1. POJO（Plain Ordinary Java Object）简单的Java对象，实际就是普通JavaBeans，是为了避免和EJB混淆所创造的简称。
2. 使用POJO名称是为了避免和[EJB](http://baike.baidu.com/view/3542.htm" \t "http://baike.baidu.com/_blank)混淆起来, 而且简称比较直接. 其中有一些属性及其getter setter方法的类,没有业务逻辑，有时可以作为[VO](http://baike.baidu.com/view/1097386.htm" \t "http://baike.baidu.com/_blank)(value -object)或[dto](http://baike.baidu.com/view/160599.htm" \t "http://baike.baidu.com/_blank)(Data Transform Object)来使用.当然,如果你有一个简单的运算属性也是可以的,但不允许有业务方法,也不能携带有connection之类的方法。
3. 自身特点：
   1. POJO是Plain OrdinaryJava Object的缩写不错，但是它通指没有使用Entity Beans的普通java对象，可以把POJO作为支持业务逻辑的协助类。
   2. POJO实质上可以理解为简单的实体类，顾名思义POJO类的作用是方便 程序员使用数据库中的数据表，对于广大的程序员，可以很方便的将POJO类当做对象来进行使用，当然也是可以方便的调用其get,set方法。POJO类 也给我们在struts框架中的配置带来了很大的方便。

1. 实例：

POJO有一些private的[参数](http://baike.baidu.com/view/327406.htm" \t "http://baike.baidu.com/_blank)作为对象的属性。然后针对每个[参数](http://baike.baidu.com/view/327406.htm" \t "http://baike.baidu.com/_blank)定义了get和set方法作为访问的接口。例如：

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;

}

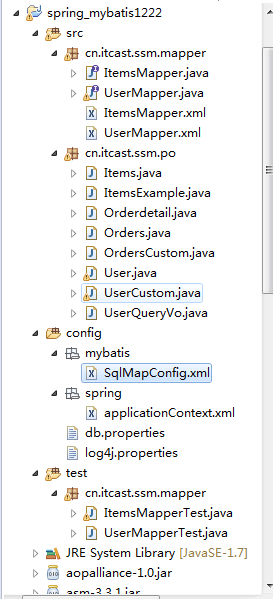
}

POJO对象有时也被称为[Data](http://baike.baidu.com/view/68441.htm" \t "http://baike.baidu.com/_blank)对象，大量应用于表现现实中的对象。如果项目中使用了Hibernate[框架](http://baike.baidu.com/view/66971.htm" \t "http://baike.baidu.com/_blank)，有一个关联的xml文件，使对象与数据库中的表对应，对象的属性与表中的字段相对应。

1. POJO = "Plain Old Java Object"，是MartinFowler等发明的一个术语，用来表示普通的Java对象，不是JavaBean, EntityBean 或者 SessionBean。POJO不担当任何特殊的角色，也不实现任何特殊的Java框架的接口如，[EJB](http://baike.baidu.com/view/3542.htm" \t "http://baike.baidu.com/_blank)，[JDBC](http://baike.baidu.com/view/25611.htm" \t "http://baike.baidu.com/_blank)等等。

即POJO是一个简单的普通的Java对象，它不包含[业务逻辑](http://baike.baidu.com/view/1642754.htm" \t "http://baike.baidu.com/_blank)或持久逻辑等，但不是JavaBean、EntityBean等，不具有任何特殊角色和不继承或不实现任何其它Java框架的类或接口。

1. 在mybatis中需要使用到pojo类，一般都是进行多表查询的时候，sql语句中查询的字段包含多个实体类中的字段，这时我们就需要创建pojo类，让该pojo类继承字段最多的实体类，然后将sql语句要查询的字段中没被继承的表的字段写到该pojo类中，其实pojo类就是联合查询的时候两个表的联合实体类。用于将sql语句中的联合查询的结果映射到pojo类中。
2. 在一个mybatis项目中主要由下面几个组成：pojo类（即实体类）、mapper.xml(相当于dao的实现类)、mapper.java（相当于dao接口）、mybatis的配置文件SqlMapConfig.xml。



1. 是否