MyBatis中对sql语句参数的传递

--这里要明白一点，由于mybatis是在dao层起作用，dao层的接口中定义了和一个sql映射文件中各条sql语句的方法，就是说接口中的方法中需要传递的参数都是与sql语句中需要的参数对应的，那么映射文件中每个sql语句中添加parameterType属性以指定传递参数是什么类型就变得可有可无了。

1、--一个参数的传递

这类情况就是mapper文件中的sql语句只需要一个参数，如按表中id查询数据。传递一个参数一般都是传递简单数据类型，dao接口中的方法一般如下：

```
public User selectUserById(Integer id);
```

映射文件中的sql语句的占位符中变量名可以随便写：select * from t_user where id=#{随便写}

当我们用接口实现类对象调用selectUserById时，动态代理生成的实现类中调用SqlSession对象的selectOne方法，将整型参数传进去，因为sql语句中只需要一个参数，这个值怎么传都会传给sql中的参数位置，所以在占位符中特意指定接收属性的参数值。

2、--多参数传递

这类情况就是mapper文件中的sql语句需要多个参数，如按表中id和（或）name查询数据，或者是insert操作，亦或是update操作等等。

多参传递有多种方式：

--1）注解@Param命名参数传参--（掌握）

@Param("参数名") 数据类型 变量名

dao接口方法中参数定义如下：

```
public List<User> selectByNameOrId(@Parama("loginName") String name,@Param("id") Integer id);
```

当调用该方法时内部是将我们输入的name（id）参数的值以注解中定义的参数名为key存储到一个集合中（这只是猜测），所以sql语句中的占位符中填写正确的对应参数名#{}

--当然sql中的条件一般都是表中的字段，所以参数名都是按表中对应字段命名。

--2）JavaBean对象传参

dao接口方法中参数定义如下：

```
public List<User> addUser(User user);
```

例如作添加数据操作：

需要创建JavaBean对象，然后给该对象相关属性赋值，最后作为参数传递给到、接口的方法中

映射文件中接收参数的占位符中填参数名就是User对象中各个属性名，为了开发方便，就得使这个实体类中JavaBean的字段名和表中的字段名一致

--3）按位置传参

dao接口方法中参数根据前后进行排序，第一个是0，第二个是1.

直接调用dao接口方法时将参数写到方法中，就完成传参操作了。

这种方式不需要我们定义0位置参数的名字，mybatis已经定义好了。mybatis3.3版本以及之前版本是0，代表第一个

3.4及以后用arg0表示第一个参数。很明显mybatis把接口方法中的参数放进了一个可变长度的参数数组，而sql语句中的占位符可以根据下标0、1、2...来接收参数，只不过下标前面加了arg。

```
select * from t_user where id=#{arg0} or loginName=#{arg1}
```

只不过这种方式不知道具体arg0代表的是什么意思的值，容易搞错。

--4）使用集合Map传值

```
public List<User> selectByMap(Map<String,Object>);
```

首先得创建Map集合，往集合中存储数据put("表中字段名",value)，集合key命名为对应表中的字段名，调用dao接口方法时把Map集合传进去。

sql语句中的占位符中填写的是map集合中的key，所以key的命名与表中一致可以很便捷。

--这四种方法用的最多的是前两种方法，在需要传递的参数涉及两张表时，首选@Param，其次是Map方式

占位符#和$的区别

mybatis映射文件中sql语句参数使用"#{参数名}"来代替JDBC中的"?"占位符,其实mybatis还有一种占位符是"${参数名}"
我们知道，JDBC有两种获取数据库操作对象的方式：
"PrepareStatement"和"Statement"，前者是将sql语句先编译，没有值的位置用"?"这个占位符来代替，后期通过PrepareStatement对象的setInt(整数值)（或setString(字符串)等等）方法给"?"赋值，然后执行sql语句。后者则是让sql语句进行按需求进行添加字符串的拼接，然后执行sql语句，有sql注入风险。
而mapper文件中sql语句里的占位符用#{}的话，底层执行时，使用的是PrepareStatement对象进行的，所以这个#{}是mybatis用来代替?的。
如果mapper文件中sql语句里的占位符用${},$表示告诉mybatis使用$表示的字符串替换这条sql语句其所在位置。主要用来替换表名列名，不同序列等操作。

这里测试根据id查询t_user表中的数据，通过输出执行日志来比较使用#或者是$占位符的区别

```
select * from t_user where id=#{id}
日志输出如下：
Opening JDBC Connection
Created connection 230643635.
Setting autocommit to false on JDBC Connection
[com.mysql.jdbc.JDBC4Connection@dbf57b3]
==>  Preparing: select * from t_user where id=?
==> Parameters: 1(Integer)
<==    Columns: id, loginName, loginPwd, realName
<==        Row: 1, xxpiaozhiyan, yanyan, 朴智妍
<==      Total: 1
User{id=1, loginName='xxpiaozhiyan', loginPwd='yanyan', realName='朴智妍'}
--所以用#得到的是：select * from t_user where id=? 然后给?赋值
```

```
select * from t_user where id=${id}
日志输出如下：
Opening JDBC Connection
Created connection 1388278453.
Setting autocommit to false on JDBC Connection
[com.mysql.jdbc.JDBC4Connection@52bf72b5]
==>  Preparing: select * from t_user where id=1
==> Parameters:
<==    Columns: id, loginName, loginPwd, realName
<==        Row: 1, xxpiaozhiyan, yanyan, 朴智妍
<==      Total: 1
User{id=1, loginName='xxpiaozhiyan', loginPwd='yanyan', realName='朴智妍'}
--所以用$得到的是：select * from t_user where id=1 然后直接运行
```