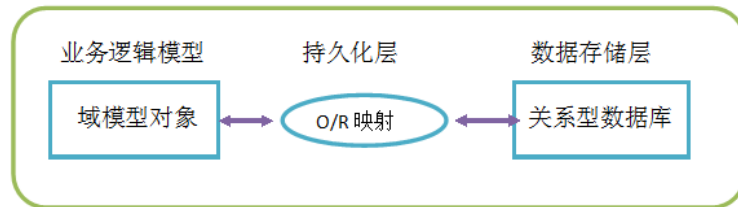
2015 年 10 月 11 日

课题名称	第7章 初识 Hibernate	计划学时	6 课时
内容分析	Hibernate 框架是当今主流的 Java 持久层框架之一，由于它具有简单易学、灵活性强、扩展性强等特点，能够大大地简化程序的代码量，提高工作效率，因此受到广大开发人员的喜爱。从本章开始，将对 Hibernate 的相关知识进行讲解。		
教学目标及基本要求	1、了解 Hibernate 的基础知识 2、学会搭建 Hibernate 环境 3、掌握 Hibernate 的核心配置和核心 API		
教学重点	1、使用 Hibernate 框架，完成一个增删改查操作 2、Hibernate 的映射文件*. hbm. xml 详解 3、Hibernate 的配置文件 hibernate. cfg. xml 详解 4、Hibernate 框架的核心 API		
教学难点	1、第一个 Hibernate 程序 2、Hibernate 的映射文件*. hbm. xml 详解 3、Hibernate 的配置文件 hibernate. cfg. xml 详解		
教学方式	讲师课堂教学要以 PPT 讲授为主，并结合多媒体进行教学		
教学过程	<p style="text-align: center;">第一课时 (Hibernate 的原理、优势、执行流程)</p> <p>一、创设情境，导入 Hibernate 的概念</p> <p>(1) 回顾 JDBC，引出 Hibernate 的概念</p> <p>让学生回顾一下使用 JDBC 操作数据库的过程，由于使用 JDBC 需要大量的重复代码，会形成过量冗余的持久层代码量。为了解决这个问题，本节我们将来学习一个当今主流的 Java 持久层框架—Hibernate。</p> <p>(2) 进入主题，讲解 Hibernate 的概念</p> <p>引出主题，Hibernate 框架是当今主流的 Java 持久层框架之一，由于它具有简单易学、灵活性强、扩展性强等特点，能够大大地简化持久层的代码量，提高工作效率。</p> <p>(3) 明确学习目标</p> <ol style="list-style-type: none"> 1、要求学生理解 Hibernate 的技术原理。 2、了解 Hibernate 的优势。 3、下载、了解 Hibernate 的目录结构。 4、理解 Hibernate 的执行流程。 <p>二、进入重点知识的讲解</p> <p>1、什么是 Hibernate?</p> <p>Hibernate 框架是由澳大利亚 Gavin King 于 2003 年开发，已成为当今主流的 Java 持久层框架之</p>		

一。Hibernate 是一个开放源代码的 ORM（Object Relational Mapping，对象关系映射）框架，它对 JDBC 进行了轻量级的对象封装，使得 Java 开发人员可以使用面向对象的编程思想来操作数据库。

2、什么是 ORM?

教师首先结合教材向学生介绍 ORM 的概念，然后结合 ORM 原理图介绍 ORM 工作原理。参考图如下：



3、Hibernate 的使用优势?

通过前面的学习，学生对 Hibernate 已经有了大致的了解。教师可提问让学生总结一下，使用 Hibernate 的优势有哪些，根据学生的回答，教师再做总结。可参考如下优势：

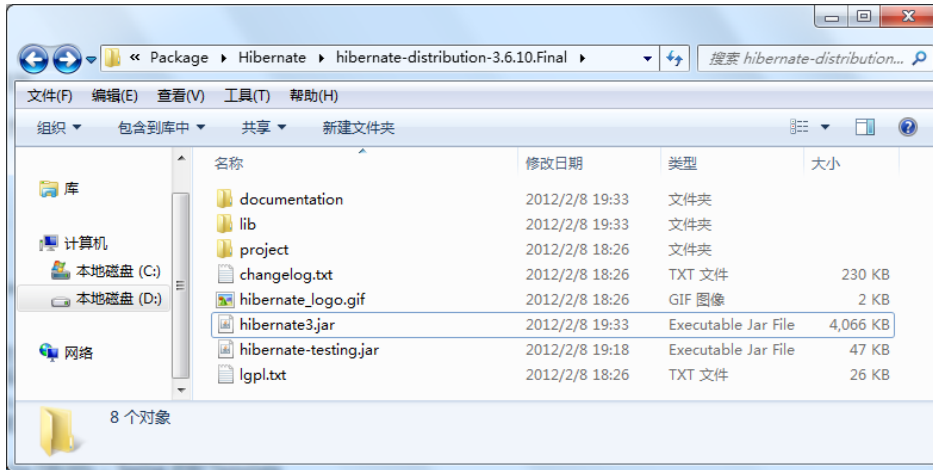
- 减化代码重复性，减少内存消耗，提高了性能
- 对 JDBC 进行轻量级封装，简化了 DAO 层代码开发量
- 使用 Java 的反射机制，而不是使用字节码增强程序类并实现透明性
- 支持很多关系型数据库，从一对一到多对多的各种复杂关系
- 开源，可自主的扩展功能

4、如何下载 Hibernate?

对 Hibernate 的概念和优势有了一个整体的了解后。教师带领学生一起下载 Hibernate，可参考如下步骤：

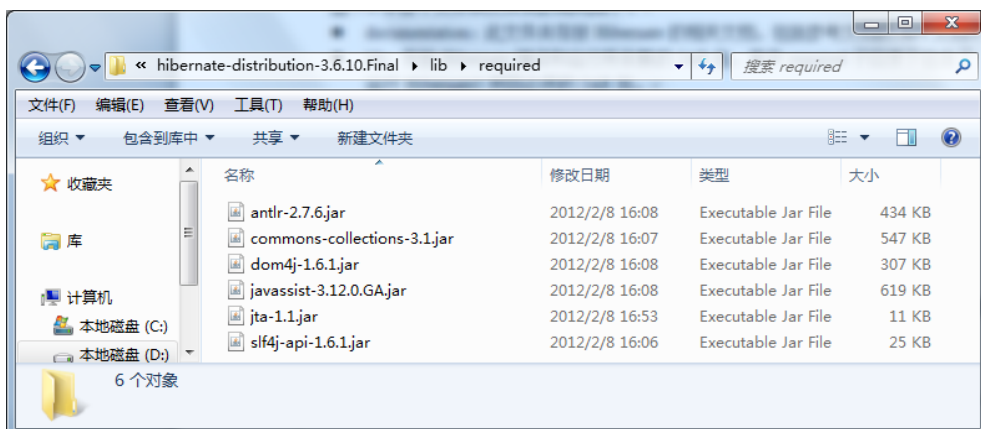
1、由于目前企业主流使用的 Hibernate 版本以 Hibernate3.x 版本为主，所以教师带领学生从网址：<http://sourceforge.net/projects/hibernate/files/hibernate3> 下载 Hibernate3.x 版本（具体细节可参考教材 7.1.2）

2、对下载完的目录结构进行介绍。hibernate3.6.10 版本下载后，解压完的目录结构如下所示：



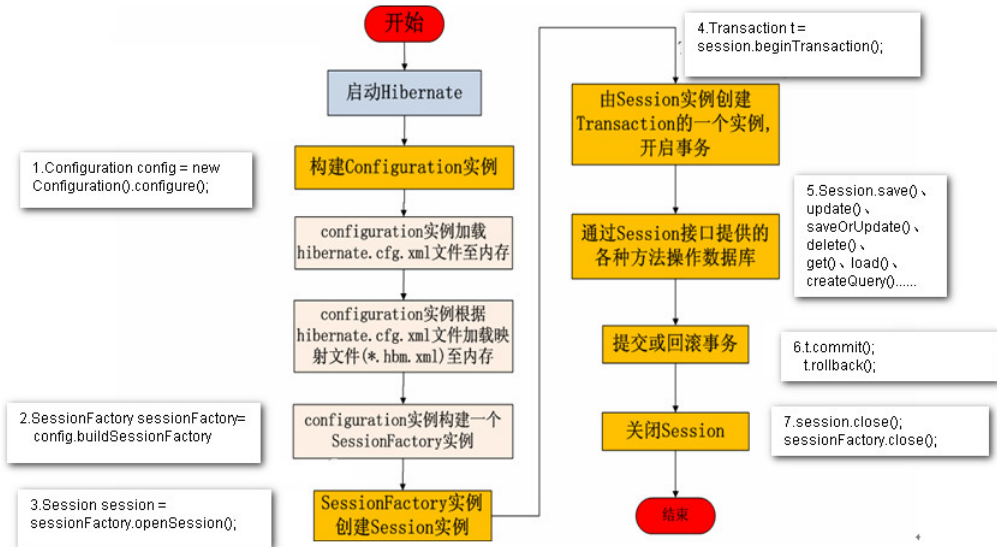
教师对子目录中存放的功能文件的作用进行介绍，具体如下：

- **documentation** 文件夹：文档目录（有用户指南、使用 API、快速实践等）
- **lib** 文件夹：Hibernate 相关 JAR 包
- **project** 文件夹：Hibernate 相关的源代码
- **hibernate3.jar** 文件：Hibernate3 的核心 JAR
- **lib/required** 子目录：Hibernate3 项目必须的 JAR 包（如下图）



5、Hibernate 的执行流程？

在使用 Hibernate 前，先了解一下 Hibernate 的执行流程。教师先来介绍在 Hibernate 开发过程中会用到的 5 个核心接口以及他们的作用然后结合下图，对 Hibernate 的执行流程进行介绍（可参考教材 7.1.3Hibernate 持久化操作的主要步骤）



三、归纳总结，随堂练习，布置作业

- (1) 对课堂上讲解的知识点进行总结，使用博学谷系统中的随堂练习题巩固本节课的知识点。
- (2) 让学生动手练习，解压查看目录结构，并画 Hibernate 的执行流程图，加深理解，巩固本节的学习内容。

第二课时

(实现第一个 Hibernate 程序)

一、回顾上节课内容，讲解如何实现第一个 Hibernate 程序

- (1) 对上节课留的作业进行答疑
- (2) 回顾上节课内容，引出本节课主题

通过上节课的学习，学生对 Hibernate 的概念、执行流程有了初步的了解。本节课将通过具体案例向学生讲解如何使用 Hibernate 对 MySQL 数据库表中数据进行增删改查操作。

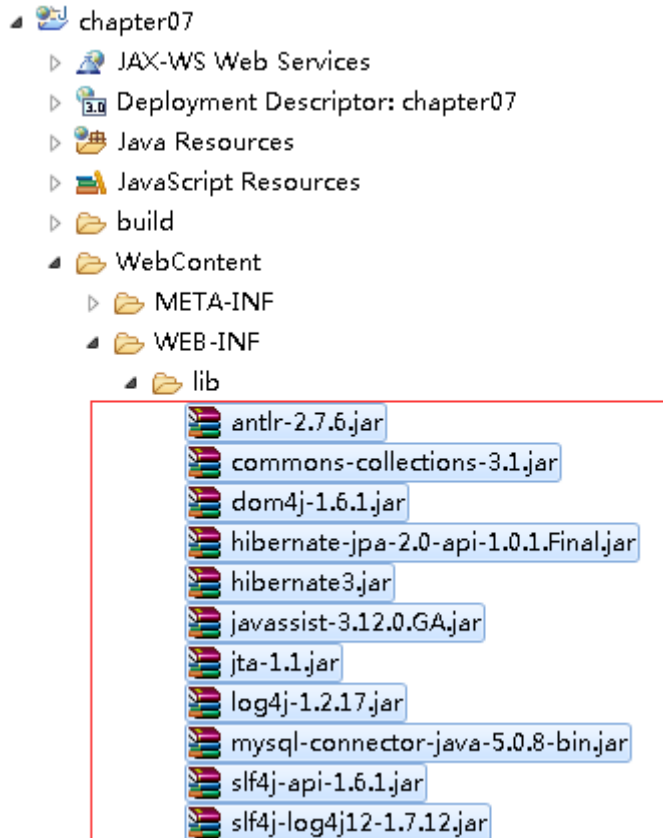
(3) 明确学习目标

- 1、掌握实现 Hibernate 程序所需要使用的 JAR 包。
- 2、了解实体类的编写要求。
- 3、熟悉 Hibernate 配置文件和映射文件的编写。
- 4、学会使用 Hibernate 完成一个基本的增删改查操作。

二、进行重点知识的讲解

1、Hibernate 程序所需要的 JAR 包有哪些？

我们都知道，一个框架的开发实现，正是由一个一个的 JAR 包组成，所以，在使用 Hibernate 框架前，先带领学生查看使用框架所需的 JAR 包有哪些。具体如下：



关于 JAR 包的说明如下：

- Hibernate 的核心 JAR 包:hibernate3. jar
- lib\required 子目录下的 6 个 JAR 包
- JPA 接口开发包:hibernate-jpa-2.0-api-1.0.1.Final.jar
- Log4j 日志包: log4j-1.2.17.jar
- Hibernate 使用的日志系统包: slf4j-log4j12-1.7.12.jar
- MySQL 数据的连接包: mysql-connector-java-5.0.8-bin.jar, 共为 11 个 JAR 包

2、创建数据库及表

教师首先向学生介绍使用的数据库版本，以及创建要求。本次使用的数据库版本为 MySQL5.5。然后在 MySQL 中创建一个名称为 hibernate 的数据库，在此数据库中创建一张名为 customer 的数据表。表中有 5 个字段，分别为 id, name, age, sex, city, 其中主键为 id。教师演示创建过程并带领学生一起查看创建完的表结构。具体 sql 语句如下所示：

```
CREATE TABLE customer (  
    id INT(11) NOT NULL AUTO_INCREMENT COMMENT '主键 id',  
    name VARCHAR(20) DEFAULT NULL COMMENT '姓名',  
    age INT(11) DEFAULT NULL COMMENT '年龄',  
    sex VARCHAR(2) DEFAULT NULL COMMENT '性别',  
    city VARCHAR(20) DEFAULT NULL COMMENT '城市',  
    PRIMARY KEY (`id`)  
);
```

3、如何编写实体类（持久化类）？

教师首先对 Hibernate 持久化类的概念进行介绍（可参见教材 7.2.3 小节）。然后对实体类进行编写，编写前需要对实体类的编写要求进行介绍，具体如下：

- 必须提供无参数 public 构造器。
- 持久化类中所有属性使用 private 修饰，提供 public 的 getters 和 setters 方法。
- 必须提供标识属性 OID，与数据库表中主键对应，例如 Customer 类 id 属性。
- 持久化类属性应尽量使用基本数据类型的包装类型。例如 int 类型需要使用 Integer 、 long 类型的使用 Long，目的是为了与数据库表的字段默认值 null 一致。
- 持久化类不要用 final 修饰，使用 final 将无法生成代理对象进行优化。

具体的代码教师可参考教材进行编写（可参见教材 文件 7-1）。

4、如何编写 Hibernate 的映射文件和核心配置文件？

1、映射文件

教师首先讲解，虽然我们已经创建了实体类，可是实体类 Customer 目前还不具备持久化操作的能力，而 Hibernate 需要知道实体类 Customer 映射到数据库 Hibernate 中的哪个表，以及类中的哪个属性对应数据库表中的哪个字段，这些都需要在映射文件中配置。

然后教师向学生演示映射文件的创建方式。首先映射文件需要在实体类 Customer 所在的包中创建，名称的命名规则为：实体类名.hbm.xml，这里我们创建的映射文件名为 Customer.hbm.xml。在该映射文件中定义实体类 Customer 的属性是如何映射到 customer 表的列上的（可参见教材 文件 7-2），对映射文件重要元素节点的介绍也可参照教材进行介绍。

2、核心配置文件

教师首先对配置文件的作用进行介绍。Hibernate 的配置文件主要用来配置数据库连接以及 Hibernate 运行时所需要的各个属性的值。此文件需要在项目的 src 目录下创建，本节所使用的配置文件名称为 hibernate.cfg.xml（可参见教材 文件 7-3）。

5、如何对数据表中的数据进行增删改查操作？

教师结合教材 7.2.6 小节，向学生分别演示使用 Hibernate 对 customer 表进行的增删改查操作的过程。

三、归纳总结，随堂练习，布置作业

- (1) 对课堂上讲解的知识点进行总结，使用博学谷系统中的随堂练习题巩固本节课的知识点。
- (2) 让学生动手练习，完成一个基本的 Hibernate 增删改查操作，巩固本节的学习内容。

第三课时

（详细讲解 Hibernate 的映射文件和核心配置文件）

一、回顾上节课内容，继续详解 Hibernate 的映射文件和核心配置文件

- (1) 对上节课留的作业进行答疑。

(2) 回顾上节课内容，引出本节课主题

上一节中，使用 Hibernate 实现了一个简单的增删改查案例，在案例中，已经接触过 Hibernate 的映射文件和配置文件。本节将对这两个文件进行详细的讲解。

(3) 明确学习目标

记住两个文件的编写要求，能够熟练编写这两个文件。

二、进行重点知识的讲解

1、如何编写 Hibernate 的映射文件？

教师首先讲解：Hibernate 映射文件的结构基本都是相同的，然后展示其代码的普遍形式，如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<!--映射文件的 dtd 信息 -->
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <!-- name 代表的是实体类名 table 代表的是表名 -->
    <class name="XXX" table="xxx">
        <!-- name=id 代表的是 XXX 类中属性 column=id 代表的是 xxx 表中的字段 -->
        <id name="id" column="id">
            <generator class="native"/><!-- 主键生成策略 -->
        </id>
        <!-- 其它属性使用 property 标签来映射 -->
        <property name="XXX" column="xxx" type="string" />
    </class>
</hibernate-mapping>
```

教师根据教材 7.3.1 小节内容，向学生介绍该文件中 hibernate-mapping、class、id、property 等各元素的作用。

2、如何编写 Hibernate 的配置文件？

教师首先介绍，Hibernate 的配置文件有 2 种格式，分别是：

- 一种是 properties 属性文件格式的配置文件，它使用键值对的形式存放信息，默认文件名称为 hibernate.properties；
- 另一种是 XML 格式的配置文件，XML 配置文件的默认名称为 hibernate.cfg.xml。

然后向学生介绍这两种配置文件的使用特点，最后教师向学生展示在 hibernate.cfg.xml 配置文件中内嵌 C3P0 的过程（参照教材 7.3.2 小节）。

三、归纳总结，随堂练习，布置作业

(1) 对课堂上讲解的知识点进行总结，使用博学谷系统中的随堂练习题巩固本节课的知识点。

(2) 让学生动手练习，练习两个核心文件的配置过程，并在核心文件中内嵌 C3P0 连接池。

第四课时

(Hibernate 的核心 API)

一、回顾上节课内容，讲解 Hibernate 的 6 个核心 API

(1) 对上节课留的作业进行答疑。

(2) 回顾上节课的内容，引出本节课主题。

上一节对 Hibernate 的核心文件进行了详细的讲解，本节课程我们将对 Hibernate 常用的 6 个 API 进行讲解。

(3) 明确学习目标

1、理解核心 API 的作用和底层原理。

2、掌握核心 API 的使用。

二、进行重点知识的讲解

1、Configuration

教师参考教材内容，首先介绍为什么使用 Configuration 以及它的作用，然后介绍 Configuration 实例的工作过程以及生命周期。最后介绍如何创建 Configuration 实例。

2、SessionFactory

教师首先介绍 SessionFactory 接口的作用，然后介绍 SessionFactory 实例的获取方式，以及 SessionFactory 的特点、应用场景，最后通过代码展示，向学生介绍 SessionFactory 的实际应用。

3、Session

教师首先介绍 Session 的作用和功能。然后介绍并演示获取 Session 实例的方式，以及 Session 的特点。最后对 Session 中的常用方法进行介绍。

4、Transaction

教师首先介绍 Transaction 接口的作用和特点。然后介绍 Transaction 接口事务对象的开启方式以及 Transaction 接口中提供的事务管理的常用方法。最后介绍 Transaction 接口的调用过程。

5、Query

教师首先介绍，什么是 Query 以及在 Hibernate 中 Query 的使用方法。然后介绍在 Hibernate 中使用 Query 对象的 5 个步骤。最后教师通过具体示例向学生演示 Query 对象的查询操作（可参

考教材 文件 7-5) 并介绍 Query 中的一些其他常用查询方法。

6、Criteria

教师首先介绍什么是 Criteria 以及 Criteria 的特点，然后介绍 Criterion 接口的创建方式和 Criteria 对象查询数据的主要步骤。最后通过具体案例向学生演示 Criteria 对象的查询操作（可参考教材 文件 7-6）。

三、归纳总结，随堂练习，布置作业

- (1) 让学生课下动手体验，通过 Session 完成增删改操作。
- (2) 让学生动手练习，对 Query 和 Criteria 接口的参数查询、匹配查询、模糊查询进行练习。

第五课时

（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（考察知识点为 Hibernate 的配置文件、Hibernate 的核心 API）

user 表数据如下：

```
CREATE TABLE user (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    name VARCHAR(20) NOT NULL,  
    pass VARCHAR(20) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    PRIMARY KEY (`id`)  
)DEFAULT CHARSET=utf8;  
//初始化数据  
insert into user values("itcast","123456","15801123456");  
insert into user values("tom1","tom1","13801117621");  
insert into user values("tom2","tom2","13801117622");  
insert into user values("tom3","tom3","13801117623");  
insert into user values("tom4","tom4","13801117624");  
insert into user values("tom5","tom5","13801117625");  
insert into user values("tom6","tom6","13801117626");  
insert into user values("tom7","tom7","13801117627");  
insert into user values("tom8","tom8","13801117628");  
insert into user values("tom9","tom9","13801117629");  
insert into user values("tom10","tom10","13801117630");  
insert into user values("tom11","tom11","13801117631");
```

请按照以下要求完成练习。

要求如下：

- 1) 要求手动编写映射文件和配置文件。
- 2) 通过 Session 对象添加 name 为 tom, phone 为 13823471223 的用户。
- 3) 通过 Session 对象把 id 为 2 的用户 name 更新为 joe。
- 4) 通过 createQuery(HQL)的 Query 对象查询 id 为 2 的用户。
- 5) 通过 createSQLQuery(HQL)的 Query 对象查询 name 为 joe 的用户。
- 6) 通过 Criterion 对象，完成匹配查询（name 为 tom 用户）。
- 7) 通过 Criterion 对象，完成模糊查询（phone 以 138 开头的用户）。

第六课时

（上机练习）

上机一：（考察数据验证，模糊查询，分页）

user 表数据如下：

```
CREATE TABLE user (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    name VARCHAR(20) NOT NULL,  
    pass VARCHAR(20) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    PRIMARY KEY (`id`)  
)DEFAULT CHARSET=utf8;  
//初始化数据  
insert into user values("itcast","123456","15801123456");  
insert into user values("tom1","tom1","13801117621");  
insert into user values("tom2","tom2","13801117622");  
insert into user values("tom3","tom3","13801117623");  
insert into user values("tom4","tom4","13801117624");  
insert into user values("tom5","tom5","13801117625");  
insert into user values("tom6","tom6","13801117626");  
insert into user values("tom7","tom7","13801117627");  
insert into user values("tom8","tom8","13801117628");  
insert into user values("tom9","tom9","13801117629");  
insert into user values("tom10","tom10","13801117630");  
insert into user values("tom11","tom11","13801117631");
```

请按照以下要求，使用 Hibernate 完成登陆、新增、查询、删除、分页等功能

- 1) 使用用户名：itcast, 密码：123456, 查询 user 表完成登陆请求。
- 2) 登陆成功后，提供 2 个模块：新增用户页面和带条件的用户查询页面。
- 3) 新增页面提供选项：用户名、密码、手机号（必填），新增一条数据用户名为 joe, 密码为 joe, 手机号为 15832333458。
- 4) 带一个条件的查询页面，要求提供条件：用户名和手机，支持模糊查询，查询结果使用

	<p>setFirstResult(int startIndex)和 setMaxResult(int endIndex)方法分页展示（每页展示5条）。</p> <p>5) 在查询结果列表的最后一列提供删除操作按钮(实现单条数据删除)。</p>
思考题和习题	
教学后记	