论文题目 计算机语言在线考试系统的设计与实现

**姓 名**

**学 院**

**专 业**

**指导教师**

**备 注**

**2017 年 X 月 X 日**

计算机语言在线考试系统的设计与实现

**Design and Implementation of Online Examination Monitoring System**

计算机语言在线考试系统的设计与实现

摘 要

随着互联网发展以及信息技术的普及，计算机的各种应用如雨后春笋纷纷涌现，大大影响和改变了我们的生活和学习方式，冲击着传统的方式。目前随着学生数量的增长、课程的增加以及考试频率的增加，传统考试方式的弊端也日益显现，并限制了教育的发展，因此越来越多的学校和相关部门意识到采用计算机来辅助考试的必要性。在传统考试方式中，教师需要事先进行一系列的准备工作，包括出试卷、印刷试卷、安排考场，同时需要安排教务人员进行监考，考试结束之后，教师需要评判试卷并对考试结果进行统计分析。与此相比，在线考试可以大大简化传统考试的各个环节，教师只需要设定一个试卷模板，也就是规定试题的难度、题型等内容，系统便能够基于该模板自动生成试卷，考试结束后，系统能够评阅试卷并对考试结果进行统计并进行合理的分析，大大提高了考试的效率、公平以及对教学的指导意义。

论文首先明确了课题的背景以及选题的意义，并结合国内外现状总结分析了各种已有的在线考试系统的优点和局限性，并大致确定了论文的研究内容。随后，对系统开发涉及到的技术进行调研，分析相关技术的优劣并确定合适的技术方案和工具。接下来对系统进行需求分析并确定系统的总体结构，给出系统的模块划分，包括用户管理模块、考试管理模块、题库管理模块、试卷模板管理模块、成绩管理模块、系统管理模块。根据系统涉及到的功能模块，确定数据表以及各个表之间的联系。最后采用MVC三层架构，综合利用Struts2，Spring，Hibernate框架，并利用MySQL数据库完成计算机语言的在线考试系统。

本文提出的计算机语言的在线考试系统，能够大大简化教师的工作压力，同时自动组卷功能能够在一定程度下避免考试作弊的现象并通过各种类型的试题来帮助教师更加有效的评判教学效果，而自动评阅试卷功能能够结合测试数据通过率以及语法分析程序匹配等方式来更加合理和公平地评估学生的答题结果，更加客观的评价学生对知识的掌握程度，方便教师进行教学方式和内容的调整。

**关键词：**计算机语言，在线考试系统，自动组卷，自动评阅试卷

**Design and Implementation of Online Examination Monitoring System**

**Abstract**

In recent years, With the development and popularity of Internet and Information technology, a large number of computer applications have emerged one after another, which simplify peoples’ life and learning style. With the increasing number of students, subjects and examinations, the malpractices of traditional examination appear day by day, so the leaders of relevant departments and schools have become conscious of the emergency of the examination system assisted with computer and it is time to develop an online examination system. In traditional manual test, teachers need to make careful preparations, such as test sheeting making, examination arrangements, examination invigilation, examination paper checking and performance management etc. By contrast, online examination system greatly simplify examination preparations and the only thing teachers need to do is to set the templates of exam papers which makes the conversion and rebuilding of exam questions much easier and more flexible. After students finish the tests, system will automatically evaluate test papers and make an accurate and objective statistical analysis about students ’performance. breaking out the traditional mode and making examination more equitable and effective.

Firstly, this thesis elaborated the background and significance of selected topic, and make an analysis about online examination system based on domestic and international research documents, which help to determine the research content of this article. Then, we expatiate current status of the relevant technology accordingly and determine suitable technology and tools by making comparison. Next, by analyzing requirements of examination determines the overall structure of online examination system, which consists of six functional modules, namely, user management, examination management, examination questions management, examination template management, student performance management and system management modules. Base on the functional modules, the structures and relationship of database tables are pointed out. At last, an online examination system based on MVC structure and an integrated framework including Structs2, Spring and Hibernate, using MySQL database is development.

The online examination system proposed in this thesis can greatly lightens the work of teachers and reduces incidents of cheating by auto-generating test paper, evaluates teaching effect effectively by various exam questions. At the same time, automation of checking the examination papers combines the pass rate of test cases and program similarity by syntax analysis, which evaluates teaching effect more effectively and fair, helping adjusting teaching mode and teaching content.

**Keywords:** Programing language, Online examination system, Auto-generating test paper, Auto-checking test paper

目 录

[摘 要 I](#_Toc496539605)

[Abstract II](#_Toc496539606)

[第1章 绪 论 1](#_Toc496539607)

[1.1 选题背景及意义 1](#_Toc496539608)

[1.2 开发目的及意义 1](#_Toc496539609)

[1.3 国内外发展现状 2](#_Toc496539610)

[1.4 论文研究内容 2](#_Toc496539611)

[1.5 论文组织结构 3](#_Toc496539612)

[第2章 关键技术 4](#_Toc496539613)

[2.1 B/S体系结构 4](#_Toc496539614)

[2.2 MVC设计模式 5](#_Toc496539615)

[2.2.1 MVC组成部分介绍 5](#_Toc496539616)

[2.2.2 MVC交互流程 5](#_Toc496539617)

[2.2.3 MVC模式的优点 6](#_Toc496539618)

[2.3 SSH集成框架 6](#_Toc496539619)

[2.3.1 Struts2框架 6](#_Toc496539620)

[2.3.2 Spring框架 8](#_Toc496539621)

[2.3.3 Hibernate框架 9](#_Toc496539622)

[2.4 MySql数据库 9](#_Toc496539623)

[2.5 系统开发环境 10](#_Toc496539624)

[2.6 本章小结 10](#_Toc496539625)

[第3章 需求分析与总体设计 12](#_Toc496539626)

[3.1 可行性分析 12](#_Toc496539627)

[3.2 系统总体需求分析 13](#_Toc496539628)

[3.3 系统功能需求分析 14](#_Toc496539629)

[3.4 用例图分析 14](#_Toc496539630)

[3.4.1 用户管理模块用例分析 15](#_Toc496539631)

[3.4.2 考试管理模块用例分析 16](#_Toc496539632)

[3.4.3 题库管理模块用例分析 17](#_Toc496539633)

[3.4.4 试卷模板管理模块用例分析 18](#_Toc496539634)

[3.4.5 成绩管理模块用例分析 19](#_Toc496539635)

[3.4.6 系统管理模块用例分析 21](#_Toc496539636)

[3.5 数据库需求分析 22](#_Toc496539637)

[3.6 系统总体设计 22](#_Toc496539638)

[3.7 本章小结 24](#_Toc496539639)

[第4章 系统设计与实现 26](#_Toc496539640)

[4.1 数据库设计 26](#_Toc496539641)

[4.2 功能模块详细设计与实现 32](#_Toc496539642)

[4.2.1 系统登录模块详细设计与实现 32](#_Toc496539643)

[4.2.2 学生在线考试模块详细设计与实现 33](#_Toc496539644)

[4.2.3 教师相关功能模块详细设计与实现 34](#_Toc496539645)

[4.3 系统关键模块设计与实现 35](#_Toc496539646)

[4.3.1 自动组卷模块设计与实现 35](#_Toc496539647)

[4.3.2 自动评阅试卷设计与实现 37](#_Toc496539648)

[4.4 本章小结 44](#_Toc496539649)

[第5章 总结与展望 45](#_Toc496539650)

[参考文献 47](#_Toc496539651)

[致 谢 49](#_Toc496539652)

# 绪 论

## 选题背景及意义

随着计算机技术和互联网技术的迅猛发展，各种相关的应用如雨后春笋纷纷涌现，颠覆了传统的方式，大大影响了人们的生活和工作方式。在这种新的发展形势下，社会对教育提出了更高的要求，计算机基础课程已经成为中国高校学生的必须课程，不仅涉及到计算机相关专业的学生，也是非计算机专业学生的公共基础课程，旨在现有技术背景和社会需要下，培养学生应用计算机来认知和解决问题的能力，为学生熟悉信息化社会中的各项基本应用，并适应社会的需要以及未来的发展奠定良好的基础。现代教育存在的一个普遍的问题是学生数量掌握了理论知识，而用已掌握的理论知识应用于实际问题的能力还很薄弱。因此如何有效地考察学生对知识的掌握程度和运用能力是高校和计算机基础课程教师关注的重点问题。相比于其他课程，计算机基础课程更加侧重于学生的动手实践能力以及举一反三的能力，因此采用在线考试的方式是一种必然选择。传统的计算机课程考试方式存在很多弊端，主要体现在以下几点：

一、采用纸质试卷的形式，仅仅能够考察学生对理论知识的掌握程度，不能考察学生的编程和实践能力，无法真实衡量学生对课程掌握的真实程度。

二、大量考试前的准备工作。教师需要事先准备准备考题并进行试卷印刷，浪费了大量的资源。同时需要安排考场、监考人员以及考试时间。

三、需要人工进行阅卷，耗费大量的人力。对于主观题，不同教师可能有不同的评分标准。为了了解学生对不同知识点的统计程度，教师需要统计不同试题的得分以及评判整个试卷的难易程度以便调节后续考试的难易程度。

四、考试不公平性。由于不同考场的纪律以及管理程度不同，考试存在不公平的现象。

因此我们需要采用在线考试的方式来取代传统的考试方式，将信息化技术应用于教育领域，改良传统教育模式，提高教学效率、教学效果以及科研管理水平。在线考试能够有效地解决传统考试的问题，大大简化了考试的过程，考生能够随时随地参与考试，还可以保证考试的公平性，同时方便了教师管理考试内容和评估考生成绩。

## 开发目的及意义

本论文的主要研究内容是通过借助计算机技术来开发一种在线考试系统来代替传统纸笔的考试模式。目前已有的在线开发系统都是为了方便考核学生的学习效果而开发的，但是这些平台大多针对以客观题为主的课程的考核。对于计算机课程的考核更加侧重于学生的编程实践能力而不仅仅局限于理论知识的考察，而本文所涉及到的在线考试系统的设计与实现目的，正是为了适应计算机课程的需要，不仅提供理论知识的考察，还有对学生的编程能力进行考察。

## 国内外发展现状

随着时代的发展，考核方式也发生了一些变化。过去高校的学生规模相对较少并且计算机基础课程的考察比较简单，传统的考试方式的局限性并不明显。随着高校招生规模的逐渐扩张以及计算机课程成为公共必修课程之后，参与计算机基础课程的学生人数日益增多，传统考试在计算机基础考试上弊端日益明显。为了适应新的发展需要，在线考试逐渐取代传统考试方式。相比传统考试方式，在线考试方式能够大幅度减少出题、监考、评阅、统计等考试工作[1] ，大大提高了学校的工作效率。早在上世纪70年代，美国考试委员会率先从事计算机模式考试的研究工作，并于80年代研发了第一个在线考试系统。目前专业提供在线教育以及测评的机构越来越多，其中美国著名的考试机构思而文学习系统有限公司专门为各大机构提供在线教育以及考试的服务，许多授证机构以及专业人员的认证考试都依托于该机构。相比之下，国内的在线考试起步相对较晚，从2004年开始，大学英语四六级以及计算机等级考试开始从人工考试方式向计算机考试方式转型。直至2006年，在线考试系统已经发展为考试的主流形式。截止到2016年，全世界141个国家，有超过25中不同语言通过在线考试的方式提供了2400多种考试，涉及到专业考试、学校教学效果检测以及企业考核等多个领域，考试能够随时随地举行，受时间地域的限制越来越小，逐渐成为一种学习方式，方便查漏补缺[2] 。

计算机基础课程考试主要包含两大部分，第一部分是理论知识考察，主要通过选择题、判断题等客观题。第二部分是编程部分，主要是通过写代码的方式进行考察。目前比较成熟的在线考试系统都是由专门的机构开发，虽然能够适合很多课程的考试需要，但是对于学校而言，采用这类的考试系统会存在很多问题。第一，学校考试相对比较频繁，考试费用或者软件开发费用过高。第二，这类系统大部分是基于标准化考试形式，能够很好的完成客观题的考核，难以对主观题进行判定，尤其是对计算机基础类课程的考核存在很大的限制。

## 论文研究内容

对于计算机基础课程，不仅要考核学生对理论知识的掌握情况还要考核学生编程实践能力。而目前已有的在线考试主要是针对客观题的考核，在客观题的评判上存在很大的问题，因此本文主要研究如何借助计算机以及互联网技术来设计和开发一种针对计算机基础课程的在线考试系统。为了更好的了解以及实现该系统，首先需要分析计算机基础课程的考核内容以及考察形式，确定系统的实际功能需求并进行业务建模。然后通过用例图以及数据流图对系统需求进行进一步的分析。最后针对相关的需求进行整体和详细的设计并确定相关功能实现所需的技术以及相关的算法。根据实际调研的需求，确定整个系统的架构，并完成学生、教师以及管理员的权限与角色管理、题目录入、考生考试验证登陆、智能组卷、防作弊、考试计时、试卷评分、试后统计分析等功能模块。

## 论文组织结构

本文主要是介绍计算机基础的在线考试系统的设计与实现，并从需求分析、系统设计、实现等方便依次展开。论文整体结构如下：

第1章，绪论。通过对比在线考试与传统考试的差异，分析在线考试的有点以及在线考试的意义。接下来分析了在线考试在国内外的研究现状，并确定本文的研究内容以及相关的功能模块。

第2章，相关技术。介绍在线考试系统的理论基础，以及该系统所涉及的技术。主要涉及到B/S体系结构的相关介绍，以及关键技术、框架，和数据库等内容，最后是本系统的开发环境的介绍。

第3章，需求分析和总体设计。对系统进行可行性分析和需求分析，并通过用例图进行系统建模，确定系统的功能模块，最后给出了系统的总体设计。

第4章，系统设计与实现。首先是数据库中表格的详细设计并通过ER图对数据库进行建模，确定数据表之间的联系。然后对系统中的功能模块进行详细设计，并通过流程图表述功能模块的业务流程，并给出每个功能模块的界面实现。最后是对系统的关键模块进行详细的说明，主要是自动组卷和自动评阅试卷模块并给出了相关的核心代码实现。

第5章，总结与展望。对本文的工作进行总结，主要是本文所涉及的系统的优缺点，以及有待改进的方面以及后续的工作。

# 关键技术

在上一章中，主要叙述了在线考试系统相比传统的纸笔考试方式的优越性，并分析了现有的在线考试系统的局限性，从而提出了针对计算机课程的在线考试系统。本文将会利用javaEE的技术来开发针对计算机语言类课程的在线考试系统，因此本章将会对该系统使用到的相关技术进行说明，了解为什么使用MVC设计模式，B/S架构，以及SSH（Struct、Spring、Hibernate）框架、MySQL数据库，最后对系统的开发环境进行简要说明。

## B/S体系结构

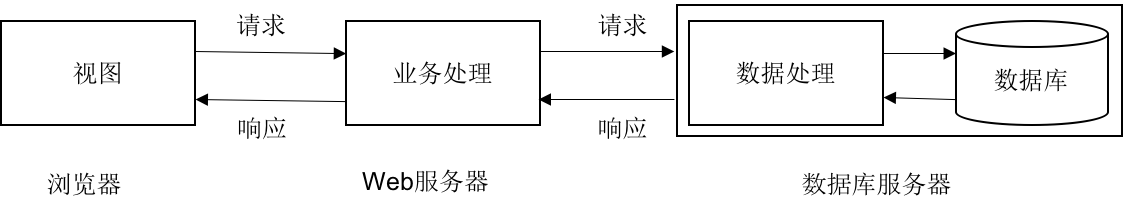
B/S体系结构的全称是浏览器/服务器结构，它是在C/S（客户端/服务器）体系结构的基础上发展而来的，并采用HTTP作为传输协议。用户通过浏览器向服务器端发送请求，服务器从数据库中获取数据并按照请求进行业务逻辑处理，然后服务器端将处理后的结果返回给服务器，最后浏览器解析返回的结果并将结果呈现给用户[3] 。相比C/S结构而言[4] ，B/S结构不需要额外开发客户端软件而是直接使用浏览器，大大提高了可扩展性以及可维护性，同时降低了开发和维护的成本。

图 1 B/S体系结构

B/S体系结构包含三个部分，浏览器、web服务器和数据库服务器，如图 1所示。首先是浏览器部分，承担视图显示部分的工作。负责接受用户的输入并封装为Http协议的格式，同时也负责将服务端返回的结果进行解析形成HTML格式的文件显示给用户。然后是Web服务器部分，主要承担业务流程以及业务处理逻辑任务。浏览器通过Http协议将用户的请求发送给Web服务器端，Web服务器接受到请求之后，与数据库服务器建立连接，并通过SQL语句的方式提交数据处理请求。等数据库服务器端执行完数据库请求之后，将执行的结果再返回给Web服务器端，最后Web服务器端将结果传送给浏览器。最后是数据库服务器部分，主要完成数据处理任务，执行Web服务器发送的SQL语句的数据处理任务，从而完成对数据库中数据的增删改查等操作，并将执行的结果返回给Web服务器。

## MVC设计模式

MVC模式[5] 为model-view-controller的缩写，与80年代提出的一种软件设计典范，包含模式、视图和控制器三个逻辑组件，将业务逻辑、数据、界面显示进行分离的方式来组织代码，这三个逻辑组件以松耦合的方式协同工作，极大提高了应用程序的可扩展性和可维护性，并受到广泛的应用。后来被Sun公司推荐为JavaEE平台的设计模式，成为面向对象程序设计语言遵守的规范。

### MVC组成部分介绍

MVC模式按照用户输入、外部世界建模以及可视化反馈进行划分，并分别使用模型、视图和控制器来进行处理。下面依次对三个部分进行描述。

模型(Model)用于处理数据，视图部分会发出一些数据请求，模型会根据请求作出回应。当控制器发出请求该别状态时，模型部分就会作出相应的处理。整体来看，模型部分是MVC的核心。

视图(View)是与用户与应用程序交互的界面。负责展示模型返回的数据同时也能够接受用户的输入，但并不能够处理独立处理数据。

控制器(Controller)是模型与视图之间的连接器，负责从不同视图接受请求，并转发给对应的模型，并根据模型返回的结果选择视图来相应用户的请求。

### MVC交互流程

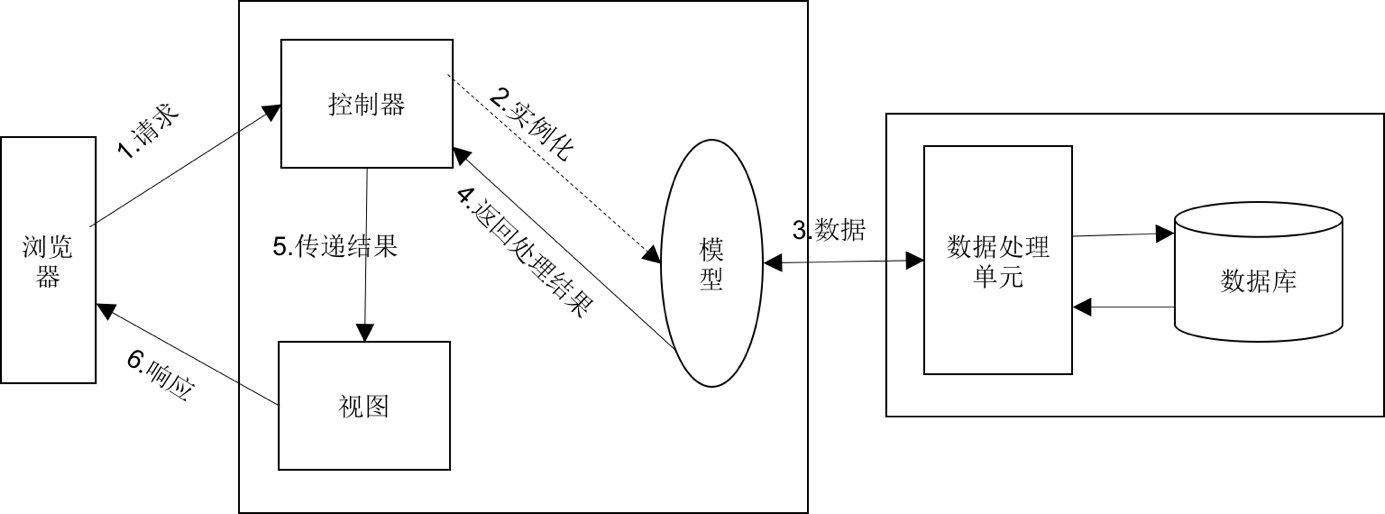
MVC中的各个组成部分之间的交互流程如图 2所示，用户发出请求，控制器根据请求来选择合适的模型进行处理，然后模型根据用户的请求进行相应的业务逻辑处理，并返回用户请求的数据，最后控制器将返回的相应数据传递给对应的视图，并由视图部

图 2 MVC标准化流程

分将数据最终呈现给用户。

### MVC模式的优点

MVC模式能够将视图与逻辑处理单元分离，能够非常容易的改变视图而不必改变后端的逻辑处理，并且能够实现视图的动态添加和删除，同时不同模块的分离和单独开发，增加了程序的可扩展性和可维护性[6] 。控制器能够连接不同的视图和模型，以适应用户的需求，增加了程序的可重用性。目前大部分web应用程序是基于过程化语言进行开发，将页面与数据库访问的代码混杂在一起，大大增加了维护的难度。

## SSH集成框架

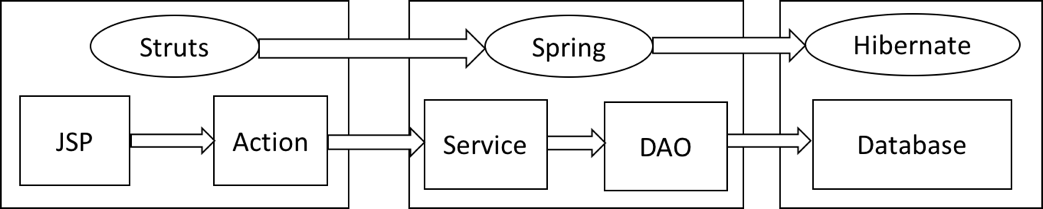
SSH是一种集成框架，包含Struts+Spring+Hibernate三种框架，是目前比较流程的一种Web应用程序开源框架。整个框架结构分为表现层、业务逻辑层、数据持久层以及域模块层四层结构，使得整个框架结构清晰、职责分明且易于开发和维护。SSH集成框架中Struts用于搭建这个应用的基础架构，负责MVC中各个部分的分离。Struts负责业务跳转，Hibernate负责数据持久化部分，而Spring则是负责管理Struts和Hibernate，Struts，Spring，Hibernate之间的交互流程如下图 3所示。在表示层中，使用Jsp页面来与用户进行交互，负责接受用户请求以及接受响应，然后Struts根据配置文件Struts-config.xml选择合适的Action来处理请求。在业务逻辑中，Spring的loC容器负责向Action提供一些业务支持，同时采用一定的技术来提高系统的性能。在持久层中，Hibernate负责对象与数据库之间的映射，完成数据库处理操作。

图 3 SSH集成框架结构

### Struts2框架

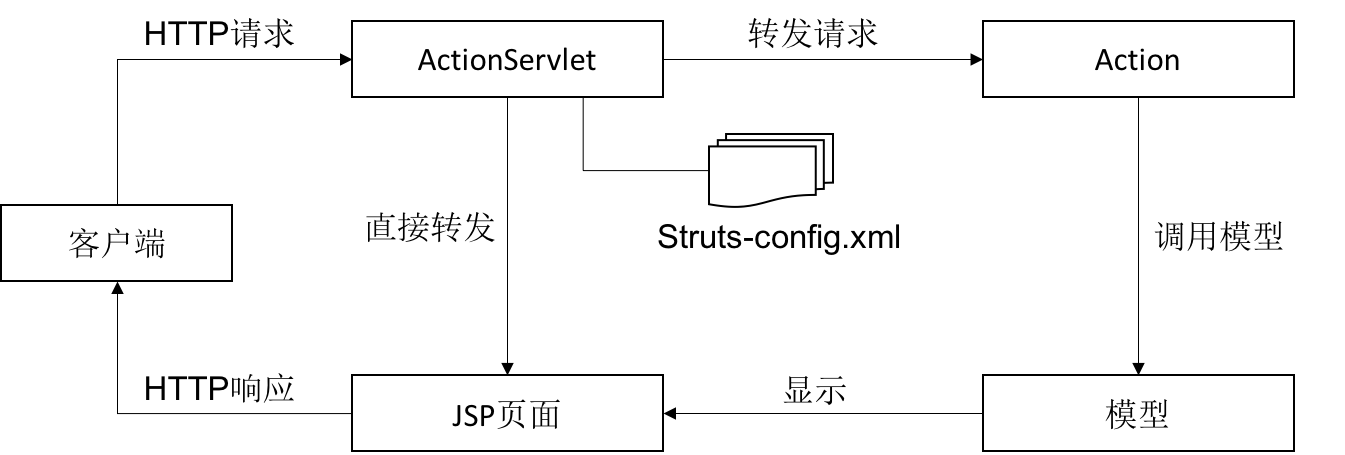
Apache Struts是一个可扩展的免费开源框架，用于搭建企业级的java web应用[6] ，图 4是Struts1的框架结构。Struts1框架是基于MVC的视图层部分的框架，利用Java Servlet 和JSP构建Web应用。Struts结合了JSP、JavaBean等技术，有效地实现了视图与业务逻辑代码的分离，大大降低了系统的复杂度以及不同业务之间的耦合程度，增强了代码的可维护性以及可扩展性，同时提高了开发效率，Struts组成部分以及各部分之间的交互过程。

图 4 Struts1框架结构

Struts2框架[7] 是在Struts1和WebWork框架基础上发展而来的，Struts2继承了WebWork的体系结构和运行流程，同时继承了Struts1的优势，并在此基础上提供了很多新的功能，比如智能默认配置、annonation等，大大减少了Struts1的XML配置。

Strut2框架包含三个部分：核心控制器、业务控制器以及业务逻辑组件，详细结构如图 5所示。

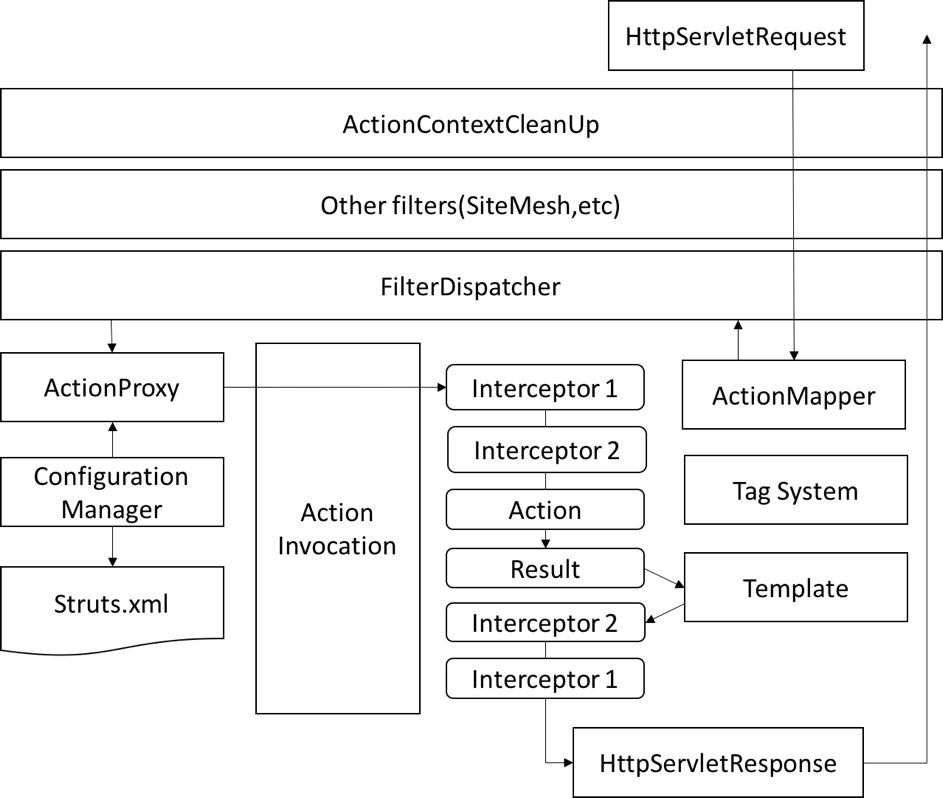


图 5 Strut2 框架结构

首先客户端提交一个HttpServletRequest请求，接下来这个请求要经过一系列的过滤器，顺序依次是ActionContextCleanUp，Other filters，然后交由FilterDispatcher处理，并通过ActionMapper确定是否需要调用某个Action来处理该请求，如果确定能够某个Action能够处理该请求，则FilterDispatcher将该请求交由ActionProxy，ActionProxy通过配置管理器来读取struts.xml配置文件来找到能够处理该请求的Action类。在找到对应的Action类之后，ActionProxy创建一个ActionInvocation对象，并通过该对象使用代理模式来调用对应Action类的实例，同时会根据配置文件为该Action类加载所需的拦截器。最后创建的Action实例会对客户端发出的请求作出相应，ActionInvocation根据struts.xml配置文件返回处理的结果。

### Spring框架

企业应用程序的业务逻辑比较繁杂，开发的复杂度比较高并且后期维护起来也比较困难。Spring是一个开源的框架，主要就是通过分层的概念来降低应用程序的复杂性，对JavaBean的生命周期进行管理[8] 。Spring构建在Spring Core之上，用于创建、配置和管理JavaBean。Spring的核心思想是IOC和AOP，其中IOC为依赖注入，避免了手动显式地创建对象，而是利用java反射机制，在程序运行过程通过Spring的配置文件动态地创建对象以及调用对象；AOP是面向方面编程，是一种编程技术，将影响多个类的行为封装到可重用的模块中。Spring框架包含7个模块，如图 6所示。

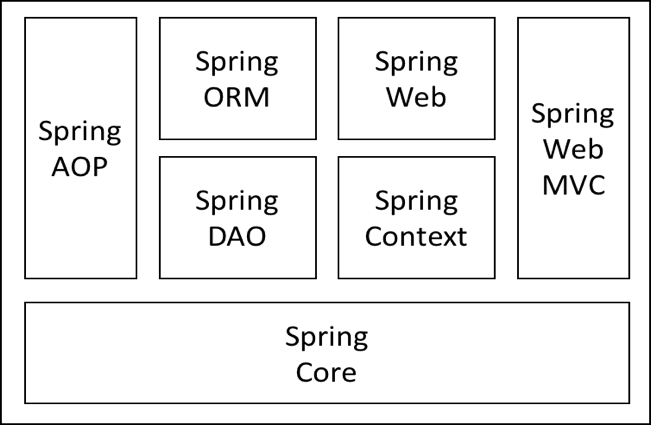


图 6 Spring框架结构

核心模块定义了创建、配置和管理对象的方式，提供了整个框架的基本功能；上下文模块是Spring的配置文件，向Spring提供上下文信息；AOP模块直接将面向方面编程功能集成到Spring框架中，从而使得Spring所管理的任何对象都能够很好的支持AOP，同时为这些对象提供事务管理服务。DAO模块提供了访问数据库的方式，免去了繁琐的数据库访问控制以及异常数据工作。ORM模块为各种数据库对象映射的框架和工具提供了支持，使得Hibernate、iBatis能够集成到Spring框架中。Web MVC模块提供了基于MVC模式的Web应用程序开发的支持。

### Hibernate框架

在面向对象软件开发过程中，关系数据库与对象之间的协同工作是非常耗时的，由于类型不匹配的原因，如何将对象中的数据与关系数据库之间建立映射关系是一个非常困难的问题，而Hibernate则是为解决这个问题而产生的对象关系映射框架。Hibernate框架将Java对象与数据库中的表以及Java数据结构与SQL数据格式进行对应，同时提供了数据库查询以及获取等操作。Hibernate框架的设计目标就是降低开发人员在持久化数据工作中的时间，减少通过SQL和JDBC处理数据的工作。

Hibernate将数据资源映射为一个或多个Java对象，将数据库中的各种操作都封装为Java对象中的函数，大大减少了JDBC以及SQL代码的编写，使得开发人员将更多的精力花费到业务逻辑单元部分[9] 。Hibernate将业务逻辑部分与数据库操作部分的代码分隔开，大大降低了代码之间的耦合程度，并且当数据库发生变化以及数据迁移时，仅需要修改配置文件，提高了应用的可扩展性以及迁移性。

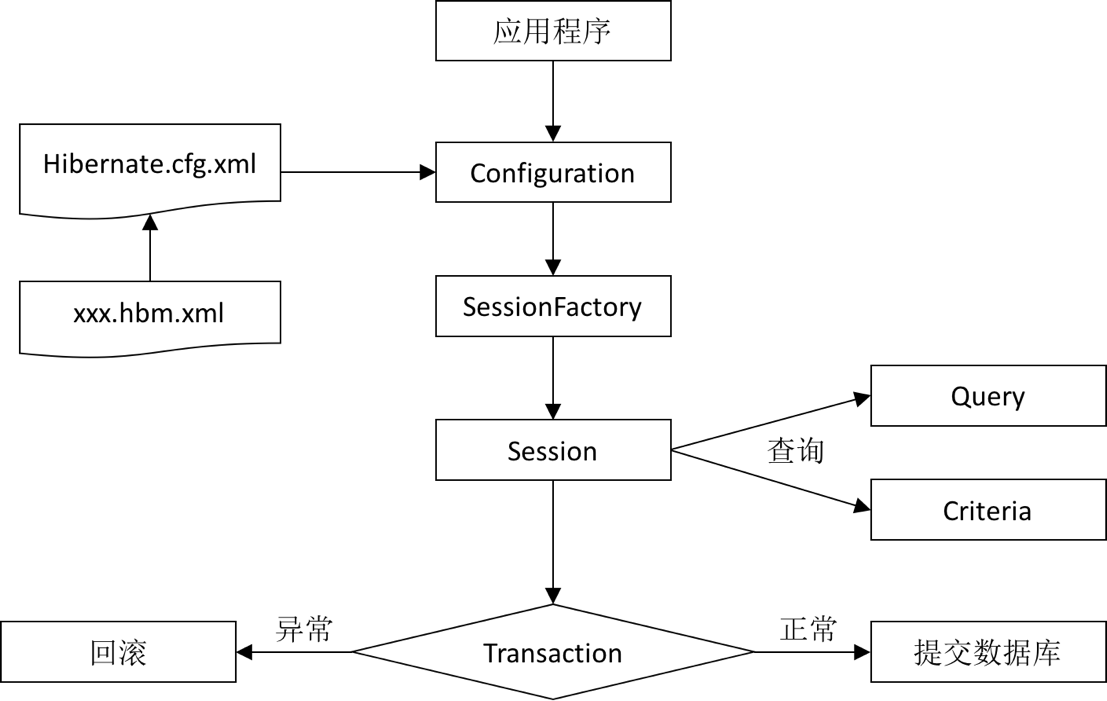


图 7 Hibernate框架结构

图 7显示了Hibernate框架的整体结构以及各个部分之间的流程，包括六大核心结构以及两个主要的配置文件。其中Configuration接口负责读取Hibernate配置文件，并创建一个SessionFactory对象，SessionFactory为Session工厂，负责生产Session对象。Session对象用于完成对象的持久化操作，例如数据库的查询等操作，同时可以产生Transaction对象，该对象负责将操作的结果提交到数据库中。

## MySql数据库

MySQL数据库是一个开源的关系数据管理系统，由瑞典MySQL AB公司开发，目前已经属于Oracle公司[9] 。MySQL借助结构化的查询语言（SQL）来进行管理，是目前比较流行的Web应用的数据库。

MySQL以其高性能、成本低、开源等特点，能够帮助我们有效的进行数据处理，大大提高了工作效率。MySQL数据库的优势可以总结为一下几个方面。

第一、支持多用户、多线程。MySQL的内核采用多线程编程和C语言编程，能够灵活地为用户提供服务，同时避免了占用过多的系统资源，能够充分利用CPU资源。

第二、完全开源。用户完全可以根据自己的需要修改MySQL，并且提供了接口以适应不同语言的连接操作。

第三、用户友好，并且提供了灵活并且安全的权限和口令认证系统。当客户端与数据库建立连接时，两者之间传送的内容以及消息都是经过加密的，并且支持主机认证。同时提供多种客户端以适应不同用户的需要，包括界面、命令行等。

第四、适用于各种操作系统，包括Linux、Unix、Windows以及OS等平台，不仅可以用于服务器还可以用于小型电脑甚至个人电脑。

## 系统开发环境

JDK版本：本系统的框架搭建采用的是JDK1.8，因此开发环境是JDK1.8。

WEB服务器：本系统需要Resin4.0.49以上版本的web服务器，涉及到JVM参数需要在\conf\ resin.properties进行修改，本系统设置的参数如图 8所示：

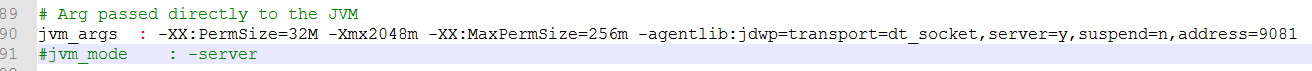
****

图 8 JVM参数

数据库：采用MySQL 5.7版本及以上的数据库，并采用navicat premimum对数据库进行可视化管理。

## 本章小结

本章节介绍了与项目相关的基础知识，包括MVC设计模式以及B/S体系结构。同时介绍了项目开发所要使用的集成框架，以及该框架下Struts2，Spring，Hibernate三个框架的结构以及工作流程，并简要介绍了项目所使用的数据库以及相比其他数据库的优势，也间接说明了选用该数据库的主要原因。从第三章开始将介绍本文的工作内容。

# 需求分析与总体设计

上一章主要介绍了项目中涉及到的基础知识以及理论框架，以及选用的数据库等内容，本章将对项目进行需求分析，包括对计算机语言在线考试系统的可行性的探讨，并通过用例图以及数据流图对系统进行功能性分析。

需求分析是软件开发过程中必不可少的阶段，包含功能性和非功能性需求，直接关系到系统最终的功能以及用户体验[11] 。需求分析就是对用户的需求进行调研，确定系统所要包含的功能以及每个模块之间的交互方式。

系统整体结构。在构建一个系统之前需要对系统的需求进行系统的分析，通过对用户进行调研等方法获取用户需求，确定系统应该包含功能并完善系统功能设计，使得开发人员能够更好地将用户需求转换为系统功能。

## 可行性分析

在开发之前对系统进行可行性分析能够用最小的代价在最短的时间内判断出系统是否能够解决问题。所谓的可行性分析就是通过调查的方法，从经济、技术以及其他方面来分析所要开发的系统是否有必要性和可行性，从而最大限度地避免经济损失，确保系统地成功开发。

从经济方面进行分析，也就是对系统开发的费用进行评估。本系统开发所要求的硬件和软件普通的PC机就可以承担，并且本系统的开发、安装、维护都很方面并且价格低廉，因此本系统的开发在经济方面是可行的。

从技术方面进行分析，也就是对从技术上角度分析系统开发成功的可能性。本系统将使用SSH框架(Struts2+Spring+Hibernate)，该框架大大简化了开发过程，并提高了系统的稳定性和可维护性。系统采用B/S模式，不需要开发额外的客户端软件，降低了开发工作量，并且采用浏览器具有广泛的适用性和可扩展性。数据库部分采用MySQL数据库完成数据的存储以及相关的数据处理操作。MySQL数据库能够存储大量的数据并且数据库内容采用了多线程，具有很好的性能。本系统使用的技术都较为成熟并且能够很好的满足在线考试系统开发的需要，因此本系统的开发在技术方面是可行的。

从需求方面进行分析，目前高校对计算机语言课程的考试相对比较落后，对于理论知识部分通过客观题考察，现有的在线考试系统能够自动进行阅卷，而对于编程类题目则是通过教师人工进行评阅。而本系统能够自动完成理论知识以及编程部分的评阅，大大减轻了教师的工作压力。

从安全方面进行分析，本系统采用MySQL数据库这个较为成熟的数据库，保证了整个系统数据的稳定性和安全性，同时在系统中，对不同的角色进行权限控制，避免了访问越界的发生，同时相比传统的考试方式，系统会对考生的信息进行验证，并采用了自动随机组卷的策略，在一定程度上避免了作弊事件的发生。

## 系统总体需求分析

在线考试系统能够大大方面教师组织考试，同时学生也能够很方便地通过网络参加考试。随着各大高校的扩招以及考试频率的增加，传统的考试方式已经不能满足现代化考试的需要，并且传统的考试方式需要浪费大量的纸张，同时教师需要投入很大的精力选择合适的试题出试卷、打印试卷、安排考试时间以及考场、组织考试，监考以及评语试卷。不仅大大浪费了纸张等资源，并且考试前和考试后都要花费大量的时间进行试卷处理。

随着时间的推移，题库中的题目数量会大大增加，如果通过人工组卷的方式，出题人的工作压力会特别大，而在本系统中，出题人只需要给定每一部分试题的难度以及每一部分试题所占有的比例或者说每一部分的题目的数量，也就是试卷模板，系统会按照出题人的要求自动生成试卷。

考生登录系统后，系统会根据出题人发布的模板自动生成试卷，由于题库中题目中的数量大于试卷中该类型题目的数量，因此相邻考生之间的题目可能并不相同，并且随着题库规模的增大，相邻考试之间考题完全相同的概率会大大降低，有效地避免了相邻考生之间抄袭的现象。

考试开始之后，系统会自动记录考试时间并显示以消耗的时间，考生在考试过程中提前交卷。考试时间结束时，系统会自动保存考生的答题结果。考试结束之后，系统会自动对考生的试卷进行评阅，不需要人工过多的参与，并给出考生的考试成绩，大大减少了学生等待考试结果的时间。相比传统考试方式，在线考试大大提高了考试的公平性，有效避免了人工阅卷产生的分数差异以及不同考场纪律产生的差异，同时在一定程度上遏制了考试作弊的现象。

在软件设计和开发过程中，不仅要考虑系统所要包含的功能还要关注系统的性能。当进行考试时，会有大量的考生同时使用该系统，因此需要考虑系统所能够承受的最大并发数。系统需要简单易用，用户能够很快上手使用而不需要繁琐的培训，另外系统界面友好，用户能够比较快速的找到所需的功能并熟悉系统。同时还要考虑系统的可扩展性和可维护性，能够比较简单的增加或者修改一些模板，并且在系统发生故障时，后续的开发或者维护人员可以简单的对系统进行维护。

## 系统功能需求分析

本系统的致力于为教师和考生提供一个更加方面、高效的考试环境，因此本系统具备传统考试所具有的功能，主要包括：管理考生信息，题库管理、自动组卷、评阅试卷以及考试结果统计等。涉及到的用户有学生、教师、管理员，其中：

管理员部分的功能模块包括：用户管理功能，包括对学生和教师信息的管理；权限管理功能，对教师和学生能够使用的模块进行管理；题库管理功能，对题目进行审核和管理。

教师部分的功能模块包括：登录注册功能；题库管理功能，教师可以对题库中的试题进行增加、修改、查询和删除等操作；试卷模板管理功能，可以对试卷模板进行操作，新建试卷模板，删除已有的试卷模板、修改和查询已有的试卷模板；考试安排管理功能，计划一场考试，并设定考试模板以及考试日期和时间等信息；考卷管理，对考生提交的试卷进行管理，包括查看成绩，统计错题信息和考试成绩等。

学生部分的功能包括：登录注册功能；在线考试，学生身份验证通过之后，系统会自动生成试卷，学生可以开始在线考试；成绩查询，考试结束之后系统会自动评阅试卷，产生成绩之后考生可以查看自己的考试成绩。

## 用例图分析

在软件工程中，对系统进行建模可以帮助我们更好的理解系统，其中用例图[12] 通过描述用户、需求以及功能单元之间的交互，帮助开发者更好的理解系统的需求并转换为系统功能模块。整个系统的用例如图 9所示。

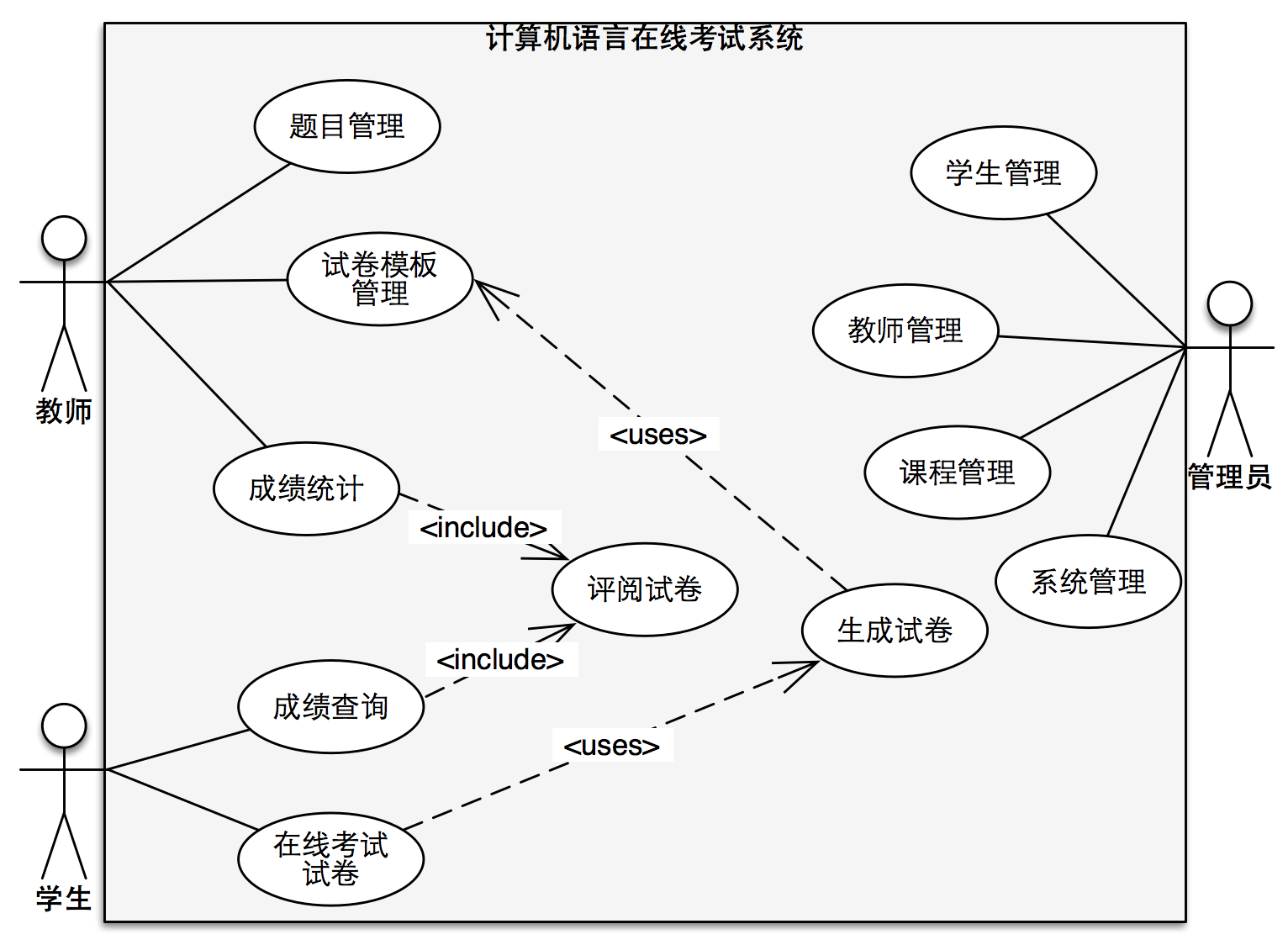


图 9 系统总体用例图

### 用户管理模块用例分析

管理员输入服务器地址后，就可以进行系统的登录操作，输入用户名和密码后点击登录按钮以完成验证，验证过程中系统会确定用户的权限，将有效的功能显示在登录成功后的界面上。管理员对应的功能的用例分析如图 10所示。

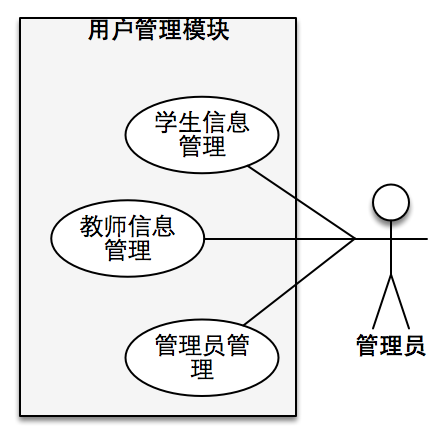


图 10 用户管理模块用例图

管理员登登录系统之后，可以对学生、教师以及管理员的信息进行管理，包括增删改查等操作。表 1是对图 10的具体描述。

表 1 用户管理模块用例分析表

|  |  |
| --- | --- |
| 用例名称 | 用户管理用例 |
| 用例编号 | OE\_001 |
| 行为角色 | 管理员 |
| 前置条件 | 管理员登录系统 |
| 后置条件 | 无 |
| 基本操作流程 | 1. 登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 选择相应的功能 |
| 可选操作流程 | 学生信息管理，教师信息管理，管理员信息管理 |

### 考试管理模块用例分析

在考试之前，教师需要管理要进行考试的考生的信息以及考试安排，即考试的时间、考试时长等信息，并设定试卷模板包括考试题目类型以及每种类型题目的个数，将设定好的试卷模板进行发布。当教师设定的考试时间达到时，学生登录系统并验证身份进行在线考试时，系统会根据教师已发布的试卷模板自动组卷并分配给学生，对应的用例图如图 11所示。

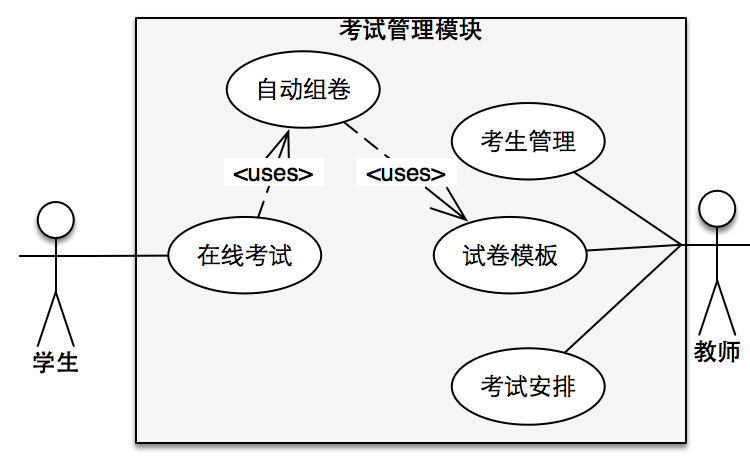


图 11 考试管理模块用例图

考试管理模块由教师进行管理，教师需要输入系统的服务器地址，并输入用户名和密码点击登录按钮提交服务器进行身份验证。当教师身份验证成功之后，系统界面上将会显示考试管理相关的功能。当且仅当教师完成对考试管理之后，学生才能进行在线考试，具体如表 2所示。

表 2 考试管理用例分析表

|  |  |
| --- | --- |
| 用例名称 | 考试管理用例 |
| 用例编号 | OE\_002 |
| 行为角色 | 教师、学生 |
| 前置条件 | 教师登录系统 |
| 后置条件 | 学生在线考试 |
| 基本操作流程 | 1. 登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 生成试卷模板 5. 设置考试时间、考试时长、考试科目、考生名单以及对应的考试模板 6. 学生登录在线考试系统，并进行在线考试 |
| 可选操作流程 | 考生管理、试卷模板、考试安排 |

### 题库管理模块用例分析

题库管理模块就是对考试题目进行日常管理，包括对题目的增删改查等功能，教师对题目进行修改之后需要管理员对题目进行审核，然后确定最终题目状态，对应用例图如图 12所示。

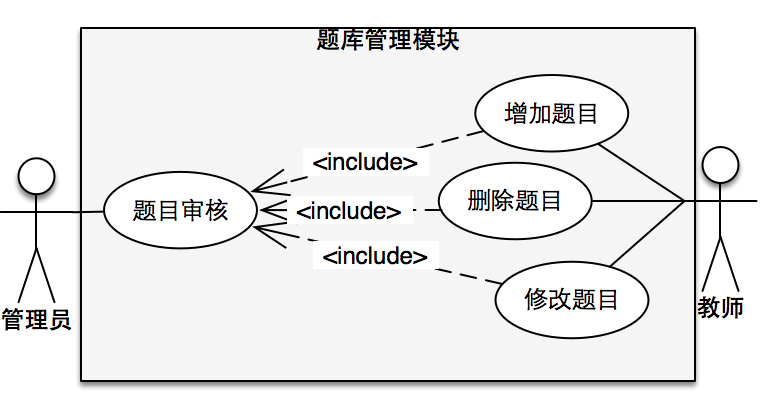


图 12 题库管理模块用例图

教师登录系统后，可以对试卷题目进行增删改查等处理，教师处理之后需要提交并经过管理员进行审核，如果审核通过，则完成对题目的操作，否则就撤销修改，具体如表 3所示。

表 3 题库管理用例分析表

|  |  |
| --- | --- |
| 用例名称 | 题库管理用例 |
| 用例编号 | OE\_003 |
| 行为角色 | 教师、管理员 |
| 前置条件 | 教师登录系统 |
| 后置条件 | 管理员对题目进行审核 |
| 基本操作流程 | 1. 教师登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 对题目进行操作，并提交审核 |
| 可选操作流程 | 增加题目、删除题目、修改题目、提交题目审核 |

### 试卷模板管理模块用例分析

考生登录系统中系统会自动组卷，并且产生的每份试卷的难度以及题目类型、题目数量是完全相同的，因此在考试前需要教师设定考试模板，然后系统根据模板来生成试卷，对应的试卷模板管理模块的用例如图 13所示。

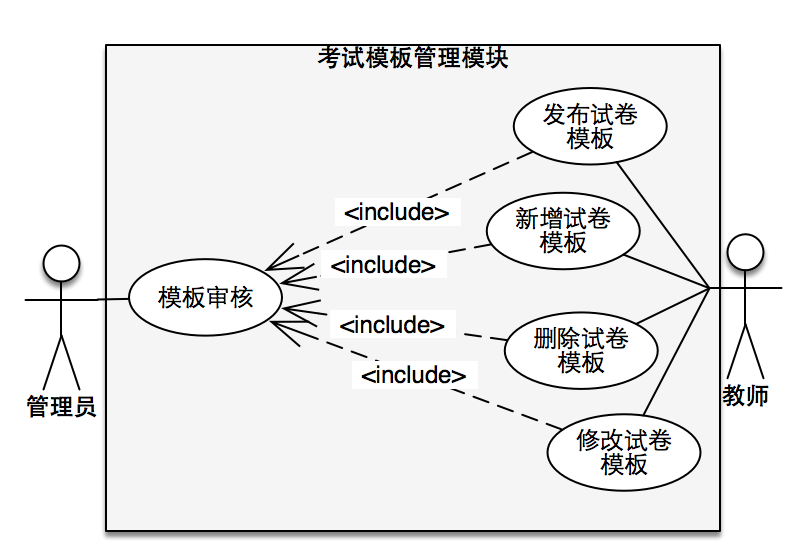


图 13 试卷模板管理模块用例图

教师要组织一场考试需要事先设定试卷的模板，首先教师输入系统服务器网址并验证身份通过之后，就可以对试卷模板进行管理，在试卷模板设定好之后并向管理员提交审核，审核通过之后才可以用于考试，具体如表 4所示。

表 4 试卷模板用例分析表

|  |  |
| --- | --- |
| 用例名称 | 试卷模板管理用例 |
| 用例编号 | OE\_004 |
| 行为角色 | 教师、管理员 |
| 前置条件 | 教师登录系统 |
| 后置条件 | 管理员对提交的试卷模板进行审核 |
| 基本操作流程 | 1. 教师登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 对试卷模板进行操作，并提交审核 |
| 可选操作流程 | 增加模板、删除模板、修改模板、提交模板审核 |

### 成绩管理模块用例分析

考试结束之后，系统会自动评阅试卷，生成考试成绩，教师可以对考试成绩进行统计，也可以对特定考生的成绩进行查询，而学生只能查看自己的成绩，对应的用例图如图 14所示。

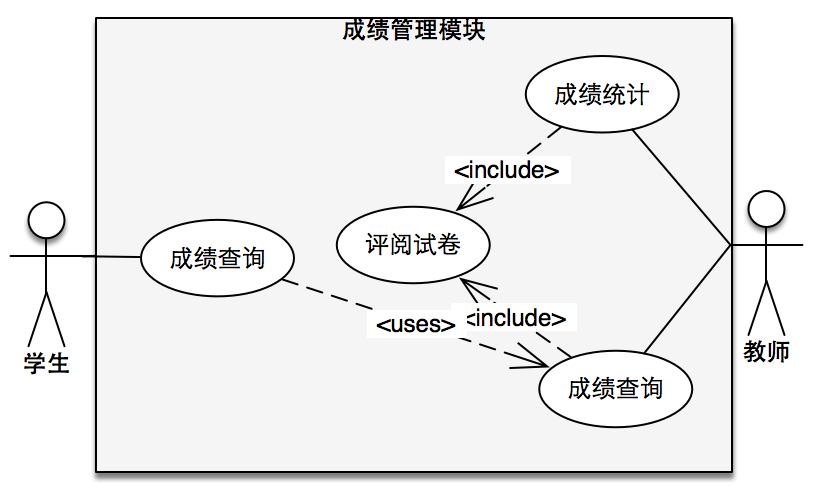


图 14 成绩管理模块用例图

成绩管理模块主要是对考试成绩进行操作的功能模块，教师输入系统地址并通过验证身份之后，可以对考试成绩进行统计操作并指定查询条件来查询符合条件的成绩。另外，考生在考试之后可以查看到自己的考试成绩，但是不可以查看其他人的考试成绩，具体的用例如表 5所示。

表 5 成绩管理用例分析表

|  |  |
| --- | --- |
| 用例名称 | 成绩管理用例 |
| 用例编号 | OE\_005 |
| 行为角色 | 教师、学生 |
| 前置条件 | 学生完成考试、教师登录系统、学生登录系统 |
| 后置条件 | 无 |
| 基本操作流程 | 1. 教师登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 对成绩统计和查询 5. 学生登录系统主页 6. 输入用户名、密码 7. 验证通过 8. 查询自己的考试成绩 |
| 可选操作流程 | 成绩统计（教师）、查询成绩 |

### 系统管理模块用例分析

本系统中对不同角色赋予不同的权限，因此管理员需要对不同角色进行管理，设置不同用户所能操作的模块，对应用例图如图 15所示。

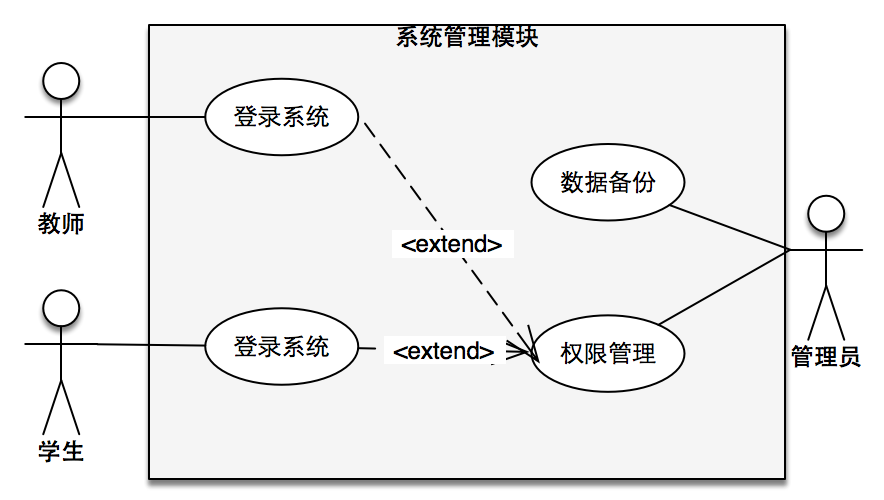


图 15 系统管理模块用例图

管理员负责整个系统的维护与管理，定期对数据库中的数据进行备份同时对系统的不同使用者赋予不同的访问权限，以保证系统的正常运转。管理员输入系统地址后，输入用户名和密码进行身份验证，对教师和学生的权限进行管理，并定期进行数据备份，具体的用例如表 6所示。

表 6 系统管理用例分析表

|  |  |
| --- | --- |
| 用例名称 | 系统管理用例 |
| 用例编号 | OE\_006 |
| 行为角色 | 管理员 |
| 前置条件 | 登录系统并通过身份验证 |
| 后置条件 | 无 |
| 基本操作流程 | 1. 管理员登录管理系统主页 2. 输入用户名、密码 3. 验证通过 4. 对系统用户进行权限管理和数据备份 |
| 可选操作流程 | 权限管理、数据备份 |

## 数据库需求分析

数据库是整个系统的基础，其设计的好坏直接决定了在线考试系统的性能功能，因此在系统开发之前需要对数据库进行调查和分析。数据库需求分析是系统数据库部分开发的基础，其工作质量直接影响到数据库的设计到整个系统开发的成败。合理的数据库设计可以提高数据库的存储效率，保证数据的一致性和完整性，从而是整个系统的性能得到了可靠保证。

通过对前面部分对系统功能的需求分析，可以看出系统应包括一下数据项：教师信息表、学生信息表、管理员信息表、课程信息表、试卷模板表、试卷内容表。学生答案表、答案缓存表、题目表。教师信息表、学生信息表以及管理员信息表是对系统中用户信息的维护，保存用户的账户名密码以及一些身份信息；试卷模板表中存放试卷模板，包括模板编号、出题人信息、试卷组成部分以及难度、考点、是否审核通过等信息；试卷内容表存放的是根据试卷模板生成的试卷，包括对应模板、学生得分等信息；学生答案表包括对应试卷号，题目的编号、答案、得分等信息；考题表包括题目编号、对应模板、正确答案等信息，该表存在的目的是为了提高成绩统计的效率，避免对题目表的查询；题目表则是存在试题的数据表，包括试题编号、试题内容、答案、难度、知识点等信息。

## 系统总体设计

在线考试系统整体分为五大模块，分别为用户管理模块、考试管理模块、题库管理模块、试卷模板管理模块，系统管理模块，结构如图 16所示。

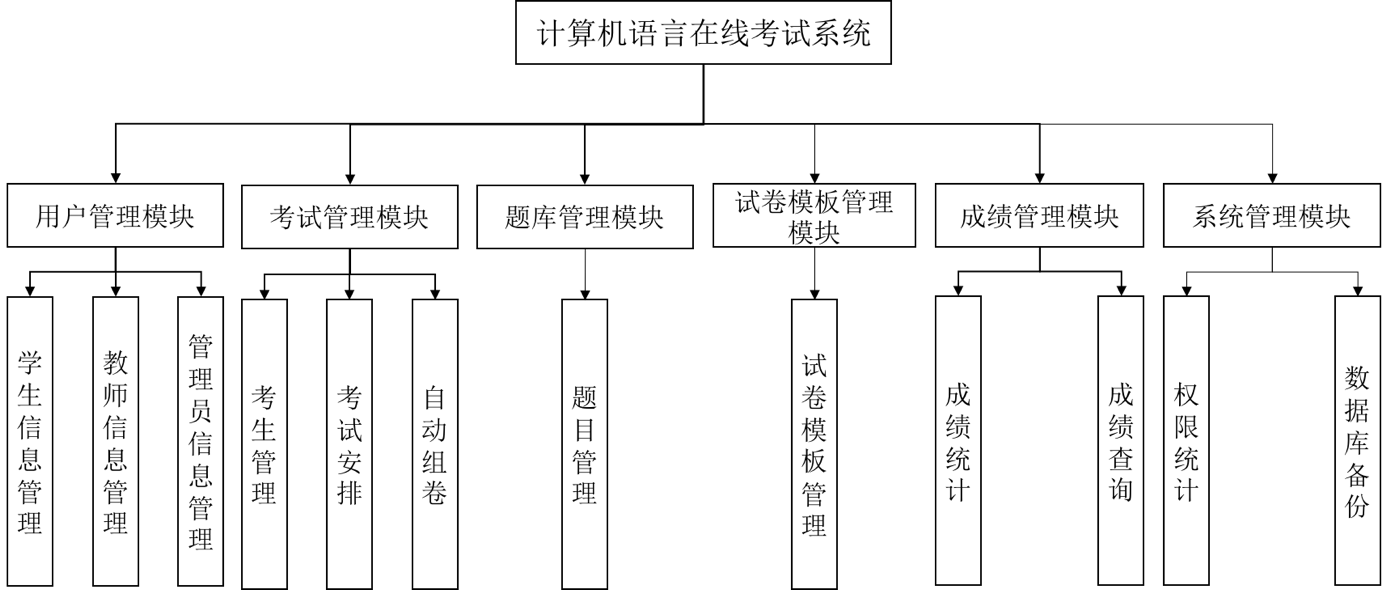


图 16 系统功能模块示意图

1. 用户管理模块

用户管理模块就是对系统的使用者进行管理，系统的使用者为教师、学生以及管理员，管理员可以通过用户管理模块对系统使用者的信息进行增删改查等处理。

1. 考试管理模块

在考试开始之前，需要教师进行一些准备工作以方面学生完成在线考试。教师需要设定考试的时间以及考试时长，通过要考核的内容确定考试所要采用的模板等内容。

1. 题库管理模块

题库是在线考试系统的核心组成部分，关系到试卷的生成和在线考生系统的考核质量。因此需要教师对题库中的题目进行细致的维护，并通过管理员监管来保证题库中数据的质量以及安全性。教师对题库中的题目进行增删改查等操作都要经过管理员的审核后才能正式修改成功。

1. 试卷模板管理模块

试卷模板是本系统的另一个核心组成部分，与题库共同作用生成最终的考试试卷。试卷模板决定了整个考卷所要考核的内容、考察的难度以及每种题型的数量等内容，也就是决定了整个考试能够有效的考核出学生对知识的掌握情况。同时教师可以通过调整试卷模板采用不同的难度来考核学生，更加完善的掌握学生的学习情况和教学效果。在考试之前，教师要先发布一个试卷模板，然后系统根据这个试卷模板进行组卷，在学生参加考试的时候自动生成试卷。教师通过这个模块可以对试卷模板进行增删改查等处理。

1. 成绩管理模块

在线考试系统的最终目标就是考核学生对知识的掌握情况，而考试成绩则是反应这一情况最直接的方式。通过对考试成绩的分析，教师能够从不同角度了解学生对知识的掌握情况，实时调整教学方式和方案，以便学生更好地掌握。教师可以通过该模块，了解出错率最高的题目以及知识点，以及从各个维度对成绩的统计情况。

1. 系统管理模块，对系统权限的控制以及数据库的维护与备份等功能。

权限管理对于任何系统都非常重要，保证了不同角色的访问权限。在本系统中，教师能够完成对考试的一系列操作，而为了考试的公平性，学生在考试之前对一切考试的操作都是未知的，只有在考试正式开始之后，学生才可以看到试卷开始答题。在线考试系统中涉及到多个数据表，用户信息表、题目表等多个数据库表，这些表格共同保证着整个系统的正常运转，因此管理员需要对数据库中的数据进行维护和备份以保证系统的正常工作。

要进行一场在线考试时，首先教师要确定考生的名单，也就是为考生分配进行该场考试的资格。然后教师安排考试时间以及确定考试模板，最后向管理员提交考试模板审核，审核通过之后，发布考试模板，具体流程如图 17所示。



图 17 教师管理流程图

到达考试时间时，考生可以根据自己的用户名和密码登录在线考试系统参加考试，等待系统自动生成试卷之后，考生点击开始考试进行答题，考生可以主动点击提交试卷按钮提交试卷或者考试时间结束时系统会自动提交试卷。试卷提交之后，系统自动对答卷进行评阅生成考试成绩，对应流程图如图 18所示。



图 18 学生在线考试流程

## 本章小结

本章首先对整个系统的可行性进行分析，从各个角度确定系统可行之后，接下来从功能性和非功能性两个方面进行需求分析，确定本系统的业务需求以及性能上的需求。然后通过用例图的方式对系统的各个功能模块进行用例分析，详细描述系统角色与功能模块之间的交互过程以更好完成对系统的建模工作。最后对系统的整体架构以及功能进行设计，并采用流程图的方式说明了不同业务的工作流程。这些工作为下一章的系统的详细设计以实现奠定了良好的基础。

# 系统设计与实现

## 数据库设计

通过第三章对系统的整体进行设计和数据库的需求分析，我们确定采用MySQL数据库存储本系统涉及到的表格。管理员信息表 ADMIN\_TABLE，教师信息表 TEACHER\_TABLE，学生信息表 STU\_TABLE，知识点表KPOINT\_TABLE，题目表 QUESTION\_TABLE，试卷表TESTPAPER\_TABLE，试卷模板表 PAPERMODULE\_TABLE，课程表SUBJECT\_TABLE，考题缓存表QUESTION\_CACHE\_TABLE。由于篇幅的限制，下面只给出一些关键表的详细结构。

题目表QUESTION\_TABLE、知识点KPOINT\_TABLE，课程表SUBJECT\_TABLE这三个表格都是由教师进行维护，每个题目都有所属课程的ID以及所要考察知识点的ID。表 7是题目表的详细字段，每个题目都有唯一的标识ID，同时记录了录入该题目的创建者ID和录入时间，以及用于描述题目的一些字段，例如难度、内容、类型等。创建者在录入一个题目的同时，需要指定该题目的正确答案。对于客观题，题目答案中只需要直接给出答案内容，而对于客观题，需要给出测试数据以及标准代码。

表 7 题目表QUESTION\_TABLE结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否为空 | 描述 |
| ID | BIGINT | 50 | 否 | 题目ID号 |
| CREATORID | BIGINT | 50 | 否 | 创建者ID |
| CREATEDATE | DATE |  | 否 | 创建时间 |
| DIFFICULITY | VARCHAR | 10 | 否 | 题目难度 |
| CONTENT | VARCHAR | 5000 | 否 | 题目内容 |
| TYPE | VARCHAR | 5 | 否 | 题目类型 |
| KPOINTID | BIGINT | 50 | 否 | 知识点ID |
| SUBJECTID | BIGINT | 20 | 否 | 课程ID |
| ANSWER | VARCHAR | 5000 | 否 | 题目答案 |
| SCORE | VARCHAR | 50 | 否 | 参考分值 |

考题缓存表QUESTION\_CACHE\_TABLE用于存储满足特定考试模板的试题，该表的设计与QUESTION\_TABLE相同，主要的用途是预先将满足考试模板的题目抽取出来存储在该表中，在考试开始之后，系统只需要对考题缓存表进行操作。随着时间的增长，题目表中的数据会逐渐增加，如果直接对题目表进行检索，速度会大大减低。而实现将符合条件的题目加入到题目缓存表中，可以大大加快题目的检索速度，进而提高系统的组卷速度。

试卷表 TESTPAPER\_TABLE用于存放考卷信息，结构如表 8所示，是根据预先设定的考试模板生成的最终的试卷内容，一个试卷仅有一个学生使用，其中MODULEID为对应试卷模板ID号，ISCHECK表征试卷是否已经打分，TITLE为试卷名称，STUDENTID为考试学生的学号，SCORE为学生在此考试中最终得分。TEST\_RESULT字段采用XML的格式存储考卷内容以及考生答题后的试卷，格式如下图 19所示。<test\_paper></test\_paper>为根元素，表征一个完整的试卷，<item></item>为子元素，用于标识该试卷中的一道题目，其中id对应的题目表中的ID号，type为题目的类型，content为题干，answer为该题目的标准答案而user\_answer为学生提交的答案，score为该题目的分值而user\_score是学生在该题目上的得分。与<item>同一等级的标签还有<total\_score>和<total\_user\_score>，前者为整张试卷的总分而后者为学生使用该试卷进行答题的总得分。

表 8 试卷表TESTPAPER\_TABLE结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为空 | 描述 |
| ID | VARCHAR | 50 | 否 | 试卷ID号 |
| TITLE | VARCHAR | 50 | 否 | 试卷名称 |
| STATE | VARCHAR | 10 | 否 | 是否有效 |
| SUBJECTID | VARCHAR | 20 | 否 | 课程ID |
| MODULEID | VARCHAR | 20 | 否 | 试卷模板ID |
| STUDENTID | VARCHAR | 20 | 否 | 考生ID |
| TEST\_RESULT | VARCHAR | 100000 | 否 | 提交试卷内容 |
| ISCHECK | VARCHAR | 10 | 是 | 是否已经评阅 |
| SOCRE | VARCHAR | 20 | 否 | 考试成绩 |

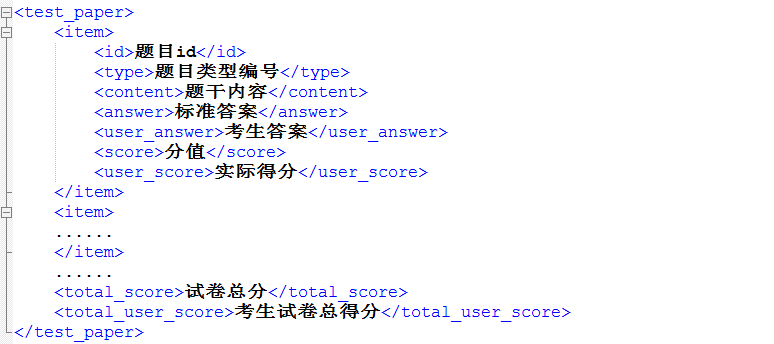


图 19 TEST\_RESULT字段格式

试卷模板表PAPERMODULE\_TABLE，教师制定的生成试卷的策略，POLICYCONTENT为组卷策略描述，仍采用XML格式进行描述。

图 20 试卷模板表PAPERMODULE\_TABLE结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为空 | 描述 |
| ID | VARCHAR | 50 | 否 | 模板ID号 |
| CREATORID | VARCHAR | 100 | 否 | 创建者ID |
| CREATEDATE | DATE | 7 | 否 | 创建时间 |
| TITLE | VARCHAR | 50 | 否 | 试卷名称 |
| STATE | VARCHAR | 10 | 否 | 是否审核通过 |
| SUBJECTID | VARCHAR | 20 | 否 | 课程ID |
| POLICYCONTENT | VARCHAR | 500 | 是 | 组卷策略 |

学生信息表STU\_TABLE，由学生自己通过学号注册或者管理员代为注册，并设定在线考试系统的登录密码。其中EXAM\_MODULE默认为null，只有当教师或者管理员将对应考试模板ID号添加到该字段，考生才能够参加对应的考试。

图 21 学生信息表STU\_TABLE结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段长度 | 是否为空 | 描述 |
| ID | VARCHAR | 50 | 否 | 考生ID号 |
| PWD | VARCHAR | 50 | 是 | 密码 |
| NAME | VARCHAR | 50 | 否 | 考生姓名 |
| MAJOR | VARCHAR | 50 | 否 | 专业 |
| GRADE | VARCHAR | 50 | 否 | 年级 |
| GENDER | VARCHAR | 10 | 否 | 性别 |
| BIRTHDAY | DATE | 7 | 是 | 出生日期 |
| ID\_NUMBER | VARCHAR | 20 | 是 | 身份证号 |
| EMAIL | VARCHAR | 50 | 是 | 电子邮箱 |
| PHONE\_NO | VARCHAR | 50 | 是 | 联系电话 |
| PHOTO | VARCHAR | 200 | 否 | 电子照片地址 |
| EXAM\_MODULE | VARCHAR | 50 | 是 | 考试模板ID |

ER图(Entity-Relationsip Diagram)是由Peter Chen发明，是概念数据模型的高层描述所使用的数据模型图，为表述试题联系模型图形式的数据模型提供了图形符号[13] 。上面介绍了主要的数据表，为了更加全面的了解整体的数据结构，我们采用实体-关系图(Entity-Relationship Diagram)的方式进行描述，如图 22所示。由于部分实体的属性过多，图 22中的实体只展示了部分属性。从图中可以看出，管理员负责管理学生、教师信息和课程信息，以及对试卷模板的审核；教师负责知识点和题目的录入，同时指定试卷模板；在考试开始之后，系统会自动根据考生模板生成试卷。

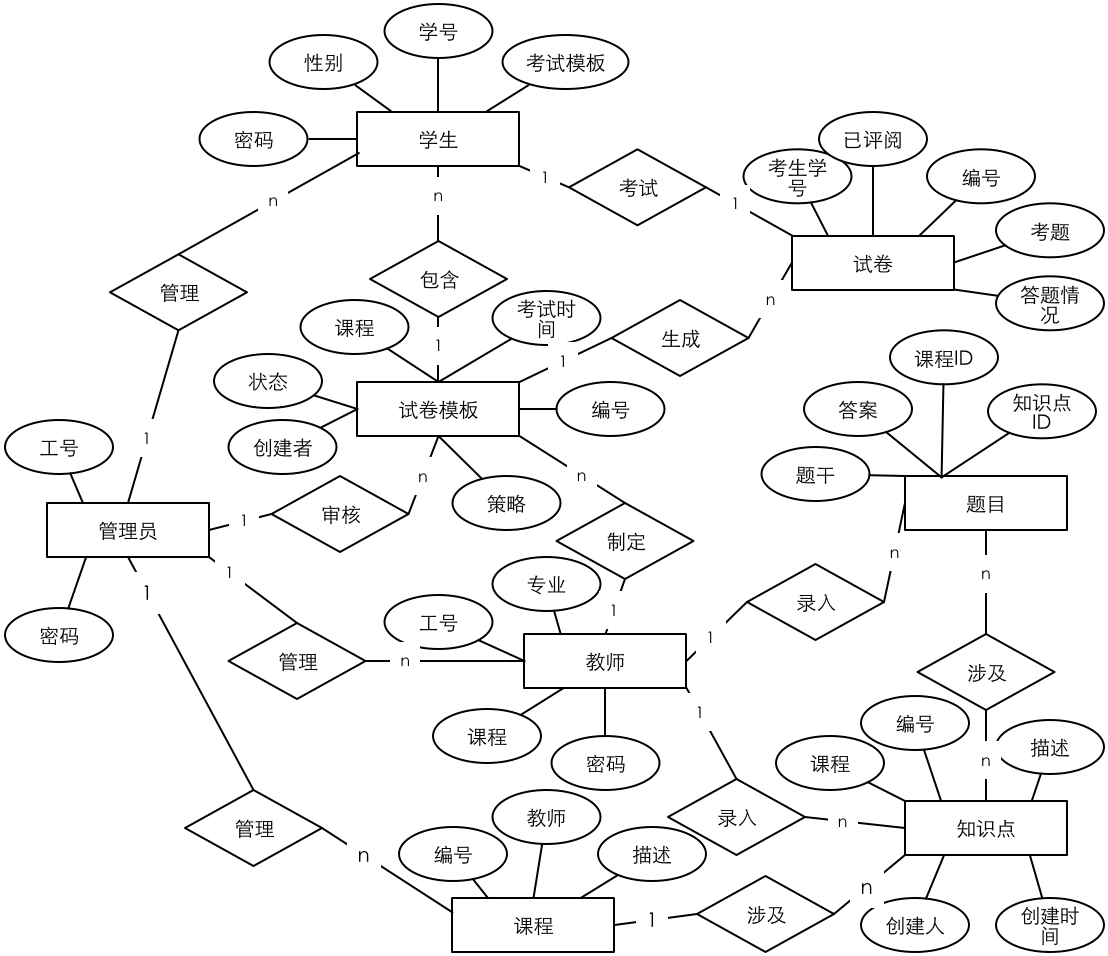


图 22 在线考试系统实体-关系图

本文采用内Hibernate实现数据层的持久化，建立实体类与数据表之间的映射关系，从而避免开发人员采用JDBC的方式直接操作数据库，从而保证整个系统遵从面向对象的方式进行开发。Hibernate通过hbm.xml格式的文件将POLO类映射到数据库的表格中，不需要编写任何额外的代码。IntelliJ IDEA支持将数据库中的表格直接转化为POLO类，大大减少了开发的工作量。POLO类对应数据库表格，每个POLO类的实例对应数据库中的一行记录，POLO类的属性对应数据库表的字段。图 23就是QuestionEntity与QUESTION\_TABLE的映射文件question.hbm.xml。这个文件通过class name与table建立联系，并将POJO类的属性一一与table的字段对应，包括名称以及数据类型。之后对数据库中数据的操作就可以简单通过操作POJO类来实现。

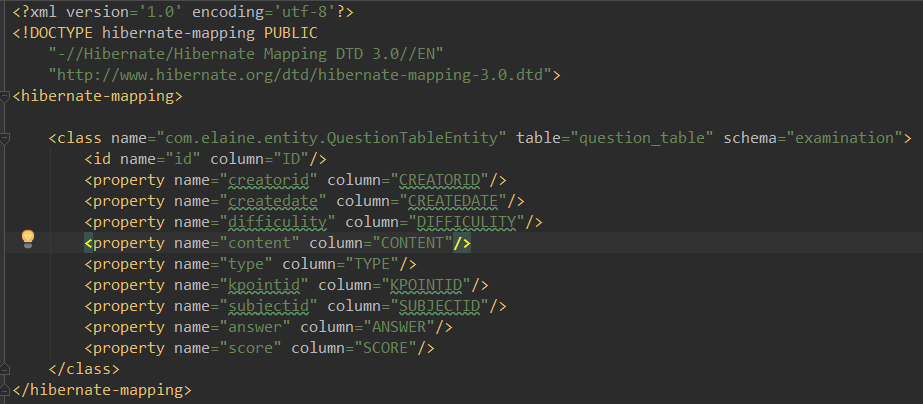


图 23 实体关系映射XML文件

单纯使用Hibernate进行事务处理，每次对数据的操作都需要经历8个步骤。首先需要获取Configuration对象，接下来通过Configuration对象创建sessionFactory对象，并通过sessionFactory对象创建session，然后通过session开启事务，进行一些数据持久化操作，处理结束后需要提交事务，如果发生异常需要进行事务回滚，最终关闭session，对应代码如下：

**public void** getPersistence(){  
 Configuration configuration=**new** Configuration().configure();  
 SessionFactory sessionFactory=configuration.buildSessionFactory();  
 Session session=**null**;  
 **try**{  
 session=sessionFactory.openSession();  
 session.beginTransaction();  
 *//数据持久化操作* ......  
 session.getTransaction().commit();  
 }**catch**(Exception e){  
 e.printStackTrace();  
 *//进行事务回滚* session.getTransaction().rollback();  
 }**finally**{  
 **if**(session!=**null**){  
 **if**(session.isOpen()){  
 session.close()  
 }  
 }  
 }  
}

而用Spring对Hibernate进行管理时会大大简化这个步骤，Spring提供了HibernateTemplate类，这个类对Session等进行了封装，并且会自动提交事务，不需要开发者手动开启事务和提交事务，只需要将对象进行save操作即可。图 24是Spring对Hibernate的配置。

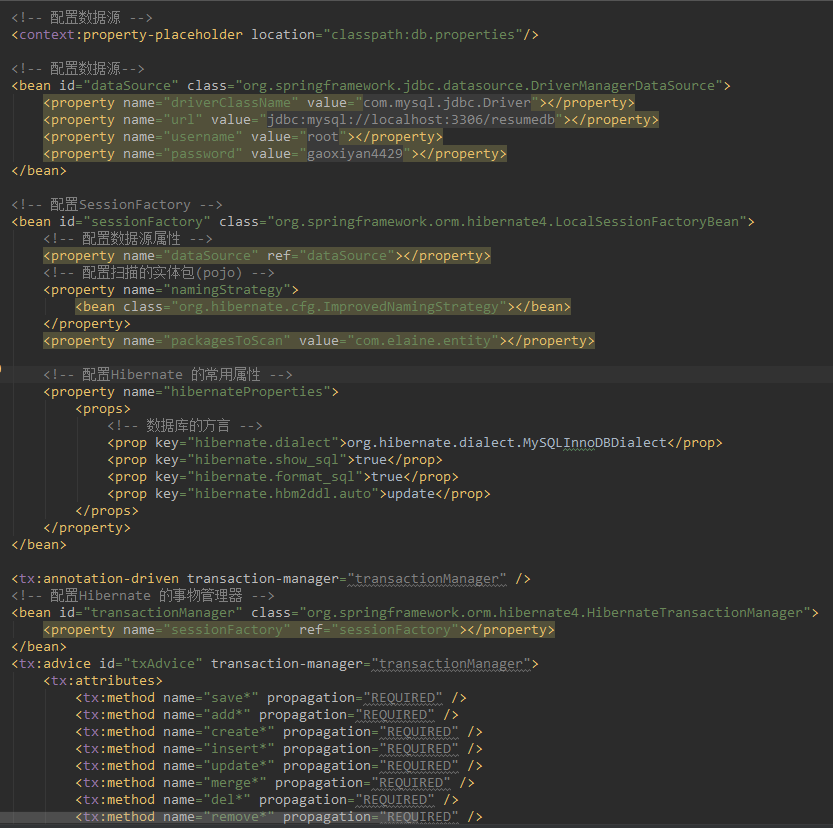


图 24 Spring配置Hibernate的XML文件

在Spring配置文件中加入Hibernate相关的配置之后，对数据持久化的操作如下图 25所示。只需要在使用sessionFactory之前对其进行类型声明，后续可以通过getCurrentSession得到session对象，然后进行数据库的相关操作，Spring会进行后续的处理。

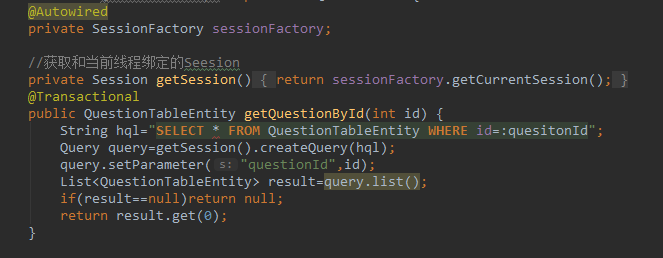


图 25 持久化操作示意代码

## 功能模块详细设计与实现

### 系统登录模块详细设计与实现

系统登录是整个系统的入口，用户需要输入用户名和密码，点击登录按钮提交。如果提交的信息没有验证通过，则会提示用户名或者密码错误，用户需要重新输入用户名和密码；否则系统会根据用户的账号判定用户的身份，并跳转到对应的系统页面，详细流程如图 26所示。



图 26 系统登录模块流程图

用户角色管理将会严格按照需求分析中的用例所展示内容进行设计区分，管理员创建用户信息时，会根据用户的角色放入到不同的用户信息表中，并附带不同的操作权限对应的操作界面如下图 27所示。



图 27 登录界面示意图

### 学生在线考试模块详细设计与实现

学生在线考试是整个系统的核心模块，考生在登录考试系统后，需要点击在线考试选项，系统会判断该学生是否具备考试资格。如果学生不具备考试资格，则直接退出；如果具备考试资格，则系统会根据试卷模板生成试卷并开始计时。如果在考试过程中，考生主动点击提交按钮，考试开始之后半个小时才能够提交试卷，否则继续答题。另外，到达考试时限后，系统会自动提交试卷结束考试，流程如图 28所示。



图 28 学生在线考试模块流程图

学生登录系统之后，可以点击在线考试选项，系统会自动显示该学生能够参加的考试，学生选择对应的考试之后，则进入核对个人信息页面如图 29所示。



图 29 学生信息核对界面

如果个人信息无误，学生点击开始考试按钮，进入正式考试界面如，系统会自动显示考试剩余时间，学生答完题并检查无误之后可以点击提交按钮进行交卷，考试过程中系统会每个十分钟自动保存考生的答题结果。考试时间结束时，系统会自动提交考生的答题结果，具体的答题界面如图 30所示。



图 30 考试界面

### 教师相关功能模块详细设计与实现

教师登录系统之后就可以维护所负责的课程的题库，没门课程的题库的初始状态都是一个以课程名称命名的目录，教师可以在该目录下添加多个子目录，每个子目录以知识点命名，包含了对应知识点下的题目。教师在添加题目时，需要指定题目类型以及难度、分值等信息。教师可以在对应目录下一一添加题目也可以下载指定excel模板，并按照模板的格式离线添加题目并批量导入到题库中，对应的界面如图 31所示。



图 31 题目管理界面

教师在组织一场考试之前，需要设定试卷模板，也就是指定组卷策略：题目类型对应的知识点、难度等信息。在学生点击开始考试时，系统会根据教师事先指定的策略自动从题库中选取满足条件的题目进行组卷，保证每份试卷的难度相同，同时保证每份试卷的题目顺序或者内容不相同，试卷模板管理页面如图 32所示。



图 32 试卷模板管理界面

## 系统关键模块设计与实现

### 自动组卷模块设计与实现

在线考试中的自动组卷是非常重要的模块，如何从题库中选取满足要求的试题并最终组成满足限制条件并合理的试卷是自动组卷需要解决的问题。想到得到高质量并且结构合理的试卷依赖于正确的组卷策略，同时在一定程度上能够比较真实地考察出学生对知识的掌握水平并避免相邻考生撞题等状况的发生。

目前最常用的是基于随机抽题法的自动组卷[14] ，通过随机函数不断从题库中抽取试题直到组卷完成或者无法在试题库中找到满足条件的试题。这种算法时间复杂度较高，并且在组卷约束条件较多的情况下，经常会出现组卷失败。因此该种算法只适用于题库比较少并且组卷相对简单的场景。回溯试探算法[15] 在随机抽题算法的基础上的改进版，它随机从题库中抽取题目，如果该题目满足约束条件则将题目放入到预先建立的栈中，如果抽取的试题不满足约束条件，则返回上一个成功状态，继续随机搜索。回溯试探算法简单，有效避免了随机法中试题重复抽取的问题，但是由于需要栈等额外存储，占用内存并且组卷时间长、效率低。基于遗传算法[16] 的组卷算法是近几年刚发展起来的，该种方法的组卷效果很好，但是算法复杂度较高，运行速度慢并且容易早熟限于局部最优解，因此本系统选择在传统的随机抽题法的基础上对组卷的方式进行限制。教师需要在设定考试模板的时候指定试卷中每道题考察的知识点和对应的难度，然后系统会将这些限制条件进行打乱，并依次从题库中抽取满足条件的试题，然后从这些试题中随机选择一个题目，并将该题目作为试卷中的一个题目，然后从限制条件中取出下一个，重复进行上述过程，直到所有的限制条件均被实现，具体流程如图 33所示。



图 33 自动组卷算法流程图

为了满足考试的实时性要求，我们选用线程池的方式来进行在线组卷，每个线程负责一份试卷的生成。我们采用了Spring容易来管理业务bean，并且Spring封装了多线程的实现，而要使用线程池只需要在ApplicationContext.xml中添加如图 34所示的配置内容，线程池中的每个线程中执行算法1所示的代码。

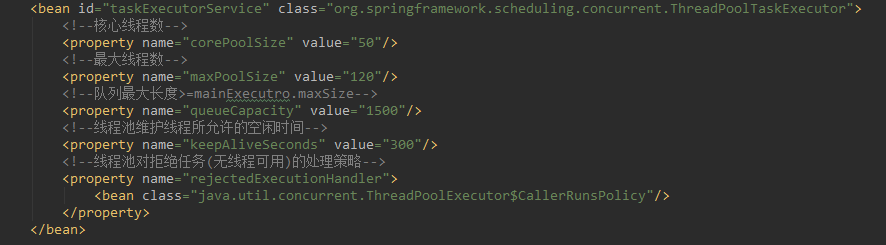


图 34 Spring中配置线程池

算法1是组卷算法的伪代码描述，首先需要根据题目的数量产生0~size-1的一个随机排列，用于打乱题目要求，从而按照不同的顺序产生试卷，在一定程度上降低了完全相同试卷的产生。然后从题目要求中取出第个要求，并随机取出一个满足该要求的题目，将该题目加入到试卷中。依次重复上述过程，知道所有的要求都被实现，那么试卷生成完毕。

|  |
| --- |
| **算法1 GeneratePaper** |
|  |
| 1. 从题目要求的中取出第个要求 2. 从数据库中随机得出满足条件的    7. 将加入到试卷 |

### 自动评阅试卷设计与实现

计算机语言类课程的考试题目可以分为两种类型客观题和程序设计题，其中客观题的批改比较简单，只需要将考生提交的答案与标准答案进行比对，如果相同则改题满分，否则为0分。而程序设计类题目的批改过程比较复杂。

自从上世纪60年代以来，许多学者致力于研究基于文本答案的主观题自动批改技术，并设计开发出了多种自动阅卷系统：Beierle等人[17] 提出了一种针对Prolog语言的课程作业的评阅方式AT(x), 通过分析学生提交的程序然后将评阅结果通过Web的方式反馈给学生。Singh[18] 等人开发的系统通过一个现有的常见错误表对学生提交程序中的简单小错误进行简单的修正。Jinrong[19] 等人则提出了一种针对语法和语义匹配的方式，通过将学生提交的程序与标准程序进行语义匹配来计算得分。Shamsi和Elnagar等人[20] 提出了一种针对java语言的评阅系统，该系统结合了Junit 框架和程序图表示两种方式，综合计算学生得分。以上这些方法适用于人数较少的场景并且只针对一种特定的计算机语言，同时采用的算法相对比较复杂，因此本文将根据计算机语言的特征采用一种相对比较简答的评阅方法。相比于自然语言，计算机语言具有以下特征[21] ：

1. 程序的基本格式固定，语法和词法的约束都比较严格，格式相对比较统一。
2. 计算机语言考试的对应程序代码一般都比较短小，一般不超过40行。
3. 对于特定的题目，程序的整体框架大体相同。

目前针对多种语言的程序设计自动阅卷方法都是通过运行学生提交的程序，然后将程序的输出与标准答案进行比对，根据比较的结果得出相应的分数。这种简单通过比对输出结果来评判分数的方式显然是充分的，因此本文提出了一种综合程序输出和程序理解的方法如图 35所示，在某些情况下通过程序理解的方式来纠正通过程序输出计算得分的方式，使得最终的结果更加的公平、合理。该模块首先对读取数据库中未评阅试卷中的程序设计题，并对学生提交的程序进行编译，如果编译通过则运行程序，将测试用例文件作为程序的输入。如果程序运行超时或者出现异常，则将该程序与标准程序进行匹配计算得分。如果程序正常运行并且程序输出与标准输出的匹配率大于0，则按照测试用例通过的比例计算得分。程序未通过编译或者运行超时、运行出错等，都将进入程序匹配阶段。



图 35 自动评阅试卷算法流程

在本系统中，学生提交试卷后，系统会直接读取学生提交的试卷答题信息，编译运行检查结果并将结果写入到数据库中。在编程部分的试题进行评阅时，需要调用服务器上安装的编译器来编译考生提交的源代码，并保证在编译提交代码发生问题后，系统不会出现致命错误。为了达到该目的，需要引入一个守护线程，负责启动评阅试卷的线程，每个线程负责启动对应语言的编译器对程序进行编译和运行。守护线程负责每个一段时间从数据库中检查是否有未评阅的试卷，如果有则启动线程对提交的结果进行评阅。守护线程的关键代码如下图 36所示：

public void initProcess(){

while(SubmitedPapaer.exists()){ //存在为评阅的试卷

List<Integer>paperIDList=submitPaper.getUnmarkedPaperID();

for(int paperID:paperIDList){

StringBuilder resultStr=new StringBuilder();

List<Integer> programIDList=submitPaper.getProgramIDList(paperID);

for(int programID:programIDList){

String program=submitPaper.getProgramByID(programID,resultStr);

int lagType=Commons.getLag(program);

if(lagType==2){//java类型

JProcessjp=new JProcess(programID,resultStr); //处理程序源代码，并将处理的结果写入到数据库中

jp.start();

}else(lagType==0){//C/C++类型

CProcesscp=new CProcess(programID,resultStr);

cp.start();

}

}

}

}

}

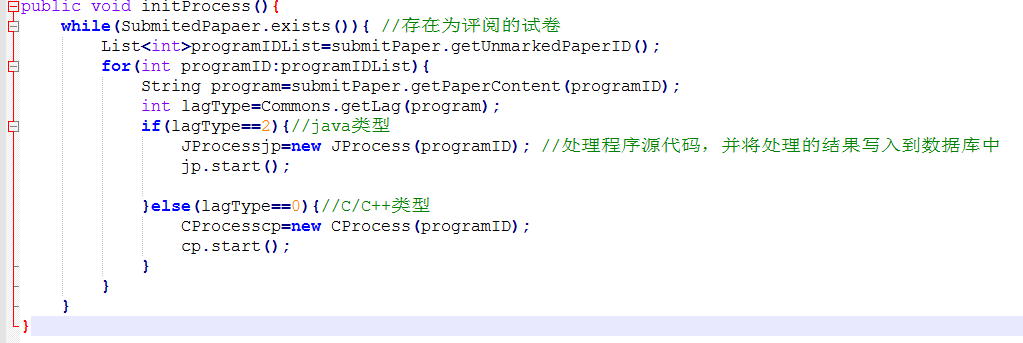


图 36 守护线程关键代码

如果源程序编译之后能够运行并产生结果，则将产生的结果与标准答案进行对比。若与标准答案完全一致，则满分；若部分正确，则按照正确的比例给分；若得分为0，则将学生提交的代码与标准代码进行比对评分；如果程序未通过编译，无法产生运行结果，则转入程序修复。

1. **C/C++评判线程**

评判线程首先启动GCC编译器对源代码进行编译，然后指定测试数据文件作为编译后.exe文件的输入，并得到运行结果。线程启动之后，首先调用compile函数对源代码进行编译，如果编译通过，则调用编译后的.exe文件并指定测试文件作为输入，运行结束之后将输出结果与标准结果进行比较，并按照测试数据通过的比例计算得分。如果分数为0或者程序编译未通过，则通过比较源代码与标准代码的相似度，根据相似度计算最终得分并在标准分数的基础上减去部分分数。线程对应的run()函数流程图如图 37所示。



图 37 C/C++线程试卷评阅算法流程

run()函数的关键代码如图 38所示：

public void run(){

Program currentProgram=submitPaper(this.programID);

String stdProgram=currentProgram.getStdProg();

String stdOut=currentProgram.getStdOut();

int stdScore=currentProgram.getScore();

boolean compiledResult=compile(currentProgram);

float score=0.0;

if(compiledResult){//编译通过

long before=Calendar.getInstance().getTimeInMills();//获取开始运行的时间

String out=execute(currentProgram);

long after=Calendar.getInstance().getTimeInMills();//运行结束时间

long realTime=after-before;

if(realTime>=this.timeThreshold){

float rightRatio=calSimilarity(currentProgram); //计算提交的源代码与标准代码的相似度

score=rightRatio\*stdScore;

}else{

score=equalRatio(out,stdOut)\*stdScore;//将输出与标准答案进行比对

if(out==null || score==0.0){//程序运行出错

float rightRatio=calSimilarity(currentProgram); //计算提交的源代码与标准代码的相似度

score=rightRatio\*stdScore;

}

}

}else{

float rightRatio=calSimilarity(currentProgram);

score=rightRatio\*stdScore\*0.8; //测试数据完全没通过，那么会减去20%的分数，然后在此基础上通过相似度比例计算最终得分。

}

this.resultStr.append('<item><id>'+this.programID+'</id><type>1</type><score>'+score+'</score><item>');//按照指定的结果记录结果

}



图 38 C/C++语言评阅线程关键代码

1. **JAVA评判线程**

JDK是java语言的软件开发工具包，包括了java语言运行的环境配置以及常用的API。JDK支持各种操作系统，安装之后配置环境变量即可使用。Java评判线程会启动Javac编译器对源代码进行编译生成Java字节码，然后由java虚拟机执行Java字节码，并将指定的测试文件作为输入，并得到程序运行结果。与C/C++进行程序编译的过程类似，JProcess启动之后，首先调用compile()函数对源代码进行编译，如果编译通过，调用execute()函数执行程序，并将测试文件作为输入，并将程序的输出结果与标准结果进行比对，按照正确的比例给分。如果测试结果正确率为0或者程序未通过编译，则通过计算提交程序与标准程序的相似度来给分，具体流程图如图 37所示。

1. **程序相似度匹配**

假定学生提交的程序代码为S，标准程序代码为P。

1. 采用词法分析进行程序预处理

程序展示方式是多种多样的，变量名称的不同或者变量定义的顺序不同都将导致程序的表示方式不同，因此在进行程序匹配之前，需要对其进行归一化处理。本文采用词法分析的方式过滤掉注释等无关信息，并将程序映射为一个token串。其中不同的关键字对应不同的token，变量名等无关信息映射为相同的token，最终将程序转为token串。

1. token串分割

将token串分割成一个个token子串，每个token子串包含m个token，相邻两个token子串之间有m-1个token是重复的。

其中len(S)为token串S中包含token的个数，为第个包含个token的子串。

1. 计算token子串的哈希值

对token子串计算哈希值的目的就是将相似的子串尽可能映射到比较相近的值，而相差较远的token子串的哈希值也应该相差较大。另外一方面，将子串映射为一个值，大大降低了空间负责度。对应的hash函数定义如下：

其中 ，为一个token，为的分值。不同关键字对应不同的分值，变量类token的分值为1，关键字的分值为5，q=5.

1. 相似度计算

S对应的哈希值串为，P对应的哈希值串为，相似度计算公式如下：

其中为S中token子串的个数，为哈希值相同的token的个数。

对应关键代码如下所示：

public float calSimilarity(){

List<String>proTokens=lexer(this.program);

List<String>stdTokens=lexer(this.stdProgram);

Set<Integer>proHashValue=new HashSet<Integer>();

Set<Integer>stdHashValue=new HashSet<Integer>();

int proSize=proTokens.size();

int stdSize=stdTokens.size();

for(int i=0;i<proSize-m;i++){

for(int j=i;j<i+m;j++){

proHashValue.add(this.getHashValue(proTokens.get(j)));

}

}

for(int i=0;i<stdSize;i++){

for(int j=i;j<i+m;j++){

stdHashValue.add(this.getHashValue(stdTokens.get(j)));

}

}

Set<Integer>resultSet=new HashSet<Integer>();

int proHashSize=proHashValue.size();

int retainSize=proHashValue.retainAll(stdHashValue).size();

return (float)retainSize/proHashSize;

}

## 本章小结

本章首先介绍了数据库的设计与实现，给出了数据库关键表的详细结构，同时采用ER图来说明各个数据表之间的联系与对应关系。接下来，对系统的功能模块进行全面的说明，包括考生相关的功能模块——在校考试，教师相关的功能模块——题目录入和考试模板管理等内容，同时给出了各个部分的实现界面。最后对系统的关键模块——自动组卷和自动评阅试卷模块进行了详细的解释，通过流程图的方式分析涉及到的算法的流程并给出了核心代码。

目前基于回溯和遗传算法的自动组卷算法复杂度较高，比较耗时，因此我们在随机抽取算法的基础上进行组卷，教师需要事先给定比较详细的试卷模板，然后系统会在已有试卷模板的基础上进行随机组卷，虽然教师会承担一部分工作量，但是大大加快了系统组卷的速度以及效果。

自动评阅试卷分为客观题的评阅和主观题的评语，对于客观题则是简单的进行答案匹配；而对于主观题也就是程序设计题，综合了程序运行结果和程序分析，如果程序能够正常运行，则按照通过测试数据的比例给定得分；否则对程序进行分析，按照提交程序与标准程序的匹配度进行打分，有效避免了由于语法错误导致得分为0情况的发生，提高了考试的公平性。

# 总结与展望

本文介绍了基于Java技术的计算机语言在线考试系统，该系统是严格按照软件工程规定的软件项目的开发流程，最终实现了一个专门针对计算机语言的在线考试系统。传统计算机类考试的题目分为客观题和主观题两大类，其中客观题就是有固定答案的题目，而主观题主要是程序设计题，学生需要通过代码的方式来答题，因此本系统目前也针对客观题和主观题两大题型，与目前常见的在线考试系统不同的是，本系统不仅能够对客观题进行自动评阅而且能够对主观题也能够进行自动评阅。对于主观题，系统会对考生提交的源代码进行编译，如果能够正常运行，则通过给定测试用例的通过率计算得分；否则对程序进行语法分析来确定得分，避免了因为一些语法错误而导致得分为0情况的发生，使得系统评阅的方式更加人性化和公平化。同时系统会根据教师制定的考试模板自动生成试卷，不仅大大降低了教师的工作量，而且减低了由于相邻考生考试作弊现象的发生，使得教师能够更加全面了解学生对知识点的掌握程度。本系统采用MVC三层架构，综合利用了Struts2，Spring，Hibernate等框架技术，后台采用MySQL数据库，建立了基于B/S架构的计算机语言在线考试系统。本文主要完成的工作总结如下：

分析传统考试存在的问题和缺点，以此来提出在线考试系统的基本概念，并对比了传统考试和在线考试，确定在线考试开发的目的与意义。

调研本系统实验所涉及到的技术和相关的工具。

对系统的可行性和需求进行调研和分析，确定系统的大体功能，并通过用例图进行建模以确定系统的各个功能模块和概要设计。

完成对数据库的设计以及数据表的设计，并通过ER图的方式分析实体之间的联系和对应关系。对系统的各个功能模块进行详细设计，通过流程图的方式确定系统的业务处理流程，最后对系统各个模块进行开发和测试，保证了系统的质量要求，满足了用户的功能性和性能方面的要求。

设计并完成了系统自动组卷和自动评阅试卷，并给出了关键模块的流程图和核心代码，大大提高了考试的效率。

本文的在线考试系统的设计与实现，加深了我对软件工程的理解同时也掌握了一些比较先进的技术和算法，同时能够对相关的技术进行熟悉的应用。在开发过程中也遇到了很多问题，查阅了大量的资料，提高了自身查阅文献和解决问题的能力。由于个人精力、时间等一系列限制，目前该系统基本具备了一个在线考试系统应该具备的功能，但是随着系统的使用会有一些问题不断的涌现出来，因此需要对系统不断进行改进和完善，主要涉及一下几个方面：

系统组卷算法的改进和完善。目前本系统的组卷算法需要教师和随机组卷算法的综合作用，教师需要给出试卷的详细结构，工作量相对比较大。因此在后续开发工作中，考虑一种新的组卷算法，教师只需要给出试卷的大体要求而不是详细结构。

系统并发性。目前由于用户规模的限制，系统在大规模用户条件下的并发情况并未经过详细的测试。因此随着学生规模的增大，对系统的并发性和实时性要求更高，系统也需要不断进行改进以适应用户规模的扩增。

系统安全性的完善。目前系统只考虑了数据库以及权限等方面的控制，随着系统的使用，一些潜在的安全问题会逐步显现，因此需要考虑更多可能存在安全威胁的方面，做好相关部分的完善。

参考文献

1. Qiao-fang Z, Yong-fei L. Research and development of online examination system[C]//Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012). 2012: 936-938.
2. 奎晓燕, 杜华坤, 刘卫国. 大学计算机基础课程考试方式改革的探索与实践[J]. 中国教育信息化, 2014, 6: 39-41.
3. Nie H. Design and Development of the Online Examination System Based on B/S Structure[C]// International Conference on Teaching and Computational Science. 2014.
4. Canglin W. The Analsis and Comparison Between Browser/Server Structure and Client/Server Structure [J][J]. COMPUTER STUDY, 1999, 5: 025.
5. Krasner G E, Pope S T. A description of the model-view-controller user interface paradigm in the smalltalk-80 system[J]. Journal of object oriented programming, 1988, 1(3): 26-49.
6. 戴翔宇.基于MVC模式的Struts框架的研究与应用[D].武汉理工大学,2003.DOI:10.7666/d.y519533.
7. 余银山, 胡正华. 基于Struts2框架的Web应用开发研究[J]. 电脑知识与技术, 2009, 5(16):4224-4225.
8. 胡启敏, 薛锦云, 钟林辉. 基于Spring框架的轻量级J2EE架构与应用[J]. 计算机工程与应用, 2008, 44(5):115-118.
9. Wu Q, Hu Y, Wang Y. Research on data persistence layer based on hibernate framework[C]//Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on. IEEE, 2010: 1-4.
10. MySQL A B. MySQL[J]. 2001:11-19.
11. Pohl K. Requirements engineering: fundamentals, principles, and techniques[M]. Springer Publishing Company, Incorporated, 2010
12. Jacobson I. Object-Oriented Software Engineering—A Use Case Drive Approach[C]// TOOLS 1993:, International Conference on Technology of Object-Oriented Languages and Systems, Versailles, France, Europe. DBLP, 1993:333.
13. Chen P P. The entity-relationship model: toward a unified view of data[J]. very large data bases, 1975, 1(1): 9-36.
14. Naik K, Sule S, Jadhav S, et al. Automatic Question paper Generation System using randomization algorithm[J]. International Journal of Engineering and Technical Research (IJETR), 2014, 2(12).
15. 龚完全. 基于最小回溯代价的智能组卷算法[D]. 湖南大学, 2005.
16. Suthar U H, Rais A, Upadhyay A, et al. Online Examination Management System Using Genetic Algorithm[J].
17. Beierle C, Kulaa M, Widera M. Automatic Analysis of Programming Assignments[C]//DeLFI. 2003: 144-153.
18. Singh R, Gulwani S, Solarlezama A, et al. Automated feedback generation for introductory programming assignments[J]. programming language design and implementation, 2013, 48(6): 15-26.
19. Li J, Pan W, Zhang R, et al. Design and implementation of semantic matching based automatic scoring system for C programming language[C]. international conference on e learning and games, 2010: 247-257.
20. Shamsi F A, Elnagar A. An Intelligent Assessment Tool for Students’ Java Submissions in Introductory Programming Courses[J]. Journal of Intelligent Learning Systems and Applications, 2012, 04(01): 59-69.
21. Li J, Pan W, Zhang R, et al. Design and implementation of semantic matching based automatic scoring system for C programming language[J]. Entertainment for Education. Digital Techniques and Systems, 2010: 247-257.

致 谢