```
//说明持久态对象确实可以更新数据库的数据
                //获取到一个持久态对象
                person p1 = session.get(person.class, 6);
                //修改p1的name属性为安启力
                p1.setName("安启力");
                //没有必要执行update语句，因为现在持久态对象已经能够更新数据库中的数据了。
                //此时数据库中相应的数据已经被更新了
```

hibernate一级缓存和快照机制

一，一级缓存

1，什么是hibernate的一级缓存

其实就是session对象的缓存，说白了一级缓存就是session内部的一个map集合。它的作用就是为了减少程序和数据库交互的次数，从而奇高程序执行的性能。

session中的map集合就是所说的一级缓存。

（1）第一次调用session.get()方法的时候会去数据库查询数据。

（2）第二次访问调用session.get()方法的时候，会有限到session内部的一级缓存中查找是否存在该对象，如果存在就不需要在进行数据库的查询了，如果不存在的话，在去查询数据库中的数据。

代码证明hibernate的一级缓存技术是存在的：

```
Session session = hibernateUtils.getSession();
                Transaction ts = session.beginTransaction();
                //使用代码证明hibernate的一级缓存技术是存在的
                //第一次查询
                person p1 = session.get(person.class, 4);
                System.out.println(p1);
                //第二次查询
```

```
person p2 = session.get(person.class, 4);
System.out.println(p2);
ts.commit();
session.close();
```

其中p1和p2打印的地址是一样的，说明第二次在查找的时候，是直接在session中取出的，而并没有去查询数据库。

二，hibernate的快照技术

快照机制：其实就是session内部的一个map集合，用于备份数据库中的数据，用于和一级缓存的数据进行对比，以便实现持久态对象更新数据库对象的效果。

为什么持久态对象可以影响数据库中的数据？
因为hibernate的session内部存在一级缓存和快照机制。

快照中的数据是不可以被开发者进行更改的。

当程序执行commit程序的时候：
（1）执行session.flush()方法，这个方法对比一级缓存和快照区的数据，如果发现数据有差异，就会产生SQL语句。
（2）hibernate就发送SQL语句到数据库执行。
（3）hibernate执行提交事务的方法，数据库提交到数据库。
（4）把快照中的数据进行更新。

在慎重操作持久态对象。