第5章 初识MyBatis框架

湖南科技大学
计算机科学与工程学院

学习目标/Target

- 了解框架的概念和优点
- 了解MyBatis框架的概念和优点
- 掌握MyBatis环境搭建
- 掌握MyBatis入门程序的编写
- 熟悉MyBatis工作原理

章节概述/ Summary

实际开发中，随着业务的发展，软件系统变得越来越复杂，如果所有的软件都从底层功能开始开发，那将是一个漫长而繁琐的过程。此外，团队协作开发时，由于没有统一的调用规范，系统会出现大量的重复功能的代码，给系统的二次开发和维护带来不便。为解决上述问题，框架应运而生。框架实现了很多基础性的功能，开发人员不需要关心底层功能操作，只需要专心地实现所需要的业务逻辑，大大提高了开发人员的工作效率。当前市场上的Java EE开发主流框架有Spring、SpringMVC和Mybatis等，本章主要对框架的概念以及Mybatis的基础知识进行介绍。

目录/Contents

5.1 初识框架

1.1.1 框架概述

- 了解框架的概念，能够知道框架是用于做什么的

5.1.1 框架概述

什么是框架

“框架（Framework）”一词最早出现在建筑领域，指的是在建造房屋前期构建的建筑骨架。在编程领域，框架就是应用程序的骨架，开发人员可以在这个骨架上加入自己的东西，搭建出符合自己需求的应用系统。

框架结构

5.1.1 框架概述

软件框架

软件框架是一种通用的、可复用的软件环境，它提供特定的功能，促进软件应用、产品和解决方案的开发工作。软件框架会包含支撑程序、编译器、代码、库、工具集以及API，它把所有这些部件汇集在一起，以支持项目或系统的开发。

软件框架可以形象地比喻成我们在盖楼房时，用梁+柱子+承重墙搭建起来的钢筋混凝土结构的建筑框架，它是整个建筑的骨架。而实现的软件功能，也就像在这个建筑框架中所要实现的不同类型、功能的房子，比如健身房、商场、酒店、饭店等。

5.1.2 框架的优势

- 了解框架的优势，能够说出框架的优势有哪些

5.1.2 框架的优势

早期Java EE开发弊端

在早期Java EE应用开发中，企业开发人员是利用 JSP+Servlet技术 进行软件应用和系统开发的，使用该技术会有以下两个 弊端。

（1）软件应用和系统可维护性差

如果全部采用JSP+Servlet技术进行软件的开发，因为分层不够清晰，业务逻辑的实现无法单独分离出来，从而造成系统后期维护困难。

（2）代码重用性低

企业希望以最快的速度，开发出最稳定、最实用的软件。如果系统不使用框架，每次开发系统都需要重新开发，需要投入大量的人力物力，并且重新开发的代码可能具有更多的漏洞，这就增加了系统出错的风险。

5.1.2 框架的优势

框架优势

相比于使用JSP+Servlet技术进行软件开发，使用框架有以下优势。

1．提高开发效率：如果采用成熟、稳健的框架，那么一些通用的基础工作，如事务处理、安全性、数据流控制等都可以交给框架处理，程序员只需要集中精力完成系统的业务逻辑设计，降低了开发难度。

2．提高代码规范性和可维护性：当多人协同进行开发时，代码的规范性和可维护性就变得非常重要。成熟的框架都有严格的代码规范，能保证团队整体的开发风格统一。

3．提高软件性能：使用框架进行软件开发，可以减少程序中的冗余代码。例如，使用Spring框架开发时，通过Spring的IOC特性，可以将对象之间的依赖关系交给Spring控制，方便解耦，简化开发；使用MyBatis框架开发时，MyBatis提供了XML标签，支持动态的SQL，开发人员无需在类中编写大量的SQL语句，只需要在配置文件中进行配置即可。

5.1.3 当前主流框架

- 了解当前主流框架，能够说出当前常用的框架有哪些

5.1.3 当前主流框架

Spring框架

Spring是一个开源框架，是为了解决企业应用程序开发复杂性而创建的，其主要优势之一就是分层架构。Spring提供了更完善的开发环境，可以为POJO（Plain Ordinary Java Object，普通Java对象）对象提供企业级的服务。

5.1.3 当前主流框架

Spring MVC框架

Spring MVC是一个Web开发框架，可以将它理解为Servlet。在MVC模式中，Spring MVC作为控制器（Controller）用于实现模型与视图的数据交互，是结构最清晰的。

Spring MVC框架采用松耦合、可插拔的组件结构，具有高度可配置性，与其他的MVC框架相比，具有更强的扩展性和灵活性。

5.1.3 当前主流框架

MyBatis框架

MyBatis 是Apache的一个开源项目iBatis，2010年这个项目由Apache Software Foundation迁移到了Google Code，并且改名为MyBatis ，2013年11月MyBatis又被迁移到Github。

MyBatis是一个优秀的持久层框架，它可以在实体类和SQL语句之间建立映射关系，是一种半自动化的ORM（Object/Relation Mapping，即对象关系映射）实现。MyBatis封装性要低于Hibernate，但它性能优越、简单易学，在互联网应用的开发中被广泛使用。

5.1.3 当前主流框架

Spring Boot 框架

Spring Boot 框架是 Pivotal 团队基于 Spring 开发的全新框架，其设计初衷是为了简化 Spring 的配置，使用户能够构建独立运行的程序，提高开发效率。

Spring Boot 框架本身并不提供 Spring 框架的核心特性及扩展功能，它只是用于快速、敏捷地开发新一代基于 Spring 框架的应用，同时它还集成了大量的第三方类库（如Jackson、JDBC、Redis 等），使用户只需少量配置就能完成相应功能。

5.1.3 当前主流框架

Spring Cloud 框架

Spring Cloud 是一系列框架的有序集合，为开发人员构建微服务架构提供了完整的解决方案，它利用Spring Boot 的开发便利性巧妙地简化了分布式系统的开发。例如，配置管理、服务发现、控制总线等操作，都可以使用 Spring Boot 做到一键启动和部署。可以说，Spring Cloud 将 Spring Boot 框架进行了再封装，屏蔽掉了复杂的配置和实现原理，具有简单易懂、易部署和易维护等特点。

5.2 MyBatis介绍

5.2.1 传统JDBC的劣势

- 熟悉传统JDBC的劣势，能够说出传统JDBC的劣势有哪些

5.2.1 传统JDBC的劣势

JDBC的劣势

JDBC是Java程序实现数据访问的基础，JDBC的劣势主要有以下几个方面。

（1）数据库连接创建、释放频繁会造成系统资源浪费，从而影响系统性能。

（2）SQL语句在代码中硬编码，造成代码不易维护。在实际应用的开发中，SQL变化的可能性较大。在传统JDBC编程中，SQL变动需要改变Java代码，违反了开闭原则。

（3）用PreparedStatement向占位符传参数存在硬编码，因为SQL语句的where条件不一定，可能多也可能少，修改SQL需要修改代码，造成系统不易维护。

（4）JDBC对结果集解析存在硬编码（查询列名），SQL变化导致解析代码变化，造成系统不易维护。

5.2.2 MyBatis概述

- 了解MyBatis框架的概述，能够说出MyBatis框架如何解决JDBC编程劣势

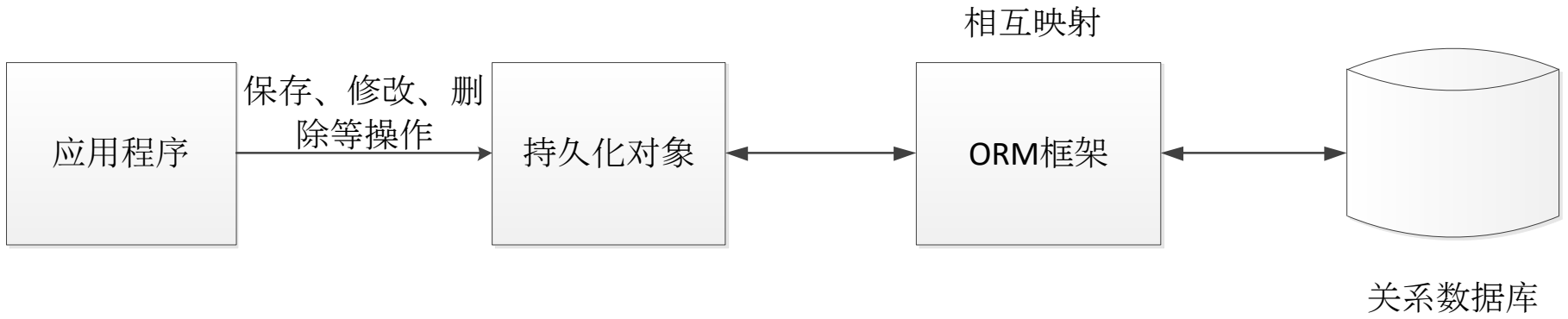
5.2.2 MyBatis概述

什么是MyBatis

MyBatis是一个支持普通SQL查询、存储过程以及高级映射的持久层框架，它消除了几乎所有的JDBC代码和参数的手动设置以及对结果集的检索，使用简单的XML或注解进行配置和原始映射，将接口和Java的POJO映射成数据库中的记录，使得Java开发人员可以使用面向对象的编程思想来操作数据库。

5.2.2 MyBatis概述

MyBatis框架是一个ORM（Object/Relation Mapping，即对象关系映射）框架。所谓的ORM就是一种为了解决面向对象与关系型数据库中数据类型不匹配的技术，它通过描述Java对象与数据库表之间的映射关系，自动将Java应用程序中的对象持久化到关系型数据库的表中。ORM框架的工作原理可以通过一张图来展示。



5.2.2 MyBatis概述

解决JDBC编程劣势

针对JDBC编程的劣势，MyBatis提供了以下解决方案，具体如下。

问题一：数据库链接创建、释放频繁会造成系统资源浪费，从而影响系统性能。

解决方案：在SqlMapConfig.xml中配置数据链接池，使用连接池管理数据库链接。

问题二：SQL语句在代码中硬编码，造成代码不易维护。在实际应用的开发中，SQL变化的可能较大。在传统JDBC编程中，SQL变动需要改变Java代码，违反了开闭原则。

解决方案：MyBatis将SQL语句配置在MyBatis的映射文件中，实现了与Java代码的分离。

5.2.2 MyBatis概述

解决JDBC编程劣势

问题三：使用preparedStatement向占位符传参数存在硬编码，因为SQL语句的where条件不一定，可能多也可能少，修改SQL需要修改代码，造成系统不易维护。

解决方案：MyBatis自动将Java对象映射至SQL语句，通过Statement中的parameterType定义输入参数的类型。

问题四：JDBC对结果集解析存在硬编码（查询列名），SQL变化导致解析代码变化，造成系统不易维护。

解决方案：MyBatis自动将SQL执行结果映射至Java对象，通过Statement中的resultType定义输出结果的类型。

5.3 MyBatis环境搭建

5.3 MyBatis环境搭建

- 掌握MyBatis环境的搭建，能够熟练搭建MyBatis开发环境

5.3 MyBatis环境搭建

MyBatis环境搭建的步骤

使用MyBatis框架进行数据库开发之前，需要先搭建MyBatis环境，MyBatis环境搭建主要有如下基本步骤。

（1）创建工程；

（2）引入相关依赖；

（3）数据库准备；

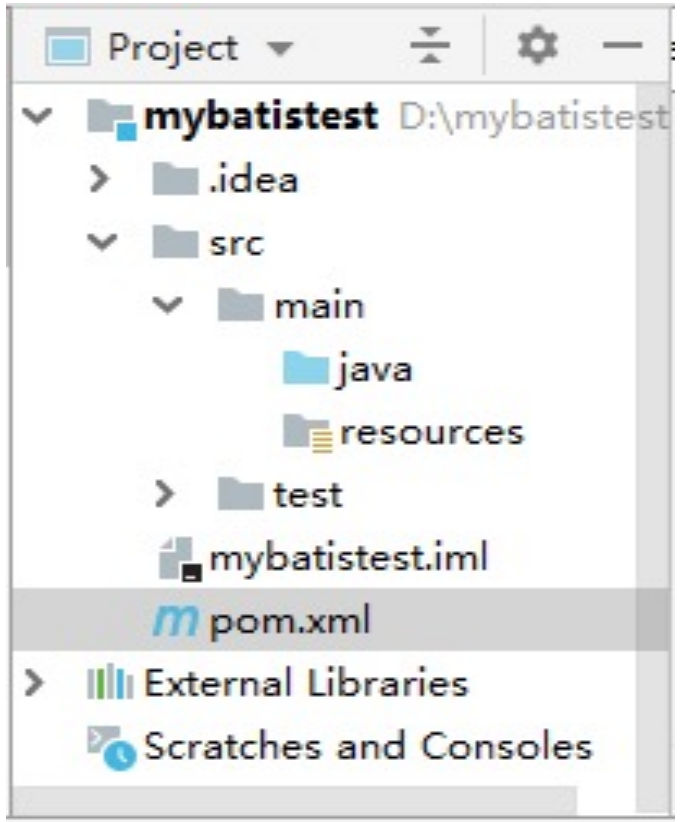
（4）编写数据库连接信息配置文件；

（5）编写核心配置文件和映射文件。

5.3 MyBatis环境搭建

STEP 01

创建工程：在IntelliJ IDEA中，创建名称为mybatistest的Maven工程



5.3 MyBatis环境搭建

STEP 02

引入相关依赖：由于本项目要连接数据库以及对程序进行测试，所以需要在项目的pom.xml文件中导入MySQL驱动包、Junit测试包、MyBatis的核心包等相关依赖。

```
<!-- 只展示了其中一个依赖-- >
<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.11</version>
    </dependency>
...
</dependencies>
```

5.3. MyBatis环境搭建

首次引依赖需要联网

IDEA默认集成的Maven，所以在第一次引入依赖时，需要在联网状态下进行，且引入依赖需要较长时间，耐心等待到依赖引入完成即可。

STEP 03

创建数据库：在MySQL中创建一个名称为mybatis的数据库，具体SQL语句如下。

```
create database mybatis;
```

STEP 04

创建数据库连接信息配置文件：在项目的src/main/resources目录下创建数据库连接的配置文件，这里将其命名为db.properties，在该文件中配置数据库连接的参数。

```
mysql.driver=com.mysql.cj.jdbc.Driver
mysql.url=jdbc:mysql://localhost:3306/mybatis?serverTimez
one=UTC&
characterEncoding=utf8&useUnicode=true&useSSL=false
mysql.username=root
mysql.password=root
```

5.3 MyBatis环境搭建

STEP 05

创建MyBatis的核心配置文件：在项目的src/main/resources目录下创建MyBatis的核心配置文件，该文件主要用于项目的环境配置，如数据库连接相关配置等。核心配置文件可以随意命名，但通常将其命名为mybatis-config.xml。

```
<configuration><properties resource="db.properties"/>
<environments default="development">
<environment id="development">
        <transactionManager type="JDBC"/>
         <dataSource type="POOLED">
              <property name="driver" value="${mysql.driver}" />
              <property name="url" value="${mysql.url}" />
              <property name="username" value="${mysql.username}" />
              <property name="password" value="${mysql.password}" />
         </dataSource>
    </environment></environments>
</configuration>
```

5.4 MyBatis入门程序

5.4 MyBatis入门程序

- 掌握MyBatis入门程序，能够手动编写MyBatis程序

5.4 MyBatis入门程序

上一节完成了MyBatis环境的搭建，本节将在MyBatis环境下实现一个入门程序来演示MyBatis框架的使用，该程序要求实现根据id查询用户信息的操作。

5.4 MyBatis入门程序

STEP 01

数据准备：在mybatis数据库中创建users表，并在users表中插入几条数据。

```
use mybatis;
create table users(
    uid int primary key auto_increment,
    uname varchar(20) not null,
    uage int not null
);
insert into users(uid,uname,uage) values(null,'张三',20),(null,'李四',18);
```

5.4 MyBatis入门程序

STEP 02

创建POJO实体：在项目的src/main/java目录下创建com.itheima.pojo包，在com.itheima.pojo包下创建User类，该类用于封装User对象的属性。

```
package com.itheima.pojo;
public class User {
    private int uid;                // 用户id
    private String uname;       // 用户姓名
    private int uage;              // 用户年龄
    // 省略getter/setter方法
    ...
}
```

5.4 MyBatis入门程序

STEP 03

创建映射文件UserMapper.xml：在项目的src/main/resources目录下创建一个文件夹，在mapper文件夹下创建映射文件UserMapper.xml，该文件主要用于实现SQL语句和Java对象之间的映射，使SQL语句查询出来的关系型数据能够被封装成Java对象。

```
<mapper namespace="com.itheima.pojo.User">
    <!--id ="接口中的方法名"parameterType="传入的参数类型"
      resultType = "返回实体类对象，使用包.类名"-->
<select id="findById" parameterType="int"
            resultType="com.itheima.pojo.User">
            select * from users where uid = #{id}
</select>
</mapper>
```

5.4 MyBatis入门程序

STEP 04

修改mybatis-config.xml配置文件：在mybatis-config.xml映射文件中添加UserMapper.xml映射文件路径的配置，用于将UserMapper.xml映射文件加载到程序中。

```
<!– mapping文件路径配置-->
<mappers>
    <mapper resource="mapper/UserMapper.xml"/>
</mappers>
```

一个项目多个配置文件

如果一个项目有多个映射文件，则mybatis-config.xml核心配置文件中需要在<mappers>元素下配置多个<mapper>元素指定映射文件的路径。

5.4 MyBatis入门程序

STEP 05

编写测试类：在项目的src/test/java目录下创建Test包，在Test包下创建UserTest类，该类主要用于程序测试。

```
public class UserTest {
    public void userFindByIdTest() {
        String resources = "mybatis-config.xml"; Reader reader=null;
        try { reader= Resources.getResourceAsReader(resources);
        } catch (IOException e) { e.printStackTrace();}
    SqlSessionFactory sqlMapper=new
            SqlSessionFactoryBuilder().build(reader);
    SqlSession session=sqlMapper.openSession();
    User user=session.selectOne("findById",1);
    System.out.println(user.getUname());
    session.close();}}
```

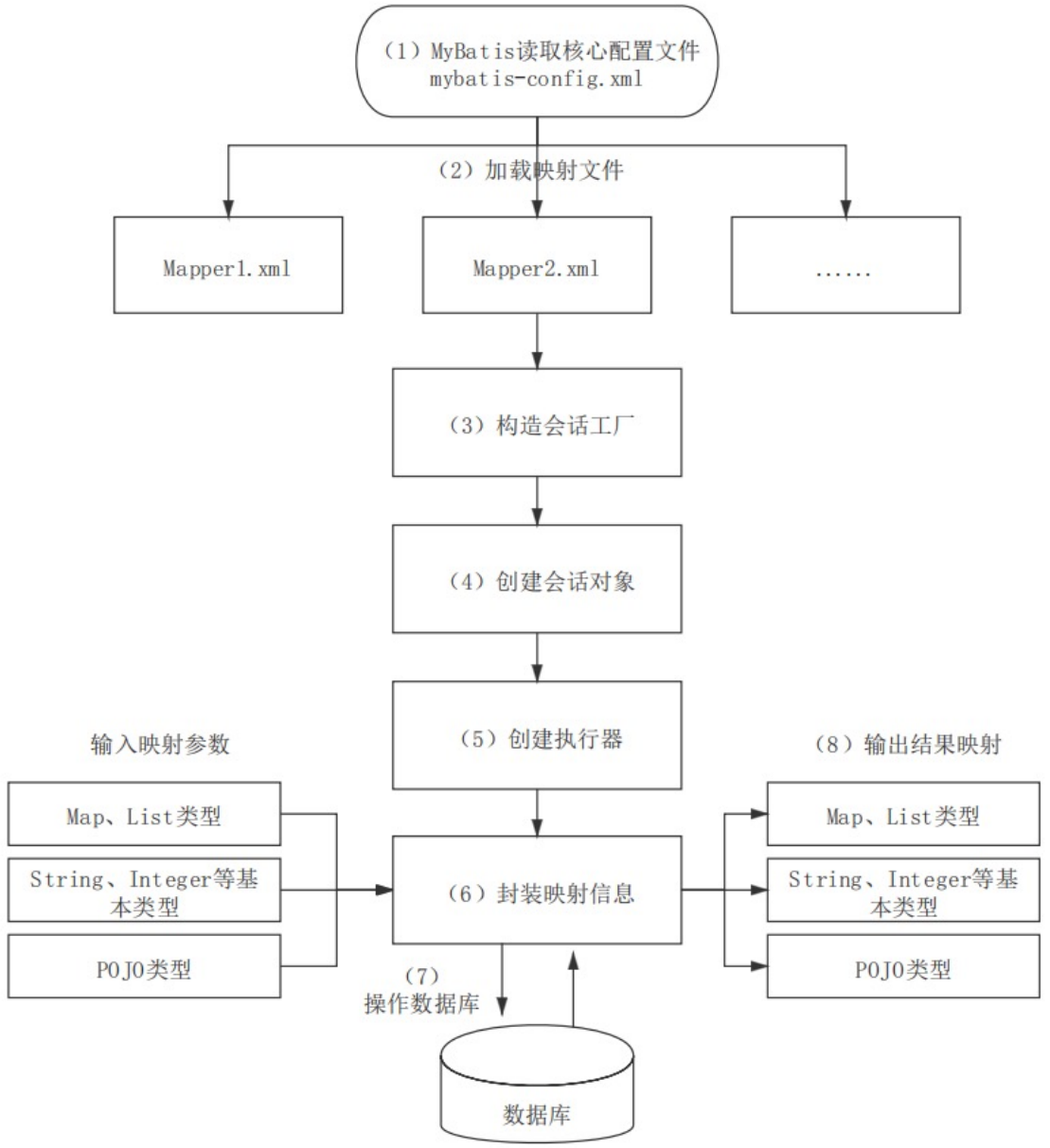
5.5 MyBatis工作原理

5.5 MyBatis工作原理

- 熟悉MyBatis工作原理，能够理解并说出MyBatis的工作原理

5.5 MyBatis工作原理

MyBatis工作原理图



5.5 MyBatis工作原理

MyBatis工作原理步骤

MyBatis框架在操作数据库时，大体经过了8个步骤。下面结合MyBatis工作原理图对每一步流程进行详细讲解，具体如下。

（1）MyBatis读取核心配置文件mybatis-config.xml：mybatis-config.xml核心配置文件主要配置了MyBatis的运行环境等信息。

（2）加载映射文件Mapper.xml：Mapper.xml文件即SQL映射文件，该文件配置了操作数据库的SQL语句，需要在mybatis-config.xml中加载才能执行。

（3）构造会话工厂：通过MyBatis的环境等配置信息构建会话工厂SqlSessionFactory，用于创建SqlSession。

5.5 MyBatis工作原理

MyBatis工作原理步骤

（4）创建会话对象：由会话工厂SqlSessionFactory创建SqlSession对象，该对象中包含了执行SQL语句的所有方法。

（5）创建执行器：会话对象本身不能直接操作数据库，MyBatis底层定义了一个Executor接口用于操作数据库，执行器会根据SqlSession传递的参数动态的生成需要执行的SQL语句，同时负责查询缓存地维护。

（6）封装SQL信息：SqlSession内部通过执行器Executor操作数据库，执行器将待处理的SQL信息封装到MappedStatement对象中。

5.5 MyBatis工作原理

MyBatis工作原理步骤

（7）操作数据库：根据动态生成的SQL操作数据库。

（8）输出结果映射：执行SQL语句之后，通过MappedStatement对象将输出结果映射至Java对象中。

本章小结

本章主要针对MyBatis框架进行了讲解。首先对框架的概念、优点和当前一些主流的Java EE框架进行了讲解，然后对传统JDBC的劣势进行了分析，由此引出了MyBatis框架，并对MyBatis框架的环境搭建、入门程序以及工作原理进行了详细地讲解。通过本章的学习，读者可以了解MyBatis框架及其作用，熟悉MyBatis的工作原理，并能够独立完成MyBatis框架环境的搭建以及入门程序的编写。