MyBatis Mapper接口的工作原理，Mapper接口里的方法参数不同是，方法能重载吗

题目难度：★★★

知识点标签：Mapper接口开发规范，Mapper接口的工作原理，Mapper接口里的方法参数不同是，方法能重载吗

学习时长：30分钟

题目描述

通常一个Xml映射文件，都会写一个Mapper接口与之对应，请问，这个Mapper接口的工作原理是什么？Mapper接口里的方法，参数不同时，方法能重载吗？

解题思路

> 面试官问题可以从几个方面来回答：Mapper接口开发规范，Mapper接口的工作原理，Mapper接口方法参数不同时，方法是否能重载？

Mapper接口开发规范

Mapper接口开发需要遵循以下规范：

1、 Mapper.xml文件中的**namespace与mapper接口的类路径相同。**

2、 Mapper**接口方法名**和Mapper.xml中定义的**每个statement的id相同**

3、 Mapper接口方法的**输入参数类型**和mapper.xml中定义的每个sql 的**parameterType的类型相同**

4、 Mapper接口方法的输出参数类型和mapper.xml中定义的每个sql的resultType的类型相同

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!-- 命名空间，对sql进行分类化管理（sql隔离） -->
<mapper namespace="com.lagou.mapper.UserMapper">
    <!-- 在映射文件中配置sql语句 -->
    <!-- 通过select执行查询，id用于标识映射文件中的sql(Statement-id)
    将sql语句封装到mappedstatement中
    #{}表示占位符
    parameterType-指定输入参数的类型
    #{id}-id表示输入的参数，参数名称就是id,如果输入参数是简单类型，#{}中的参数可以任意
    resultType-指定sql输出结果所映射的java对象类型
```

```
    -->
    <!-- 通过id查询用户表的记录 -->
    <select id="findUserById" parameterType="int"
resultType="com.lagou.po.User">
        select *from user where id=#{id}
    </select>

    <!-- 根据用户名称模糊查询用户信息 -->
    <!-- resultType-指定单条记录所映射的对象类型
    ${}拼接sql串，接收参数的内容，不加任何修饰，拼接在sql中(存在sql漏洞)
    ${}接收输入参数的内容，如果传入的类型是简单类型，${}中只能使用value
    -->
    <select id="findUserByName" parameterType="java.lang.String"
resultType="com.lagou.po.User">
        SELECT *FROM USER WHERE username LIKE '%${value}%'
    </select>

    <!-- 添加用户 -->
    <!-- 指定输入参数类型是pojo(包括用户信息)
        #{}中指定pojo（User）属性名,接收到pojo的属性值
        Mybatis通过OGNL获取对象的属性值
     -->
    <insert id="insertUser" parameterType="com.lagou.po.User">
        <!-- 获取刚增加的记录主键
            返回id到poio对象（User）
            SELECT LAST_INSERT_ID():得到刚插入金进去记录的主键值，只适用于自增逐主键
            keyProperty：将查询到的主键值设置到parameterType指定的对象User里面的用来做
id的属性，这里是：id,                          然后在执行插入的时候，从parameterType（也
就是这里的User）的Id里面取出来，进行插入
            order:指SELECT LAST_INSERT_ID()的执行顺序，相对于insert来说
（before/after）
            resultType:指定SELECT LAST_INSERT_ID()的结果类型
        -->
        <selectKey keyProperty="id" order="AFTER"
resultType="java.lang.Integer">
            SELECT LAST_INSERT_ID()
</selectKey>
        INSERT INTO USER (id,username,birthday,sex,address) VALUES(#{id},#
{username},#{birthday},#{sex},#{address})

        <!-- 使用mysql的uuid生成主键返回
             执行过程：
                首先通过uuid得到主键，然后将主键设置到id属性中
                其次在Inster执行的时候从User对象中取出id的属性值
         -->
       <!--            <selectKey keyProperty="id" order="BEFORE"
resultType="java.lang.String">
             SELECT UUID()
         </selectKey>
```

```
        INSERT INTO USER (id,username,birthday,sex,address) VALUES(#{id},#
{username},#{birthday},#{sex},#{address})         -->
    </insert>

    <!-- 根据id删除用户 -->
    <delete id="deleteUser" parameterType="java.lang.Integer">
        DELETE FROM USER WHERE id=#{id}
    </delete>

    <!-- 根据id更新用户
        传入用户id以及相关更新信息
        #{id}:从输入的user对象中获取user的属性值
    -->
    <update id="updateUser" parameterType="com.lagou.po.User">
        UPDATE USER SET username=#{username},birthday=#{birthday},sex=#
{sex},address=#{address} WHERE id=#{id}
    </update>
</mapper>
```

Mapper接口工作原理

Dao接口，就是人们常说的Mapper接口，接口的全限名，就是映射文件中的namespace的值，接口的方法名，就是映射文件中MappedStatement的id值，接口方法内的参数，就是传递给sql的参数。

Mapper接口是没有实现类的，当调用接口方法时，接口全限名+方法名拼接字符串作为key值，可唯一定位一个MappedStatement，举例：com.mybatis3.mappers.StudentDao.findStudentById，可以唯一找到namespace为com.mybatis3.mappers.StudentDao下面id = findStudentById的MappedStatement。在Mybatis中，每一个