第八章 Hibernate数据持久化技术

8.1.1 ORM的基本概念(page211)

- ORM
  > （**Object/Relation Mapping**）对象关系映射
- 作用
  > 实现了**Java**应用中的对象到关系数据库中的表的自动的（和透明的）持久化，使用元数据描述对象鱼数据库间的映射。
- ORM的优点
  > 可以提高生产率，增加程序的可维护性，提供更好的性能。

8.1.2 POJO与PO的概念(page211)

- POJO
  > **(Pure Old Java Object**或者**Plain Ordinary Java Object**，纯**Java**对象**)**，用来与数据库表建立映射的**Java**文件。
- PO
  > （**Persistent Object**，持久化对象），是在操作数据库时创建的对象。
- 有一下的数据库表

```
CREATE TABLE User
(
        id int,
        name varchar(20)
)
```

- 编写一个与之对应的持久化对象的类

```
public class User {
        private long id;
        private String name;
        public void setId(long id){
                this.id = id;
        }
        public void setName(String name) {
                this.name = name;
```

```
        }
        public long getId() {
                return id;
        }
        public String getName() {
                return name;
        }
}
```

加强训练

- 给出以下数据库表（page217）

```
use pubs;
go
CREATE TABLE Usertab
(
        userid int Identity primary key,
        username varchar(20),
        userpwd varchar(20)
)
INSERT INTO usertab values('tom','tom');
```

- 编写一个与之对应的持久化对象的类(page223)

```
import java.io.Serializable;
public class Usertab implements Serializable {
        private Integer userid;
        private String username;
        private String userpwd;
        public Usertab(){}
        public Usertab(String username, String userpwd) {
                this.username = username;
                this.userpwd = userpwd;
        }
        public Integer getUserid() {
                return userid;
        }
        public void setUserid(Integer userid) {
                this.userid = userid;
        }
        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
```

```
                this.username = username;
        }
        public String getUserpwd() {
                return userpwd;
        }
        public void setUserpwd(String userpwd) {
                this.userpwd = userpwd;
        }
}
```