（只做使用过的笔记，有些没用过的用到了再来补充）

常用的元素

select：查询

insert：插入

update：修改

delete：删除

resultMap：最强大的地方，字面意思，结果集，灵活处理查询返回的结果，实际运用会有体会

select常用的属性

id：相当于一个select的名字，唯一标识符

parameterType：指定了参数的类型

resultType：结果的返回类型，不涉及一些乱七八糟的关系映射，只要返回简单的一些属性使用

resultMap：结果集，区别于上一个，这个就是专门处理乱七八糟的关系映射时使用，所以看情况使用两者

insert,update和delete一般只会用到id这个属性

insert还会用到的属性

userGeneratedKeys：获取由数据库内部自动生成的主键

keyProperty：设置到userGeneratedKeys获取的目标属性上，一般就是id

parameterType：如果有多个参数但都是一个持久化类中属性类型，直接以这个持久化类作为参数类型，然后随你用几个属性，但是用xml配置时，记住这样使用的话，接口的方法参数都必须是这个持久化类

```
<insert id="saveUser" parameterType="com.jlb.entity.User" useGeneratedKeys="true"
    keyProperty="id">
    insert into tb_user(username,loginname,password,phone,address)
    values(#{username},#{loginname},#{password},#{phone},#{address})
</insert>
```

```
void saveUser(User user);
```

就像这样，否则你还按xml中的参数一个个列举出来，最后会报not found错误，当然有一个特例，你只用了一个属性，可以正常执行，用了两个以上的属性时，在使用xml配置时，要记得以上的操作

resultMap：结果集，谁用谁知道，很强大，但也很烦，但多写点多理解点，习惯后也很简单。主要难度在哪呢，要能灵活运用其中的collection（一对多的关联映射，多对多对一方的某个实例来言也是一对多）和association（一对一的关联映射，多对一对多方的某个实例来言也是一对一）

resultMap的常用属性

id：一个resultMap的唯一标识符

type：这个结果集最终返回的类型（对应一个持久化类）

resultMap的子标签

result：数据库表的普通列，property对应持久化类中的属性，column对应数据库表中的列名

association：一对一的关联映射，property对应持久化类中的属性，column对应数据库表中的列名，javaType对应property属性对应的类型（另一个持久化类），select对应一个查询（一般就是另一个持久化类对应的mapper中的一个查询），没有用多表查询时，property属性对应的对象总得有数据获取源，就通过这种方式，这反向select的参数一般就是column对应的属性在本例中查询获得的值

collection：一对多的关联映射，property不详细说了，javaType在这一般都是ArrayList集合，因为获取多方嘛，肯定不止一组数据，ofType指定ArrayList集合中的类型，column表示使用本例哪个属性作为反向select的参数，select一个意思