目录

[1. 引言 2](#_Toc1835808)

[1.1设计背景 2](#_Toc1835809)

[1.2现状分析 3](#_Toc1835810)

[1.3实现功能 3](#_Toc1835811)

[2. 开发环境 3](#_Toc1835812)

[2.1语言环境 3](#_Toc1835813)

[2.2编辑工具 3](#_Toc1835814)

[2.3数据库 3](#_Toc1835815)

[3. Lucky框架的概述 3](#_Toc1835816)

[3.1解决的问题 3](#_Toc1835817)

[3.2使用说明 3](#_Toc1835818)

[3.3框架的扩展和完善 3](#_Toc1835819)

[4. 需求分析 3](#_Toc1835820)

[4.1原生jdbc 3](#_Toc1835821)

[4.2现有的数据持久层框架 3](#_Toc1835822)

[4.3问题总结 3](#_Toc1835823)

[4.4有关如何解决问题的讨论 3](#_Toc1835824)

[5. 总体设计 3](#_Toc1835825)

[5.1对原生jdbc的分析 4](#_Toc1835826)

[5.2结果集自动封装成特定实体类的讨论 4](#_Toc1835827)

[5.3实体类对象信息自动转化未sql语句的讨论 4](#_Toc1835828)

[5.4根据实体类自动建表的讨论 4](#_Toc1835829)

[5.5实现事务处理机制的讨论 4](#_Toc1835830)

[5.6框架日志的控制输出 4](#_Toc1835831)

[5.7框架的整体运行模式 4](#_Toc1835832)

[6. 详细设计 4](#_Toc1835833)

[6.1实现结果集到实体类的自动封装 4](#_Toc1835834)

[6.2实现实体类对象到sql语句的自动生成 4](#_Toc1835835)

[6.3原生jbbc操作的保留与简化 4](#_Toc1835836)

[6.4实现java实体类到MySql表的自动创建 4](#_Toc1835837)

[6.5实现处理事务的机制 4](#_Toc1835838)

[6.6实现框架日志的控制输出 4](#_Toc1835839)

[7. Lucky框架的测试与调试 5](#_Toc1835840)

[7.1简单java项目的测试与调试 5](#_Toc1835841)

[7.2复杂java web项目的测试与调试 5](#_Toc1835842)

[8. 结果分析 5](#_Toc1835843)

[9. 致谢 5](#_Toc1835844)

[10. 参考文献 5](#_Toc1835845)

[11. 附录 5](#_Toc1835846)

# 引言

互联网已经成为我们生活学习中不可或缺的一部分，各种功能强大的软件的诞生为我们的生活带来了极大的便利，甚至改变了我们的生活方式。然而在这些软件中大部分的开发过程中必定都会有一个不可缺少的环节，那就是数据库，数据库好比人的大脑的记忆系统，没有了数据库就没有了记忆系统。计算机也就不会如此迅速的发展。数据库的应用已经深入到生活和工作的方方面面。数据库的发展体现了一个国家信息发展的水平，由此可见数据库的重要性了。在Java语言在中用于操作数据库的最常用方式是JDBC。JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序 ，但是原生的JDBC使用起来也并不是那么方便，所以市面上出现了很多的基于JDBC的数据持久层框架 如：Hibernate，MyBatis等等。这些框架的出现大大简化了操作数据库的难度，也很大程度上提高了开发人员的开发效率，本文将讨论由本人开发的针对MySQL数据库的数据持久层框架Lucky的设计实现以及测试过程。

## 1.1设计背景

在如今的

## 1.2现状分析

## 1.3实现功能

# 开发环境

## 2.1语言环境

Java是由Sun Microsystems公司于1995年5月推出的Java面向对象程序设计语言和Java平台的总称。由James Gosling和同事们共同研发，并在1995年正式推出。Java语言是一种优秀的编程语言。它最大的优点就是与平台无关，在Windows 9x、Windows NT、Solaris、Linux、MacOS以及其它平台上，都可以使用相同的代码。“一次编写，到处运行”的特点，使其在互联网上广泛采用。Java语言是一种相当简洁的“面向对象”程序设计语言。Java语言省略了C++语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。它更加严谨、简洁。

在大学中学到过种类繁多的语言，但其中最令我感兴趣的还是java,相比C语言，java的面向对象编程更加简洁明了，相比起C++的指针，内存操作和多重继承等java显得更加清爽，比起C#众多的关键字，Java更显优雅。所以本次的开发我选择Java作为项目的编程语言。

## 2.2编辑工具

谈到Java，Eclipses时我们不得不提编辑器，Eclipse是著名的跨平台开源集成开发环境（IDE）。最初主要用来Java语言开发，当前亦有人通过插件使其作为C++、Python、PHP等其他语言的开发工具。Eclipse的本身只是一个框架平台，但是众多插件的支持，使得Eclipse拥有较佳的灵活性，所以许多软件开发商以Eclipse为框架开发自己的IDE。

提到Eclipse就会有一种莫名的亲切感，从第一个java的入门程序“Hello Word”开始到如今即将毕业，Eclipse陪我走过了漫长的大学时光，Eclipse不仅功能强大，而且开源免费，现阶段我将它视为我最好的选择，所以本次的毕设项目我选择Eclipse作为我的编辑器。

# 需求分析

## 4.1原生jdbc

只要提到java操作数据库，我想没人会不知道JDBC吧， JDBC（Java Data Base Connectivity，java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

JDBC是一个很强大的数据库操作工具，但是也不得不说它的操作其实是十分繁琐的，首先需要通过Driver来获得连接对象Connection，然后还需要获得预编译对象PreparedStatement来接受和处理SQL语句，如果是查询操作，还需要一个结果集对象ResultSet来储存查询结果，每执行一次数据库操作我们都需要借助三个类，不便于记忆而且操作复杂。JDBC对非查询操作的处理算是比较完美的，最令人抓狂的是对查询操作的处理，当我们执行完查询操作后JDBC只返回给我们一个结果集对象（ResultSet），想要取出详细的结果我们还需要进行繁琐的遍历操作。JDBC提供的方法不具备扩展性，就拿查询操作来举例：一般情况下，我们数据库中每多一张表时我们都会为其创建对应的实体类。那么就产生了一个很令人头疼的问题：如果想得到实体类形式的查询结果，我们就不得不一遍又一遍的去写“先遍历结果集，后封装数据”的代码，综上所述JDBC是有很大的局限性的，是满足不了当今这个追求效率的时代的，所以我们期待更好的解决方案！

## 4.2现有的数据持久层框架

在众人的期待下Hibernate诞生了并很快的进入了开发人员的视野，2001年，澳大利亚墨尔本一位名为Gavin King的27岁的程序员，上街买了一本SQL编程的书，他厌倦了实体bean，认为自己可以开发出一个匹配对象关系映射理论，并且真正好用的Java持久化层框架，因此他需要先学习一下SQL。这一年的11月，Hibernate的第一个版本发布了。Hibernate是一种Java语言下的对象关系映射解决方案。它是使用GNU宽通用公共许可证发行的自由、开源的软件。它为面向对象的领域模型到传统的关系型数据库的映射，提供了一个使用方便的框架。它的设计目标是将软件开发人员从大量相同的数据持久层相关编程工作中解放出来。无论是从设计草案还是从一个遗留数据库开始，开发人员都可以采用Hibernate。

Hibernate不仅负责从Java类到数据库表的映射（还包括从Java数据类型到SQL数据类型的映射），还提供了面向对象的数据查询检索机制，从而极大地缩短了手动处理SQL和JDBC上的开发时间。Hibernate的诞生大大提高了开发的效率，我们甚至都不用写SQL就可以操作数据库，但是Hibernate也带来了新的问题，我们的确不用写SQL语句了，但是我们有得去学习他的HQL语言，这在无形之中也带来了压力！

Mybaits也是众多解决方案中得一种，Mybaits的前身是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月迁移到Github,目前mybaits是由Github维护的，mybatis是一个java持久层框架，它封装少、高性能·可优化、维护简单等优点成为了目前java移动互联网网站服务的首选持久层框架，它特别适合分布式和大数据网络数据库编程。Mybaits是一个半自动映射的框架，之所以称它为半自动，是因为它需要手工匹配提供POJO、SQL和映射关系，而全表映射的Hibernate只需要提供POJO和映射关系便可。Mybaits功能强大而且十分灵活，但是配置复杂，上手有一定得难度，需要有扎实得数据库功底。

## 4.3问题总结

# Lucky框架的概述

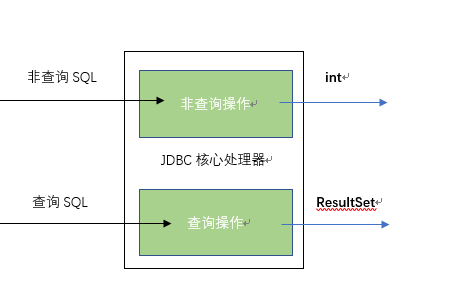
## 3.1解决的问题

## 3.2使用说明

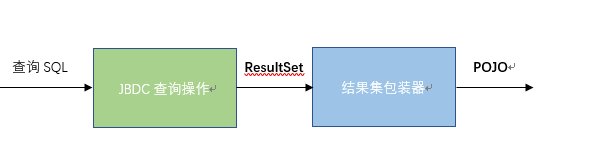
## 3.3框架的扩展和完善

# 总体设计

## 5.1对原生jdbc的分析



## 5.2结果集自动封装成特定实体类的讨论



## 5.3实体类对象信息自动转化未sql语句的讨论

## 5.4根据实体类自动建表的讨论

## 5.5实现事务处理机制的讨论

## 5.6框架日志的控制输出

## 5.7框架的整体运行模式

# 详细设计

## 6.1实现结果集到实体类的自动封装

## 6.2实现实体类对象到sql语句的自动生成

## 6.3原生jbbc操作的保留与简化

## 6.4实现java实体类到MySql表的自动创建

## 6.5实现处理事务的机制

## 6.6实现框架日志的控制输出

# Lucky框架的测试与调试

## 7.1简单java项目的测试与调试

## 7.2复杂java web项目的测试与调试

# 结果分析

# 致谢

# 参考文献

# 附录