1.MyBatis在处理#{}时，是把#{}替换为 （1） ，在处理${}时，是把${}替换为 （2）

参考答案：? 变量的值

2.请将下列横线处的程序填写完整,实现1对1关联查询配置：  
<mapper namespace="com.isoft.mapping.userMapper">   
 <select id="getClass" parameterType="int" （1） ="ClassesResultMap">   
 select \* from class c,teacher t where c.teacher\_id=t.t\_id and c.c\_id=#{id}   
 </select>   
 < （2） type="com.isoft.user.Classes" id="ClassesResultMap">   
 <!-- 实体类的字段名和数据表的字段名映射 -->   
 <id property="id" column="c\_id"/>   
 <result property="name" column="c\_name"/>   
 < （3） property="teacher" javaType="com.isoft.user.Teacher">   
 <id property="id" column="t\_id"/>   
 <result property="name" column="t\_name"/>   
 </ 同（3） >   
</ 同（2） >   
</mapper>

参考答案：resultMap resultMap association

3.如何获取自动生成的(主)键值？  
 <insert id="insertUser" parameterType="cn.isoft.pojo.User">  
 <selectKey keyProperty="id" order="AFTER" resultType="int">  
 select （1）   
 </selectKey>  
 INSERT INTO USER(username,birthday,sex,address)   
 VALUES(#{username},#{birthday},#{sex},#{address)  
 </insert>

参考答案：LAST\_INSERT\_ID()

4.实现分页查询使用什么关键词？  
<select id="queryStudentsBySql" parameterType="map" resultMap="studentmapper">  
 select \* from student （1） #{currIndex} , #{pageSize}  
</select>

参考答案：limit

5.MyBatis动态注解-插入注解使用 （1） 注解？

参考答案：@InsertProvider

6.将下列程序填写完整,  
<configuration>  
 <environments default="development">  
 <environment id="development">  
 <transactionManager type="JDBC" />  
 <!-- 配置数据库连接信息 -->  
 <dataSource type="POOLED">  
 <property name="driver" value="\_\_(1)\_\_\_\_\_" />  
<property name="\_\_(2)\_\_" value="jdbc:mysql://127.0.0.1/test " />  
<property name="\_\_\_\_(3)\_\_\_" value="root" />  
<property name="\_\_(4)\_\_\_\_" value="root" />  
</dataSource>  
 </environment>  
 </environments>  
<mappers>  
<!-- 注册映射接口，该接口文件包含了数据访问方法和SQL -->  
 <mapper class="dal.dataaccess.IAdminInfoMapper" />  
 </mappers>  
</configuration>

参考答案：com.mysql.jdbc.Driver url username password

7.使用MyBatis调用存储过程，完成如下程序代码：  
 <select id="getUserCount" parameterMap="getUserCountMap" statementType=" （1） ">  
 CALL mybatis.isoft\_user\_count(?,?)  
 </select>

参考答案：CALLABLE

8.依赖注入的作用就是在使用Spring框架创建对象时，动态的将其所依赖的对象注入到( )组件中

参考答案：Bean

9.MyBatis的删除操作在映射文件中是通过配置( )元素来实现的

参考答案：delete

10.使用MySQL中的( )函数进行字符串拼接，可以有效防止模糊查询时可能出现的SQL注入问题

参考答案：concat()

11.当在MyBaits文件中使用了<bind>元素，在SQL语句中可以直接引用< bind>元素的( )属性值即可进行动态SQL组装

参考答案：name

12.对于数据源的配置，MyBatis框架提供了UNPOOLED、( )和JNDI三种数据源类型。

参考答案：POOLED

13.当数据表中的列和需要返回的对象的属性不完全一致， MyBatis是不会自动赋值的。此时，就可以使用( )元素进行处理

参考答案：resultMap

14.MyBaits映射文件中使用< incclude>元素的( )属性可以引用自定义的代码片段，其属性值为自定义代码片段的id

参考答案：refid

15.MyBaits中<trim> 元素的作用是去除一些特殊的字符串，它的( )属性代表的是语句的前缀，而prefixOverrides属性代表的是需要去除的哪些特殊字符串。

参考答案：prefix

16.Spring MVC提供了一个前端控制器( )，使开发人员无需额外开发控制器对象

参考答案：DispatcherServlet

17.RESTful风格在HTTP请求中，使用( )、delete、post和get方式分别对应添加、删除、修改和查询的操作。

参考答案：put

18.如果拦截器类中的preHandle()方法的返回值为( )，则程序会继续向下执行处理器中的方法

参考答案：true

19.HttpStatus类型代表的是Http协议中的状态，示例中的HttpStatus.OK状态码是( )，即服务器已成功处理了请求

参考答案：200

20.使用基于注解方式的事务管理需要在使用事务的Spring Bean类或者Bean类的方法上添加注解（ ）

参考答案：@Transactional

21.jQuery中方法ajax()实现异步上传，其中选项success表示( )

参考答案：提交成功的回调函数

22.请将下列横线处的程序填写完整,实现UserMapperI接口的sql映射:  
package com.iss.mapper  
public interface UserMapperI{  
// 使用@insert 注解指明add方法要执行的向数据库添加一个用户信//息 只要求添加用户姓名和年龄  
@Insert(“insert into users(name,age) values(\_\_1\_\_,\_\_\_2\_\_\_)”)  
Public int add(User user) throws SqlExcetption;  
 // 根据id删除用户的信息  
 @Delete(“delete from users where id=(\_\_3\_\_)  
Public int deleteById(int id) throws SqlExcetption;  
 // 修改一个用户的信息  
 @update(“（ 4 ） users set name=#{name},age=#{age},where id= #{id}”)  
 Public int ModidfyUser(User user) throws SqlExcetption;  
}  
public class User{  
 private int id,  
 private String name,   
 private String age;  
 此处省略getter setter 方法和构造方法  
}

参考答案：#{name} #{age} #{id} update

23.Mybatis动态sql可以让我们在Xml映射文件内，以（1）的形式编写动态sql，完成逻辑判断和（2）的功能.

参考答案：标签 动态SQL拼接

24.Mybatis进行一对一和一对多关联查询时使用的标签分别是（1）和（2）

参考答案：association collection

25.@RequestMapping注解选项value是一个（1）类型，表示一个控制器可以映射多个请求名称。

参考答案：String[]