###### 什么是动态SQL？

 动态sql是mybatis的核心，主要是对sql语句进行灵活操作，通过表达式进行判断，对sql进行灵活的拼接、组装。

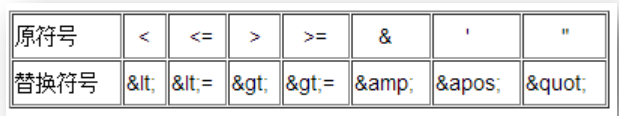
* 动态SQL用于解决查询条件不确定的情况：
* 在程序运行期间，根据用户提交的查询条件进行查询。
* 提交的查询条件不同，执行的SQL语句不同。
* 若将每种可能的情况均逐一列出，对所有条件进行排列组合，将会出现大量的SQL语句。

动态SQL，即通过MyBatis提供的各种标签对条件作出判断以实现动态拼接SQL语句。条件判断使用的表达式为OGNL（Object-Graph Navigation Language对象图导航语言）表达式。

常用的动态SQL标签有<if>、<where>、<choose/>、<foreach>等。MyBatis的动态SQL语句，与JSTL中的语句非常相似。

###### 注意事项：

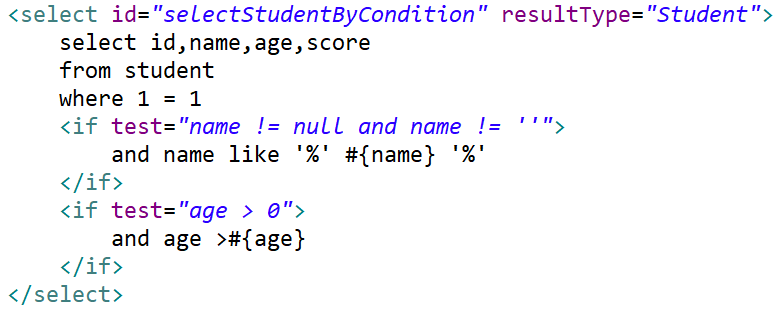
在mapper的动态SQL中若出现大于号，小于号，大于等于号，小于等于号等符号，最好将其转换为实体符号。否则，XML可能出现解析错误问题。（特别是小于号绝对不能出现）

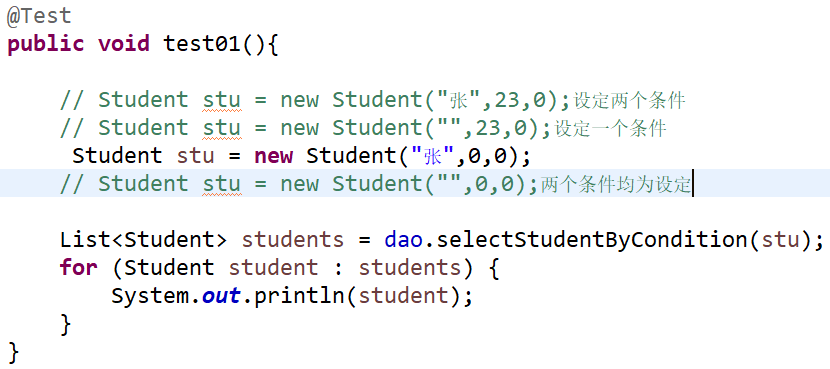


###### <if/>标签

对该标签的执行，当test值为true时，会将其包含的SQL片段拼接到其所在的SQL语句中。

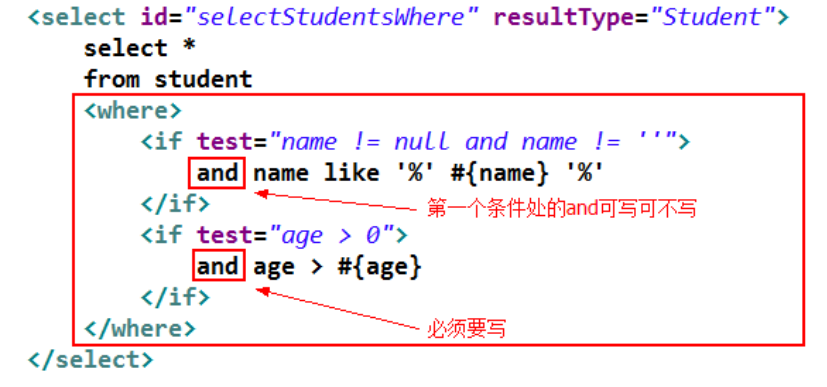
本例实现的功能是：查询出满足用户提交查询条件的所有学生。用户提交的查询条件可以包含一个姓名的模糊查询，同时也可以包含一个年龄的下限。当然用户在提交表单时可能两个条件均做出了设定，也可能两个条件均不做设定，也可以只做其中一项设定。为了解决两个条件均未做出设定的情况，在where后添加一个“1=1”的条件。这样就不至于两个条件均未设定而只剩下一个where，而没有任何拼接的条件的不完整SQL语句。

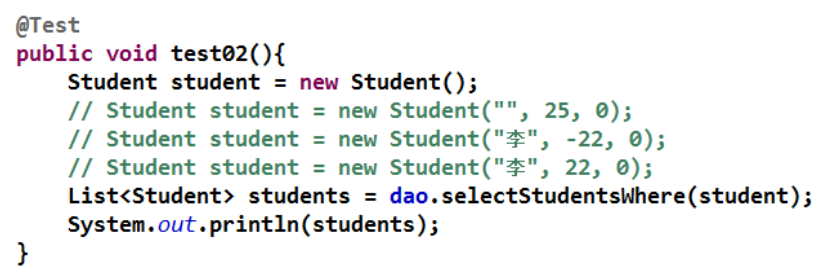




###### <where/>标签

<if/>标签中存在一个比较麻烦的地方：需要在where后手工添加1=1的句子。因为，若where后的所有<if/>条件均为false，而where后若没有1=1，则SQL中就会只剩下一个空的where，SQL出错。所以，在where后，需要添加永为真子句1=1，担当数据量很大的时候，会严重影响查询效率。



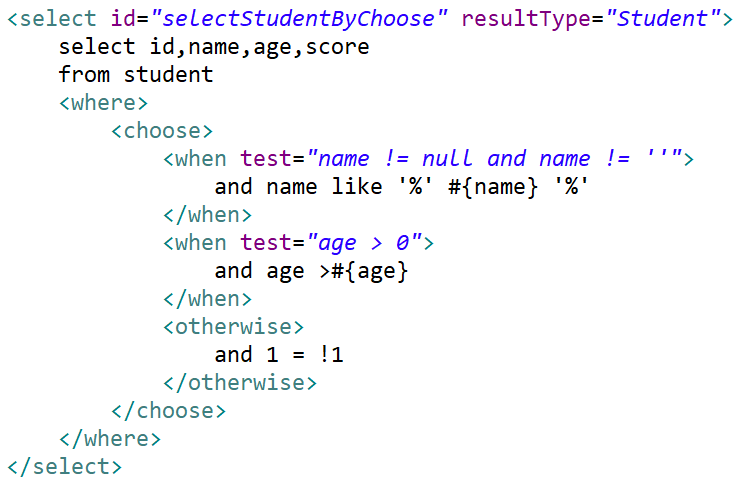


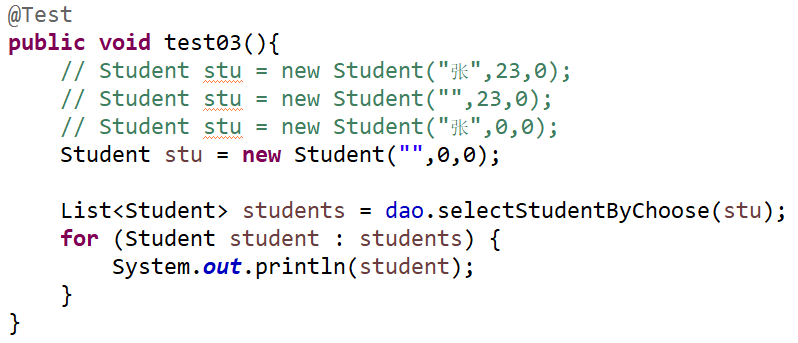
###### <choose/>标签

该标签中只可以包含<when/><otherwise/>，可以包含多个<when/>与一个<otherwise/>。他们联合使用，完成Java中的开关语句switch...case功能。

本例要完成的需求是，若姓名不空，则按照姓名查询；若姓名为空，则按年龄查询；若没有查询条件，则没有查询结果。

对于<choose/>标签，其会从第一个<when/>开始逐个向后进行条件判断。若出现<when/>中的test属性值为true的情况，则直接结果<choose/>标签，不再向后进行判断查找。若所有的<when/>的test判断结果均为false，则会执行<otherwise/>标签





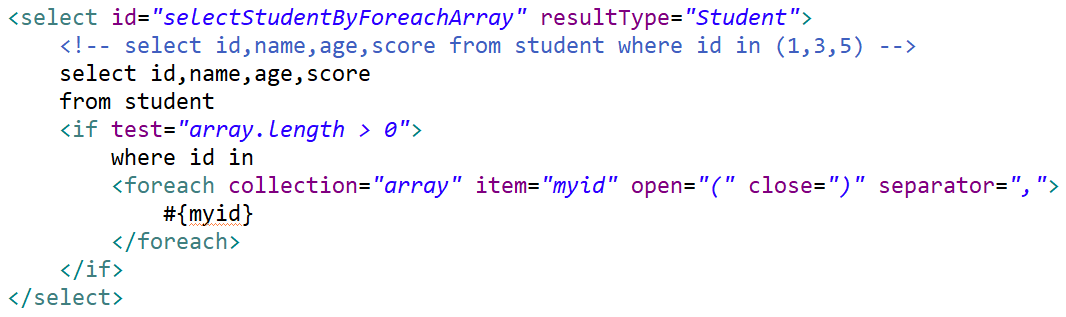
###### <foreach/>标签遍历数组

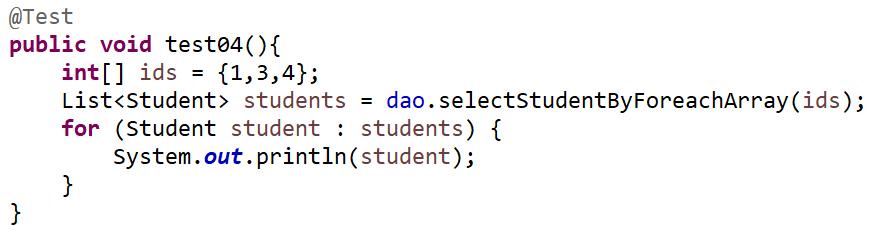
<foreach/>标签用于实现对于数组与集合的遍历。

注意点：

* collection表示要遍历的集合类型，这里是数组，即array。
* open、close、separator为对遍历内容的SQL拼接。

本例实现需求是，查询id为1，3和4的学生信息。

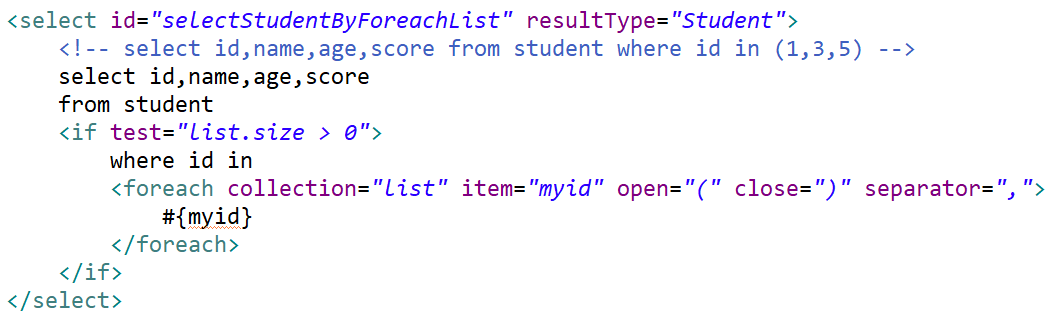


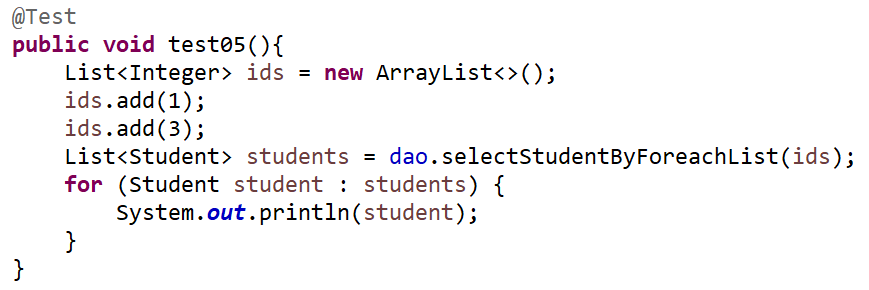


###### <foreach/>标签遍历泛型为基本类型的List

本例实现的需求是，查询出id为1，3的学生信息。

OGNL表达式中的List使用list表示，其大小使用list.size表示。

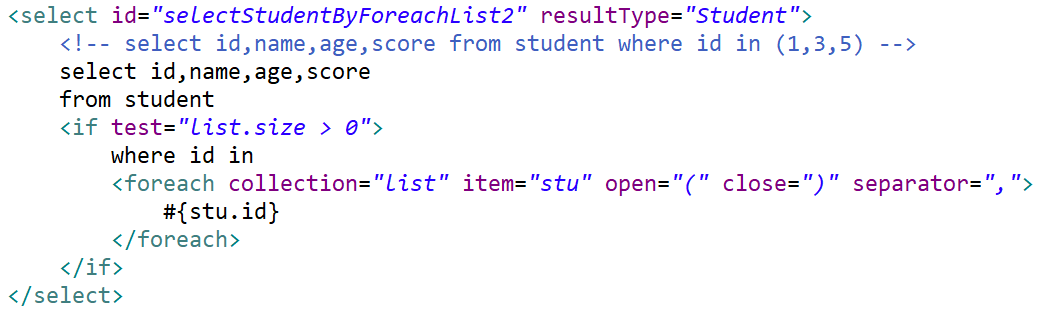


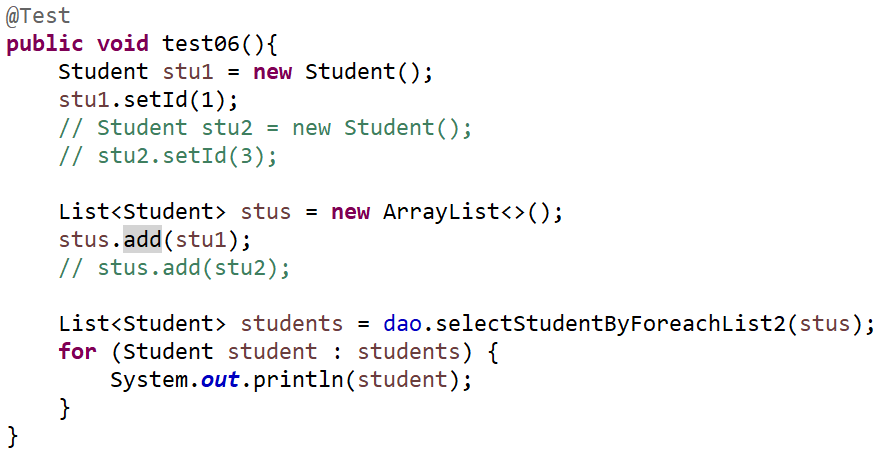


###### <foreach/>标签遍历泛型为自定义类型的List

本例实现的需求是，查询出id为1与3学生信息。

注：这里的当前遍历对象类型是List中的泛型，即是Student对象。





###### <sql/>标签

<sql/>标签用于定义SQL片段，以便其他SQL标签复用。而其它标签使用该SQL片段，需要使用<include/>子标签。该<sql/>标签可以定义SQL语句中的任何部分，所以<include/>子标签可以放在动态SQL的任何位置。