MyBatis现在主流的写法就是两个xml文件（mybatis-conf.xml和mapper.xml）和一个接口（供数据调用的口）构成，这样写的好处是把持久层和代码分隔开来了，一下是这三个代码部分详细（可以作为模板使用）

1、 myBatisConfig.xml(全局配置文件)

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <properties resource="jdbc.properties"></properties>
    <environments default="development">
    <!-- environment这里可以配置多个环境，如开发的环境、测试的环境 -->
        <environment id="development">
            <transactionManager type="JDBC" />
            <dataSource type="POOLED">
                <property name="driver" value="${driver}" />
                <property name="url" value="${url}" />
                <property name="username" value="${username}" />
                <property name="password" value="${password}" />
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <mapper resource="userMapper.xml" />
        <!-- <mapper class="daiwei.test.myInterface.UserMapper"/> -->
    </mappers>
</configuration>
```

environment：数据源环境　mapper resource：　mapper.xml文件位置　mapper class：mapper接口和mapper.xml文件在同一目录下且俩个文件同名，mapper class为接口位置

2、mapper,xml

```
<?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">
<mapper namespace="daiwei.test.myInterface.UserMapper">
    <select id="getUserById" resultType="daiwei.test.pojo.User">
        select * from user where id = #{id}
    </select>
```

```
</mapper>
```

namespace: 接口的全路径名　id：该方法对应的函数名　resultType：返回的类型

3、userMapper.java

```
package daiwei.test.myInterface;

import daiwei.test.pojo.User;

public interface UserMapper {

    public User getUserById(Integer id);

}
```

4、Test类（调用mybatis的类）

```
package daiwei.test.mybatisTest;

import java.io.IOException;
import java.io.InputStream;
import java.net.URL;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;
import org.apache.log4j.chainsaw.Main;
import org.junit.Test;

import daiwei.test.myInterface.UserMapper;
import daiwei.test.pojo.User;

public class MyTest {

    public static void main(String[] args) throws Exception {
        String path = 
Thread.currentThread().getContextClassLoader().getResource("").getPath();
        System.out.println(path);
        String resource = "MybatisConfig.xml";
        InputStream inputStream = Resources.getResourceAsStream(resource);
        SqlSessionFactory sqlSessionFactory = new
SqlSessionFactoryBuilder().build(inputStream);
        SqlSession openSession = sqlSessionFactory.openSession();
        try {
//          查询
//          UserMapper mapper = openSession.getMapper(UserMapper.class);
//          User user = mapper.getUserById(1);
//          System.out.println(user.toString());
//
//          添加
            UserMapper mapper = openSession.getMapper(UserMapper.class);
            User user = new User("daiwei", 21, 1);
            mapper.insertUserById(user);
            System.out.println(user.getId());
            openSession.commit();

//          更新
//          UserMapper mapper = openSession.getMapper(UserMapper.class);
//          User user = new User(1,"daiwei", 21, 1);
//          mapper.updateUserById(user);
//          openSession.commit();

//          删除
//          UserMapper mapper = openSession.getMapper(UserMapper.class);
//          boolean deleteUserById = mapper.deleteUserById(1);
//          System.out.println(deleteUserById);
//          openSession.commit();

        } finally {
            openSession.close();
        }
    }
}
```