使用session接口进行增删改查

session接口的常用方法：

save（object obj）： 保存对象

update（object obj）： 更新对象

saveOrUpdate(object obj):添加或修改一个对象

delete（object obj）： 删除一个对象

get（实体类.class,Serialize id）： 获取对象

load（实体类.class,Serialize id）： 获取一个对象

简单的crud操作：

```
package hibernateday1;


import org.hibernate.Session;
import org.hibernate.Transaction;
import org.junit.Test;


import utils.hibernateUtils;


public class demo {
        @Test
public void test1(){
                //session的save方法使用
        user user = new user();
        user.setName("jiming");
        user.setPassworld("250016");
        Session session = hibernateUtils.getSession();
        Transaction ts = session.beginTransaction();
        session.save(user);
        ts.commit();
        session.close();
}
        //update更新一个对象
        @Test
        public void test2(){
                user user = new user();
        //给user的id赋值
        user.setId(13);
        user.setName("jiming");
        user.setPassworld("250016");
        Session session = hibernateUtils.getSession();
        Transaction ts = session.beginTransaction();
        session.update(user);
        ts.commit();
        session.close();
    }
    //saveOrUpdate添加或更新一个对象
    @Test
    public void test3(){
        user user = new user();
        //当id是数据库中存在的时候，则进行更新操作，当id是数据库中不存在的时候，则进行添加操作
        user.setId(6);
        user.setName("yangzhiqin");
        user.setPassworld("anxinghua");
        Session session = hibernateUtils.getSession();
        Transaction ts = session.beginTransaction();
        session.saveOrUpdate(user);
        ts.commit();
        session.close();
    }
    //delete删除一个对象
    @Test
    public void test4(){
        user user = new user();
        //给定的id必须是数据库中存在的id，否则报错
        user.setId(21);
        Session session = hibernateUtils.getSession();
        Transaction ts = session.beginTransaction();
        session.delete(user);
        ts.commit();
        session.close();
    }
    //get根据id查询一个对象
    @Test
    public void test5(){
        Session session = hibernateUtils.getSession();
        Transaction ts = session.beginTransaction();
        user user2 = session.get(user.class,8);
        System.out.println(user2);
        ts.commit();
        session.close();
    }
    //load根据id查询一个对象
        @Test
        public void test6(){
            Session session = hibernateUtils.getSession();
            Transaction ts = session.beginTransaction();
            user user2 = session.load(user.class,17);
            System.out.println(user2);
            ts.commit();
            session.close();
        }
}
```