目录

[1. 开发背景 1](#_Toc10068)

[2. 系统分析 1](#_Toc28841)

[2.1 需求分析 1](#_Toc13803)

[2.2 技术分析 1](#_Toc14674)

[3. 系统概述 2](#_Toc20677)

[3.1 目的 2](#_Toc5089)

[3.2 开发工具和运行环境 2](#_Toc8925)

[3.2.1 开发工具技术 2](#_Toc20203)

[3.2.2 MVC架构模式 3](#_Toc23024)

[4. 系统总体结构 3](#_Toc31943)

[4.1 系统文件组织结构 3](#_Toc13374)

[4.2 数据库分析设计 4](#_Toc10969)

[4.3 数据库概念信息 4](#_Toc14647)

[4.4 系统权限划分 8](#_Toc7168)

[4.4.1 普通用户 8](#_Toc23717)

[4.4.2 管理员用户 9](#_Toc26531)

[4.5 系统功能划分 9](#_Toc2069)

[4.5.1 用户模块 9](#_Toc10734)

[4.5.2 管理员模块 9](#_Toc9350)

[4.5.3 在线评测系统模块 9](#_Toc5955)

[4.5.4 题库模块 10](#_Toc20145)

[4.5.5 比赛模块 10](#_Toc5098)

[4.5.6 排名模块 10](#_Toc29281)

[4.5.7运行模块 10](#_Toc14981)

[5 系统演示 10](#_Toc11699)

[5.1 系统首页 10](#_Toc20174)

[5.2 练习页面 11](#_Toc9017)

[5.3 题目分类 12](#_Toc18352)

[5.4 题目运行结果 13](#_Toc23126)

[5.5 比赛页面 14](#_Toc12983)

[5.6 排名信息页面 14](#_Toc1864)

[5.7 系统介绍页面 15](#_Toc1128)

[5.8 用户登陆个人信息 15](#_Toc8382)

[5.9 后台管理平台 16](#_Toc17417)

[6 工具类 16](#_Toc5873)

[6.1 Spring工具类 16](#_Toc14035)

[6.2 日期工具类 17](#_Toc24717)

[7 开发技巧和难点分析 18](#_Toc3507)

[7.1 多用户 18](#_Toc29686)

[7.2 逻辑判断 18](#_Toc29179)

[7.3 异常处理 18](#_Toc14257)

[7.4 动态配置 18](#_Toc28852)

[致谢 19](#_Toc2795)

**[参考文献](#_Toc371)** [19](#_Toc371)

ACM在线评测系统的设计与实现

网络技术学院 网络编程 刘雪岗 20131106103

指导教师 玉霞

摘要 ACM在线评测系统采用MVC架构，以mysql作为后台管理数据库，Java作为服务器端语言进行开发。在分析系统功能的基础上，着重阐述了该系统设计与实现的关键技术。该系统可供用户查看题目、在线解答题目、在线比赛测试、编译程序判断解题的正确性。以及用户的排名信息、编译错误信息、运行状态信息。该系统实际运行稳定、可靠，为用户提高编程提供了一种有效的管理途径。

关键字 在线评测；编译原理 ；

1. 开发背景

随着互联网技术和计算机技术的发展，对计算机人员的要求越来越高。为了更好的适应这一现象，通过评测系统与专业知识相结合来提高自身编程水平。突破课堂教学内容与教学方法，有效培养用户的学习能力，创新意识。对社会提供所需要的专业技术人才。

1. 系统分析
   1. 需求分析

用户可以在系统上查看并进行练习，当用户针对某一个题号编写了程序以后,在在线评测系统上将程序代码提交到服务器端。服务器接收到提交信息后，将该程序的相应题目的信息等传送到后台测试平台，后台根据信息对提交上来的程序代码进行编译执行，并做出相应判断，将测试结果返回给前端供用户查看。

在用户每次提交练习代码后，服务端将对其进行相应的判断，并将该题的通过率修改，用户在答题时，可以看到该题目的提交率。用户可以用不同的语言来进行解答题目，服务器端将代码运行在服务器端，判断是否可以正确执行，然后进行数据比对，比对成功后，题目才算解答成功。

* 1. 技术分析

本系统是基于Java、MySQL、Tomcat等技术建立的在线评测系统，其目标是对计算机爱好者提供一个学习编程语言的平台，并可以加强自身的基础知识，逻辑能力等。为信息化时代培养所需要的人才。本系统包括对题目的管理、分类、答题情况、用户信息、后台管理、运行等多个模块。整个项目采用MVC架构，使用SpringMVC+Mybatis+Beetl+Tomcat+Mysql技术所实现的在线评测系统。整体开发采用了分层结构，从而利于系统后期的维护。

利用轻量级的Java开发框架Spring来实现JavaSE/EE full-stack（一站式）开发。它具有方便解耦，简化开发、AOP编程支持、声明事物的支持、方便程序的测试、方便集成各种优秀的框架、降低Java EE API的使用难度等的特点。[1]利用Sping的Junit构建简单的单元测试，进行程序的初始测试，从而使后续程序正常进行，减少程序初始阶段的错误，从而节省了开发时间。

持久层框架运用了Mybatis框架，它具有高性能，灵活，大数据，可优化等特点，而且可以直接将开发人员所写的sql语句，进行数据库操作。MyBatis的基础组成、配置、映射器、动态SQL与Spring整合，可以使用注解，大大减少了开发的工作量，也减少了一些配置信息，利用spring的依赖注入，使得程序开发更加便捷。[2]

对于一个系统来说，同样至关重要的前端技术也影响着系统的功能。前端对于产品以及和作为产品开发人员来说，仍然是费力费时的一块。如果采用的前端技术不恰当，将抵消后台优化带来的用户体验，将成倍增加开发和维护时间。尽管我们都意识到此问题，但可选方案几乎没有。Beetl作为新一代的模板技术，相比与其它模板技术自己独特的语法来说，无疑提高了学习效率和开发效率。所以，在前端模板引擎展示时，采用了该技术。

1. 系统概述
   1. 目的

在线评测系统是对于计算机爱好者提供的一个学习锻炼的平台，该系统将生活中所遇到实际问题转换成编程题目。用户在开始解题时，先对题目进行分析，分析完成后，对其进行解答，最后，将代码提交。提高用户分析能力，解决问题的能力。

* 1. 开发工具和运行环境
     1. 开发工具技术

系统开发工具：IDEA

数据库管理系统软件：MySQL5.5

Java开发包：JDK1.6以上

Web服务器：Tomcat6.0以上

前台模板引擎：Beetl

后台模板框架：Easy-ui

前端技术：jQuery、css、html5

系统框架：SpringMVC+Mybatis+Shiro

运行环境：Windows7/Windows8/Windows10

* + 1. MVC架构模式

显示层：使用JSP技术开发

控制层：使用SpringMVC技术开发

模型层：使用JavaBean技术开发

1. 系统总体结构
   1. 系统文件组织结构

在项目开始编写时，先将各个业务模块建立相应的包，这样方便网站开发，使得网站的整体结构规范化，合理性。本系统的文件组织结构图，如下图4-1-1所示。

|  |
| --- |
|  |
|  | C:\Users\Administrator\AppData\Local\Temp\ksohtml\wps44C3.tmp.jpg |

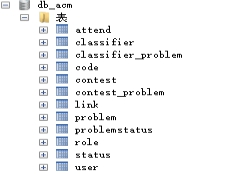
**图4-1-1 代码组织架构图**

整体的代码布局，进行多次更改，最终将代码合理的分层，按功能模块将整个项目分成Java代码、测试代码和静态页面等几个部分。整体的项目使用统一的字符编码UTF-8格式，可以使中文正确的显示，防止代码或者其它地方乱码的出现。

在shiro包模块中，拦截所有的未登陆用户，在用户未登录时且点击代码提交后，直接拦截用户，直接跳转到登陆界面。

* 1. 数据库分析设计

根据本系统所要实现的内容，采用MySQL5.5作为后台数据库，数据库名为db\_acm,字符编码统一为utf-8字符集，与开发编码统一，防止后期乱码的出现。数据库中一共包含12张表，用来存储不同的业务信息，具体表信息如下图4-2-1图所示



**图4-2-1 数据库信息**

* 1. 数据库概念信息

本系统设计了12个实体，分别包含了用户实体，题目实体，状态实体，分类实体，比赛实体等。本系统的数据库名为db\_acm,其中具体表的描述如下表4-3-1：

**表 4-3-1数据库表描述**

|  |  |  |
| --- | --- | --- |
| 数据库表名 | 表名称 | 表描述 |
| user | 用户信息表 | 存储用户个人信息 |
| problem | 题目信息表 | 存储题目具体信息 |
| role | 用户角色信息表 | 存储用户及角色信息 |
| contest | 比赛信息表 | 存储比赛描述信息 |
| contest\_problem | 题目与比赛关联表 | 存储比赛中的题目信息 |
| link | 友情链接表 | 存储网站友情链接信息 |
| status | 提交状态表 | 存储代码运行状态信息 |
| problemstatus | 题目与问题关联表 | 存储题目的状态信息 |
| code | 提交代码表 | 用户提交代码信息表 |
| classifier | 题目分类表 | 题目信息分类描述 |
| classifier\_problem | 问题分类关联表 | 分类中的题目信息 |
| attend | 比赛统计信息表 | 存储比赛统计的信息 |

以下详细描述了各个表的设计和字段说明：

（1）在线评测用户表（user）

**表 4-3-2用户信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | uid | bigint | 20 | 用户标识 |
| 2 | username | varchar | 100 | 用户名 |
| 3 | email | varchar | 100 | 用户邮箱 |
| 4 | submit | int | 11 | 用户提交总数 |
| 5 | accepted | int | 11 | 用户通过总数 |
| 6 | solved | int | 11 | 用户解决题目总数 |
| 7 | defunct | char | 1 | 是否显示 |
| 8 | password | varchar | 100 | 用户密码 |
| 9 | createtime | datetime |  | 用户创建时间 |
| 10 | nick | varchar | 100 | 昵称 |
| 11 | school | varchar | 100 | 学校 |

（2）在线评测用户权限表（role）

**表 4-3-3用户权限表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | rid | bigint | 20 | 权限标识 |
| 2 | type | varchar | 10 | 用户权限类型 |
| 3 | uid | bigint | 20 | 用户标识 |
| 4 | username | varchar | 100 | 用户名称 |

（3）在线评测题目信息表（problem）

**表 4-3-4题目信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | pid | bigint | 20 | 问题标识 |
| 2 | title | varchar | 200 | 问题标题 |
| 3 | description | text |  | 问题描述 |
| 4 | input | text |  | 问题输入描述 |
| 5 | output | text |  | 问题输出描述 |
| 6 | sampleInput | text |  | 问题实例输入 |
| 7 | sampleOutput | text |  | 问题实例输出 |
| 8 | hint | text |  |  |
| 9 | source | varchar | 100 | 题目来源 |
| 10 | sampleCode | text |  | 实例代码 |
| 11 | createTime | datetime |  | 创建时间 |
| 12 | timeLimit | int | 11 | 运行时间最大限制 |
| 13 | memoryLimit | int |  | 运行大小最大限制 |
| 14 | defunct | char | 1 | 是否显示 |
| 15 | contestId | bigint | 20 | 比赛标识 |
| 16 | accepted | int |  | 通过总数 |
| 17 | submit | int |  | 提交总数 |
| 18 | ratio | float |  | 通过与提交比例 |
| 19 | error | int | 11 | 错误次数 |
| 20 | difficulty | tinyint | 4 | 难度等级 |
| 21 | submitUser | bigint | 20 | 通过用户 |
| 22 | solved | int | 11 | 解决总数 |

（4）题目状态表（problemstatus）

**表 4-3-5题目状态表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | sid | bigint | 20 | 运行状态标识 |
| 2 | uid | bigint | 20 | 用户标识 |
| 3 | pid | bigint | 20 | 问题表示 |
| 4 | username | varchar | 100 | 用户名 |
| 5 | result | int | 11 | 结果 |
| 6 | time | int | 11 | 运行时间 |
| 7 | memory | int | 11 | 运行大小 |
| 8 | language | int | 11 | 运行语言 |
| 9 | submittime | datetime |  | 提交时间 |
| 10 | code | text |  | 代码 |

（5）在线评测友情链接表（link）

**表 4-3-6友情链接表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | id | int | 11 | 链接标识 |
| 2 | name | varchar | 150 | 链接名称 |
| 3 | url | varchar | 150 | 链接地址 |
| 4 | type | varchar | 150 | 链接类型 |

（6）代码运行状态表（status）

**表 4-3-7状态表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | sid | bigint | 20 | 运行状态标识 |
| 2 | uid | bigint | 20 | 用户标识 |
| 3 | pid | bigint | 20 | 问题标识 |
| 4 | username | varchar | 100 | 用户名 |
| 5 | result | int | 11 | 运行时间 |
| 6 | memory | int | 11 | 运行大小 |
| 7 | language | int | 11 | 运行语言 |
| 8 | submittime | datetime |  | 提交时间 |
| 9 | code | text |  | 代码 |

（7）题目分类表（classifier）

**表 4-3-8题目分类表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | cid | bigint | 20 | 分类标识 |
| 2 | title | varchar | 100 | 分类标题 |
| 3 | createTime | datetime |  | 分类创建时间 |
| 4 | modifyTime | datetime |  | 分类修改时间 |

（8）问题分类管理表（classifier\_problem）

**表 4-3-9题目分类表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | cpid | bigint | 20 | 分类与问题关联标识 |
| 2 | pid | bigint | 20 | 问题标识 |
| 3 | cid | bigint | 20 | 分类标识 |

（9）比赛信息表（contest）

**表 4-3-10比赛信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | cid | bigint | 20 | 比赛标识 |
| 2 | title | varchar | 255 | 比赛标题 |
| 3 | startTime | datetime |  | 开始时间 |
| 4 | endTime | datetime |  | 结束时间 |
| 5 | defunct | char | 1 | 是否显示 |
| 6 | description | text |  | 比赛描述 |

（10）比赛题目信息表（contest\_problem）

**表 4-3-11题目比赛关联表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | cpid | bigint | 20 | 比赛问题关联标识 |
| 2 | cid | bigint | 20 | 比赛标识 |
| 3 | pid | bigint | 20 | 问题标识 |
| 4 | title | varchar | 255 | 标题 |
| 5 | num | int | 11 | 编号 |

（11）用户提交代码信息表（code）

**表 4-3-12代码提交表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | sid | bigint | 20 | 代码标识 |
| 2 | code | text |  | 代码 |

（12）比赛信息统计表（attend）

**表 4-3-13比赛统计表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 长度 | 中文名 |
| 1 | uid | char | 20 | 用户标识 |
| 2 | cid | int | 11 | 比赛标识 |
| 3 | accepts | int | 11 | 通过总数 |
| 4 | penalty | int | 11 | 提交错误加时 |
| 5 | A\_time | int | 11 | A语言时间 |

* 1. 系统权限划分
     1. 普通用户

作为普通用户登陆后，导航右侧只有用户登陆信息和退出按钮，无后台管理入口按钮。普通用户只拥有前端页面的题目信息查看，基本资料查看，自己提交的代码查看。普通用户导航信息如下图4-4-1：

C:\Users\Administrator\AppData\Local\Temp\ksohtml\wps44E5.tmp.jpg

**图4-4-1 普通用户首页导航**

* + 1. 管理员用户

作为管理员用户，导航右侧拥有后台管理入口按钮，点击按钮后进入后台管理页面，从而开始进行系统管理。管理员导航信息如下图4-4-2：

C:\Users\Administrator\AppData\Local\Temp\ksohtml\wps44E6.tmp.jpg

**图4-4-2 管理员用户首页导航**

* 1. 系统功能划分
     1. 用户模块

用户分为管理员用户和普通用户，对于普通用户，可以登陆前台，可以对题目浏览、答题、查看用户的排名信息，提交信息，还可以修改自己的个人信息资料。管理员可以登陆前台后台，并对于题目、用户、分类、考试等信息，进行管理。

* + 1. 管理员模块

管理员与普通用户使用shiro技术实现权限控制，首先对于登陆的用户，获取该用户所拥有的角色，然后再根据角色分配他拥有的权限，从而进行对于用户的管理，该技术是用Java实现的框架，是一个简单易用的API提供身份验证和授权的实用性框架。Shiro对于该系统提供了安全性而且还无需从头编写所有代码。它主要包涵认证：用户身份识别，常被称为用户“登录”；授权 访问控制；密码加密：保护或隐藏数据防止被偷窥；会话管理：每个用户相关的时间敏感的状态。Shiro还支持一些辅助特性，如Web应用安全、单元测试和多线程，它们的存在强化了上面提到的四个要素。[3]在参考及查阅了一些资料后使用了该框架来管理该系统的权限控制。

* + 1. 在线评测系统模块

用户提交答案到服务器时，服务器接收用户所提供的题目编号，以及所提交的信息，服务端根据题目编号将所提交的信息进行与用户所提交的语言进行编译，服务端根据所配置的编译命令，进行编译，并在本地生成缓存文件，然后再用所提供的编号与服务器中的输入信息所对应，看是否与输出文件对应，如果对应，则返回正确信息，否则返回错误信息，最终将正确或错误的信息返回给前台进行展示，将状态记录更新显示。

将用户提交的代码进行编译，生成可执行的缓存文件，在将预先设置好的输入信息输入到可执行文件中，看输出结果与预定义的输出信息是否一致。

* + 1. 题库模块

随着软件和互联网技术的兴起，出现了一些网络题库。实际上它们是把试题集合放到数据库中管理，利用信息化的优势来组织试题供学习使用。

在线评测系统的题目信息是由管理员登陆后台，进行题目管理，进行题目添加，添加成功后，并将题目显示权限打开，使题目在前台显示，供用户进行查看，答题等。

4.5.5 比赛模块

比赛模块是对用户的每个阶段的测试，添加一些新题目到库中，供用户答题，实现对用户间断性的测试，看用户近期的成绩。

4.5.6 排名模块

排名模块是用户答题的总体情况信息，按答题数量，答题时间等排序。

4.5.7运行模块

运行模块是用户提交答题资料到服务端，服务端进行严重并返回答题信息，将答题信息的正确性展示给用户，看用户是否解决了问题情况等。

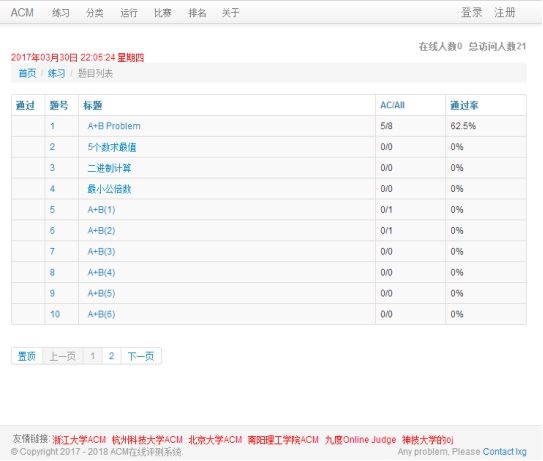
1. 系统演示
   1. 系统首页



**图5-1-1 系统首页**

首页主要是介绍ACM在线评测系统的由来，以及对现代计算机产生的影响。导航信息分别为题目练习、题目分类、题目运行状态、题目比赛信息、比赛排名信息入口页面。导航右侧有用户登陆、用户注册按钮，供用户注册登陆使用，从而开始进行题目练习。

* 1. 练习页面



**图5-2-1 题目练习**

练习题目列表页面主要为显示练习的标题 、难易程度、通过率。通过状态是根据用户登陆后，判断用户是否通过相应的题目，若通过，则显示‘对号’，否则，显示‘X‘，若从未提交过该题目，则显示为空。在题目下方，显示列表分页，使用户找到相应的题目。

* 1. 题目分类



**图5-3-1 题目分类**

题目的分类信息页面是将题目按类型整理到相应的分类下，可使用户做同一个类型的题目，加强用户对于同一类型题目的解题能力。

* 1. 题目运行结果



**图5-4-1 运行状态信息**

运行状态列表页面，是用户实时提交代码后，反馈到页面，使用户看到他所提交的代码是否通过。

* 1. 比赛页面



**图5-5-1 比赛信息**

主要用户在一段时间，进行比赛或者测试等来提高学生们的答题能力。

* 1. 排名信息页面



**图5-6-1 排名信息**

显示用户答题的具体排名

* 1. 系统介绍页面



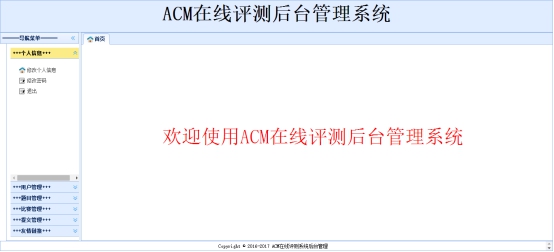
**图5-7-1 系统介绍**

* 1. 用户登陆个人信息



**图5-8-1 用户个人信息**

* 1. 后台管理平台



**图5-9-1 后台管理**

1. 工具类
   1. Spring工具类

**public class** UtilToSpring **implements** ApplicationContextAware {  
 **private static** ApplicationContext *applicationContext* = **null**;  
 **public void** setApplicationContext(ApplicationContext applicationContext)  
 **throws** BeansException {  
 **this**.*applicationContext* = applicationContext;  
 }  
 **public static** ApplicationContext getApplicationContext() {  
 **return** *applicationContext*;  
 }  
 **public static** <T> T getBean(String beanName) {  
 **return** (T) *applicationContext*.getBean(beanName);  
 }  
 **public static** <T> T getBean(Class<?> clazz) {  
 **return** (T) *applicationContext*.getBean(clazz);  
 }  
}

在程序中需要动态的根据Bean的id来获取Bean实例，在Spring容器中注入的Bean过多时，按Bean的id来获取时，用Java特性和Spring的特性写一个工具类，来简化开发任务。

* 1. 日期工具类

**public class** DemoDate {  
 **public static** String dateToFormat(Date date, String format){  
 String result=**""**;  
 SimpleDateFormat sdf=**new** SimpleDateFormat(format);  
 **if**(date!=**null**){  
 result=sdf.format(date);  
 }  
 **return** result;  
 }  
 **public static** Date stringToFormat(String str,String format) **throws** Exception{  
 **if**(StringUtil.*isEmpty*(str)){  
 **return null**;  
 }  
 SimpleDateFormat sdf=**new** SimpleDateFormat(format);  
 **return** sdf.parse(str);  
 }  
 **public static** String nowDate()**throws** Exception{  
 Date date=**new** Date();  
 SimpleDateFormat sdf=**new** SimpleDateFormat(**"yyyyMMddhhmmss"**);  
 **return** sdf.format(date);  
 }  
}

对于时间这个问题，有多种格式，为了统一时间格式，生成一个可以相互转换的公用方法，来实现时间显示的问题。

1. 开发技巧和难点分析
   1. 多用户

由于本系统是多人同时登陆，进行题目提交获取提交状态，提交信息到后台时，出现的情况较多，可能某些用户的代码判断逻辑等影响编译时间，导致后提交的用户编译时间较长，所以引进了Java线程池，它将处理过程中的任务添加到队列，然后在创建线程后自动启动这些任务。在队列满的情况下，进行等待状态，若有空闲队列，将等待中任务放入队列中，来实现多人提交信息时带来的问题。

* 1. 逻辑判断

评测系统判断用户提交是否正确，是与本地的输入、输出文件进行对比，所以需要对文件流等进行多种操作，运用缓存等将编译结果保存到内存中，将输入文件运行到缓存的命令中，在将输出结果与本地的输出文件进行比对。

* 1. 异常处理

有些判断没有考虑周全，从而产生了很多异常，导致系统不能正常工作。

* 1. 动态配置

本系统在properties属性文件中配置了数据库连接的信息、前台模板beetl的配置文件信息、日志输出的配置文件。在xml中配置了spring和编译环境信息。由于编译的环境有多种，所以将各种编译命令，执行命令都动态的加载到xml文件中，在运行时，用dom4j按属性节点等获取所需要的信息。

致谢

本文的完成，首先感谢我的导师玉霞老师，在论文选题中给了我一个很好的提示，使我有方向有目标的进行下去。在玉霞老师的多次审查和建议中，经过最后修改整理，使得论文顺利完成。

其次，也要感谢我在实习单位的几位同事，在生活工作中给予我的关心和教导，使我在技术上有了突破性的进展。感谢我的同事，在工作中给了我很多值得学的东西，感谢我的领导，接纳我，让我有一个学习的机会，从而加强自身的专业知识与技术。感谢他们在我遇到问题时，给我悉心的照顾。在这以后我将努力工作，回馈社会！

**参考文献**

[1] 孟劼. 精通Spring--Java轻量级架构开发实践：人民邮电出版社，2006年10月

[2] 杨开振. 深入浅出MyBatis技术原理与实战：电子工业出版社，2016年9月

[3] Les Hazlewood ，译者 胡伟红. 让Apache Shiro保护你的应用，2011年5月

**The design and implementation of ACM online evaluation system**

Institute of technology of network programming Liu Xuegang 20131106103

Directed by Yu Xia

Abstract ACM online evaluation system USES the MVC architecture to mysql database as a background management, Java as a server-side language development. On the basis of analyzing the system function, emphatically elaborated the system design and realization of key technology. The system is available for users to view the topic, answer questions online, online game test, the correctness of the compiler to judge the problem solving. And user ranking information, compile error information, running state information. The system operation is stable, reliable and improve the programming for the user provides a means of effective management.

**Keywords** Online reviews; Compilation principle;