

**四川工商学院**

**《面向对象课程设计》**

|  |
| --- |
| **单项选择题标准化考试系统设计** |

|  |  |
| --- | --- |
| 学生姓名 | 陶继鑫 |
| 学 号 | 2020109044 |
| 所在学院 | 计算机学院 |
| 专业名称 | 软件工程（本） |
| 班 级 | 20级1班 |
| 任课教师 | 魏全彬 |

四川工商学院

二○二二年六月

单项选择题标准化考试系统

学生：陶继鑫 指导教师：魏全彬

内容摘要：

在网络技术逐渐渗入社会生活各个层面的今天，传统的考试方式也面临着变革，而标准化网络考试是一个很重要的发展方向。基于JavaSwing技术的网络考试系统可以借助于Java虚拟机在任何设备进行，大大拓展了考试的灵活性；可以采用大量标准化试题，使用计算机阅卷可大大提高阅卷效率：试卷也可以根据题库中的内容即时生成，还可以直接把成绩送到数据库中，进行统计、排序等操作。因此，采用标准化考试方式将是以后考试发展的趋势。本单项选择题标准化考试系统页面展示采JavaSwing，用Java语言编写，数据库使用目前主流的MySQL，以B/S架构作为系统开发架构。通过需求分析，该系统包含查看试题库，录入试题，删除试题，修改试题，生成试卷等基本功能。该系统可实现真正意义上的无纸化考试，不仅是建设节约社会的需要，同时计算机考试还具有快捷、方便、高效、安全等许多优点，考试的信度、效度都可大大提高。

关键词：Java、JavaSwing、MySQL、JDBC

目 录

[1 概述 1](#_Toc106562889)

[2 需求分析 1](#_Toc106562890)

[2.2 系统功能需求分析 1](#_Toc106562891)

[3 总体设计 2](#_Toc106562892)

[3.1 系统功能结构设计 2](#_Toc106562893)

[3.2 数据库设计 2](#_Toc106562894)

[3.2.1 数据库概念设计 2](#_Toc106562895)

[3.2.2数据库逻辑设计 3](#_Toc106562896)

[3.2.3数据库物理设计 3](#_Toc106562897)

[4 详细设计 3](#_Toc106562898)

[4.1 登录模块详细设计 4](#_Toc106562899)

[4.2 功能模块详细设计 4](#_Toc106562900)

[5 系统实现 5](#_Toc106562901)

[5.1 登录模块 5](#_Toc106562902)

[5.2 功能模块 7](#_Toc106562903)

[6系统测试 18](#_Toc106562904)

[6.1 登录模块测试 18](#_Toc106562905)

[6.2 录入试题模块测试 18](#_Toc106562906)

[6.3 删除试题模块测试 19](#_Toc106562907)

[6.4 修改试题模块测试 19](#_Toc106562908)

[6.5 生成试题模块测试 19](#_Toc106562909)

[7 总结与展望 19](#_Toc106562910)

[参考文献 20](#_Toc106562911)

[附录：程序的源代码 20](#_Toc106562912)

单项选择题标准化考试系统

# 1 概述

近年来，随着计算机技术的发展，上机考试成为网络研究与应用的热点之一，在国外一些发达国家已经得到蓬勃发展，已成为现代教育的有力补充，非常适合个性化学习。在国内，上机考试还处于试点阶段，传统的考试方式已不再适应线上教育的发展。因此，可以开发一套适合线上教育课程的单选题标准化考试系统，以适应线上教育的发展。该单项选择题标准化考试系统以B/S架构作为系统开发架构。该系统包含查看试题库，录入试题，删除试题，修改试题，生成试卷等基本功能，从而满足上机考试的各种需求。

# 2 需求分析

为了保证能够顺利开发出实用的系统，进行需求分析尤为重要。本章从可行性、系统功能需求方面展开分析。

2.1 可行性分析

2.1.1时间可行性

该系统更加节省了对试题的管理时间，试题管理效率提高。

2.1.2经济可行性

该系统零成本开发，无需任何额外费用，经济可行。

2.1.3技术可行性

该系统完全使用Java实现，界面通过Java swing实现，使用jdbc连接MySQL数据库，技术可行。

2.1.4法律可行性

该系统完全由自己独立自主开发，未使用其他有版权的技术，不构成侵权。

## 2.2 系统功能需求分析

2.2.1管理员用户需求分析

管理员用户可以通过密码登录单项选择题标准化考试系统，然后通过该系统对试题库中的试题进行管理，如查看所有试题、录入新的试题、删除试题、修改试题等操作。

2.2.2 普通用户需求分析

普通用户可以通过该系统自动生成试题，然后进行作答，作答完成后该系统会生成答题报告，显示正常答案以及选择的选项，同时显示正确了多少题，错误了多少题。

# 3 总体设计

首先是一个登录界面，用户登录后才可进行相关的操作，保障的系统的安全可行性。登录成功后进入有多个按钮可选择的页面，该页面可以选择所需要的功能进行相关的操作。有五大功能，分别为查看所有试题模块、录入试题模块、删除试题模块、修改试题模块、生成试题模块。在所有试题模块中，用户可以查看数据库中的所有试题的相关信息，如试题号、试题内容、试题选项以及正确答案等。在录入试题模块中，用户可以输入试题内容、选项等内容录入试题。在删除模块中用户可以通过输入试题号进行删除试题。在生成试题模块中，用户输入抽取试题数，系统随机生成试题，供用户作答，最后生成试题报告，提示用户正确了多少题错误了多少体。

## 3.1 系统功能结构设计

根据系统的需求分析包括以下的功能，如图 3-1所示。

单项选择题标准化考试系统

修改试题

查看题库

删除试题

生成题库

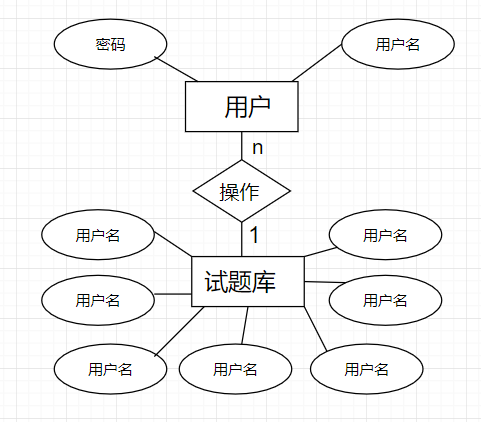
录入试题

**图3-1 系统功能结构图**

## 3.2 数据库设计

### 3.2.1 数据库概念设计

拥有两个实体，分别为用户和试题这两个实体。用户实体有用户名和密码这两个属性，试题这个试题含有试题号、内容、选项1、选项2、选项3、选项4以及正确答案这七个属性，一个用户可以有多个试题库，一个试题库只能有一个用户，用户可以对试题库进行操作。



### 3.2.2数据库逻辑设计

设计了两个表，一个users表，该表有字段username用于存放用户名，userpassword字段用于存放用户密码，该有一个题库表，字段id用于记录题号，字段content为试题内容，字段answer1,answer2,answer3,answer4用于存放选项，字段right\_answer用于存放正确答案。

### 3.2.3数据库物理设计

用户表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 主键 | 备注 |
| username | varchar(10) | 否 | 是 | 用户名 |
| userpassword | varchar(20) | 否 | 是 | 密码 |

题库表：

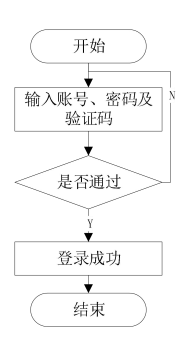
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为空 | 主键 | 备注 |
| id | Int | 否 | 是 | 题号 |
| content | VARCHAR(200) | 否 | 否 | 试题内容 |
| answer1 | VARCHAR(10) | 否 | 否 | 选项1 |
| Answer2 | VARCHAR(10) | 否 | 否 | 选项2 |
| Answer3 | VARCHAR(10) | 否 | 否 | 选项3 |
| Answer4 | VARCHAR(10) | 否 | 否 | 选项4 |
| right\_answer | VARCHAR(10) | 否 | 否 | 答案 |

# 4 详细设计

由登录界面和各个功能界面组成，登录界面，用户登录后才可进行相关的操作，保障的系统的安全可行性。登录成功后进入有多个按钮可选择的页面，该页面可以选择所需要的功能进行相关的操作。有五大功能，分别为查看所有试题模块、录入试题模块、删除试题模块、修改试题模块、生成试题模块。

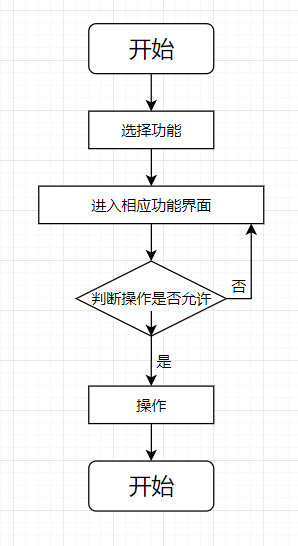
## 4.1 登录模块详细设计

用户在此页面登录。用户输入正确的用户名以及正确的密码使用用户的登录。若为输入用户名或密码点击登录，会提示为输入用户名或密码不能为空，若用户输入错误的用户名或密码，则提示用户名或密码错误，知道输入正确的用户名和密码，则跳转到新的操作页面。用户登录流程图如下：



## 4.2 功能模块详细设计

功能模块为一个单页面，该页面中有多个按钮，每个按钮都会进入一个新的功能，如查看所有试题，录入试题，修改试题，删除试题，以及生成试题等。用户操作功能模块程图如下：

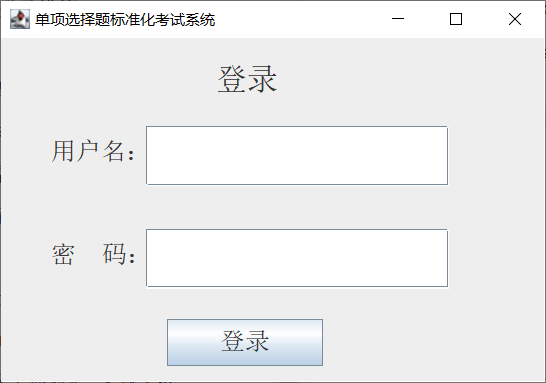


# 5 系统实现

## 5.1 登录模块

（1）登录模块界面的实现

用户在此页面登录。用户输入正确的用户名以及正确的密码使用用户的登录。若为输入用户名或密码点击登录，会提示为输入用户名或密码不能为空，若用户输入错误的用户名或密码，则提示用户名或密码错误，知道输入正确的用户名和密码，则跳转到新的操作页面。如图：



（2）登录模块功能的实现

当点击登录按钮时，会触发登录按钮的点击事件，点击事件中，会获取输入框中的内容，然后进行查询操作。

获取到输入框的内容后，连接数据库，通过查询判断用户输入的用户名和密码与数据库中的用户名和密码是否相同，不相同则进行相应的提示，相同则进行页面跳转。

登录按钮点击事件核心代码：

JButton btn = (JButton)e.getSource();

String str = btn.getText();

**if**(str.equals("登录")) {

String userName = login.textField.getText();

String password = login.passwordField.~~getText~~();

String sqlUserName = **null**;

String sqlUserPassword = **null**;

// 连接数据库

**try**

{

Class.*forName*("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

**catch** (Exception e1)

{

System.***out***.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

**try**

{

Connection connect = DriverManager.*getConnection*( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

Statement stmt = connect.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM users;");

**while** (rs.next())

{

sqlUserName = rs.getString("username");

sqlUserPassword = rs.getString("userpassword");

}

connect.close();

}

**catch** (Exception e1)

{

System.***out***.print("获取数据错误!");

e1.printStackTrace();

}

// 判断密码是否正确

**if** (userName.equals(sqlUserName) && password.equals(sqlUserPassword)) {

login.setVisible(**false**);

Home frame = **new** Home();

frame.setVisible(**true**);

} **else** {

JOptionPane.*showMessageDialog*(**null**, "密码或用户名错误！","提示",JOptionPane.***INFORMATION\_MESSAGE***);

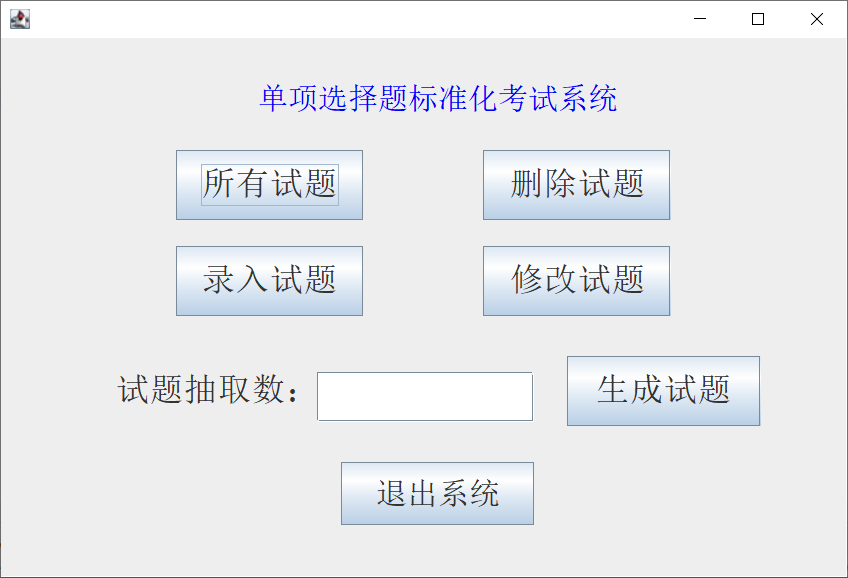
}

}

## 5.2 功能模块

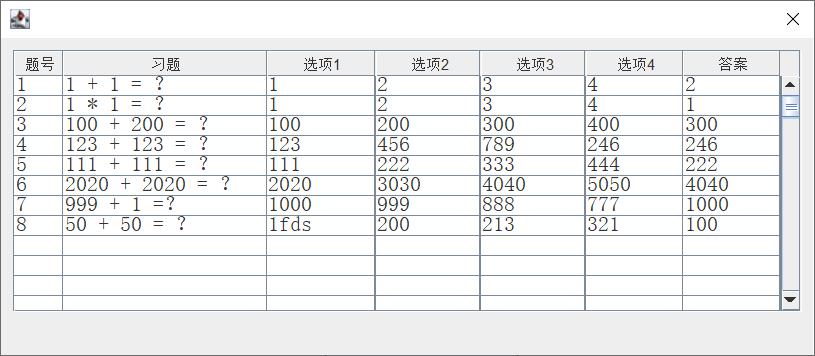
（1）功能模块界面的实现

功能模块界面包含多个按钮，如所有试题按钮、删除试题按钮、录入试题按钮、删除试题按钮、修改试题按钮、生成试题按钮等，每一个按钮都会跳出新的界面和新的功能；用户通过点击相应的按钮进入到相应的功能模块中去。如图：



（2）查看所有试题模块的实现

该模块可以查看题库中的所有试题内容，如试题的题号，试题的内容以及选项和正确答案，还模块通过表格的方式让用户更加清楚的看到该模块的所有内容。如图：



该模块通过JDBC查询到数据库中的试题信息，然后通过循环将获取的信息保存到一个二维对象数组中保存起来，再通过Java swing将试题信息展示出来。核心代码：

// 连接数据库

**try**

{

Class.*forName*("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

**catch** (Exception e1)

{

System.***out***.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

**try**

{

Connection connect = DriverManager.*getConnection*( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

Statement stmt = connect.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM exercises;");

Object[][] data = **new** Object[100][7];

**while** (rs.next())

{

**int** num = rs.getInt("id");

data[num-1][0] = rs.getInt("id")+"";

data[num-1][1] = rs.getString("content");

data[num-1][2] = rs.getString("answer1");

data[num-1][3] = rs.getString("answer2");

data[num-1][4] = rs.getString("answer3");

data[num-1][5] = rs.getString("answer4");

data[num-1][6] = rs.getString("right\_answer");

}

connect.close();

LookExercise dialog = **new** LookExercise(data);

dialog.setVisible(**true**);

}

**catch** (Exception e1)

{

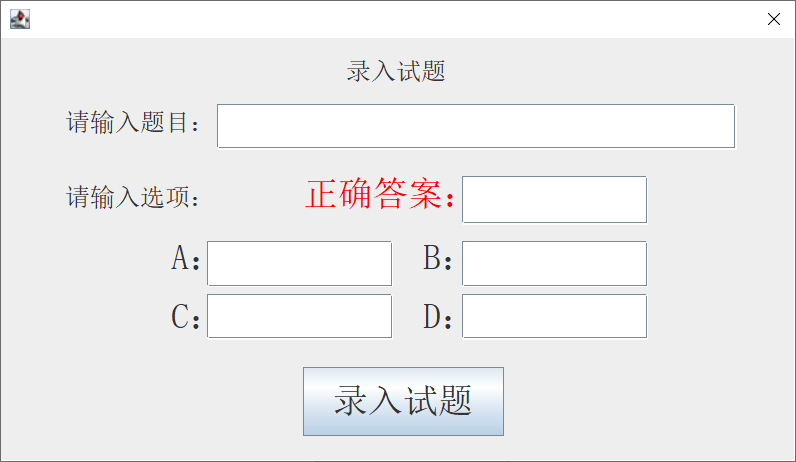
System.***out***.print("获取数据错误!");

e1.printStackTrace();

}

（3）录入试题模块的实现

该模块可以让用户往题库中添加新的试题，在该模块界面中用户需要输入试题的题目，试题的各个选项以及正确的答案，如图所示：



然后通过JDBC连接数据库保存到数据库中，部分代码：

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JButton btn = (JButton)e.getSource();

String str = btn.getText();

// 录入试题按钮

if(str.equals("录入试题")) {

String content = ee.textField.getText();

String answerA = ee.answera.getText();

String answerB = ee.answerb.getText();

String answerC = ee.answerc.getText();

String answerD = ee.answerd.getText();

String rightAnswer = ee.answer.getText();

// 连接数据库

try

{

Class.forName("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

catch (Exception e1)

{

System.out.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

try

{

Connection connect = DriverManager.getConnection( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String sql = "INSERT INTO exercises(content,answer1,answer2,answer3,answer4,right\_answer) VALUES(?, ?,?,?,?,?);";

// 获取pstmt对象

PreparedStatement pstmt = connect.prepareStatement(sql);

// 设置？的值

pstmt.setString(1, content);

pstmt.setString(2, answerA);

pstmt.setString(3, answerB);

pstmt.setString(4, answerC);

pstmt.setString(5, answerD);

pstmt.setString(6, rightAnswer);

// 执行sql语句

int rs = pstmt.executeUpdate();

// 判断是否成功

if(rs == 1){

System.out.println("录入成功~");

JOptionPane.showMessageDialog(null, "录入成功！","成功",JOptionPane.INFORMATION\_MESSAGE);

// 清空文本框

ee.textField.setText("");

ee.answera.setText("");

ee.answerb.setText("");

ee.answerc.setText("");

ee.answerd.setText("");

ee.answer.setText("");

}else{

JOptionPane.showMessageDialog(null, "录入失败！","成功",JOptionPane.INFORMATION\_MESSAGE);

}

connect.close();

}

catch (Exception e1)

{

System.out.print("获取数据错误!");

e1.printStackTrace();

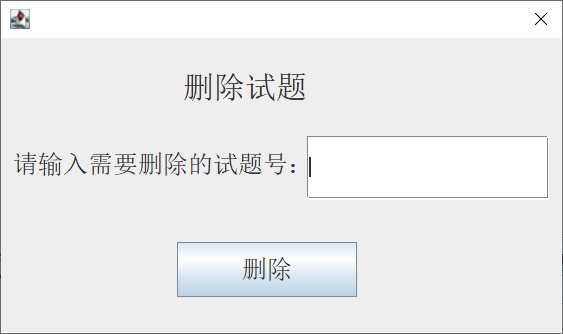
}

}

}

（4）删除试题模块的实现

该模块用户可以通过输入试题号进行对某个试题的删除，若试题不存在则弹窗提示试题不存在，存在则提示删除成功，如图：



部分代码：

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JButton btn = (JButton)e.getSource();

String str = btn.getText();

if(str.equals("删除")) {

String number = de.textField.getText();

if (number.equals("")) {

JOptionPane.showMessageDialog(null, "题号不能为空且必须为数字！","提示",JOptionPane.INFORMATION\_MESSAGE);

} else if(number.equals("0")) {

JOptionPane.showMessageDialog(null, "题号不能为0！","提示",JOptionPane.INFORMATION\_MESSAGE);

} else {

// 连接数据库

try

{

Class.forName("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

catch (Exception e1)

{

System.out.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

try

{

Connection connect = DriverManager.getConnection( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String sql = "DELETE FROM exercises WHERE id=?;";

// 获取pstmt对象

PreparedStatement pstmt = connect.prepareStatement(sql);

// 设置？值

int num = Integer.parseInt(de.textField.getText());

pstmt.setInt(1, num);

// 执行sql语句

int rs = pstmt.executeUpdate();

if (rs == 1) {

JOptionPane.showMessageDialog(null, "删除成功！","提示",JOptionPane.INFORMATION\_MESSAGE);

de.textField.setText("");

} else {

JOptionPane.showMessageDialog(null, "该试题不存在！","提示",JOptionPane.INFORMATION\_MESSAGE);

de.textField.setText("");

}

connect.close();

}

catch (Exception e1)

{

System.out.print("获取数据错误!");

e1.printStackTrace();

}

}

}

}

（5）修改试题模块的实现

该模块可以对某个试题进行修改，如题目、选项以及正确答案的修改，首先输入题号进行查询，查询后便可修改，如图：



部分代码：

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JButton btn = (JButton)e.getSource();//获取点击的按钮

String str = btn.getText();//将按钮的文本值存到str中

if(str.equals("查询")) {

String exeNum = me.textField.getText();

int exerciseNum = Integer.parseInt(exeNum);

// 连接数据库

try

{

Class.forName("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

catch (Exception e1)

{

System.out.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

try

{

Connection connect = DriverManager.getConnection( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String sql = "SELECT \* FROM exercises WHERE id=?;";

// 获取pstmt对象

PreparedStatement pstmt = connect.prepareStatement(sql);

// 设置？值

pstmt.setInt(1, exerciseNum);

// 执行sql语句

ResultSet rs = pstmt.executeQuery();

if (rs.next()) {

me.textField\_1.setText(rs.getString("content"));

me.textField\_2.setText(rs.getString("answer1"));

me.textField\_3.setText(rs.getString("answer2"));

me.textField\_4.setText(rs.getString("answer3"));

me.textField\_5.setText(rs.getString("answer4"));

me.textField\_6.setText(rs.getString("right\_answer"));

} else {

JOptionPane.showMessageDialog(null, "该试题不存在！","提示",JOptionPane.INFORMATION\_MESSAGE);

me.textField\_1.setText("");

me.textField\_2.setText("");

me.textField\_3.setText("");

me.textField\_4.setText("");

me.textField\_5.setText("");

me.textField\_6.setText("");

me.textField.setText("");

}

while (rs.next())

{

me.textField\_1.setText(rs.getString("content"));

me.textField\_2.setText(rs.getString("answer1"));

me.textField\_3.setText(rs.getString("answer2"));

me.textField\_4.setText(rs.getString("answer3"));

me.textField\_5.setText(rs.getString("answer4"));

me.textField\_6.setText(rs.getString("right\_answer"));

}

connect.close();

}

catch (Exception e1)

{

System.out.print("获取数据错误!");

e1.printStackTrace();

}

} else if (str.equals("修改")) {

// 连接数据库

try

{

Class.forName("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

}

catch (Exception e1)

{

System.out.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

try

{

Connection connect = DriverManager.getConnection( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

String sql = "UPDATE exercises SET content=?,answer1=?,answer2=?,answer3=?,answer4=?,right\_answer=? WHERE id=?;";

// 获取pstmt对象

PreparedStatement pstmt = connect.prepareStatement(sql);

// 设置？值

pstmt.setString(1, me.textField\_1.getText());

pstmt.setString(2, me.textField\_2.getText());

pstmt.setString(3, me.textField\_3.getText());

pstmt.setString(4, me.textField\_4.getText());

pstmt.setString(5, me.textField\_5.getText());

pstmt.setString(6, me.textField\_6.getText());

pstmt.setInt(7, Integer.parseInt(me.textField.getText()));

// 执行sql语句

int rs = pstmt.executeUpdate();

if(rs == 1) {

JOptionPane.showMessageDialog(null, "修改成功！","提示",JOptionPane.INFORMATION\_MESSAGE);

me.textField\_1.setText("");

me.textField\_2.setText("");

me.textField\_3.setText("");

me.textField\_4.setText("");

me.textField\_5.setText("");

me.textField\_6.setText("");

me.textField.setText("");

} else {

JOptionPane.showMessageDialog(null, "修改失败！","提示",JOptionPane.INFORMATION\_MESSAGE);

}

connect.close();

}

catch (Exception e1)

{

System.out.print("获取数据错误!");

e1.printStackTrace();

}

}

}

（6）做题试题模块的实现

该模块可以自动从题库中随机抽取一定数目的题目，该数目为客户输入的题数，生成题目后会进入答题界面，用户作答完后会生成答题结果的报告，提示正确了多少题，错误了多少题。如图：





部分代码：

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

JButton btn = (JButton)e.getSource();//获取点击的按钮

String str = btn.getText();//将按钮的文本值存到str中

GroupButtonUtils groupBtn = new GroupButtonUtils();

// 获取选择的选项答案

String selectAnswer = groupBtn.getSelectedButtonText(gt.group).split("：")[1];

//判断点击的是哪个按钮

if(str.equals("下一题")) {

// 更改题目

if(num < this.data.length) {

gt.textPane.setText((String) data[num][0]);

gt.answer1.setText("A：" + (String) data[num][1]);

gt.answer2.setText("B：" + (String) data[num][2]);

gt.answer3.setText("C：" + (String) data[num][3]);

gt.answer4.setText("D：" + (String) data[num][4]);

gt.lblNewLabel.setText("第" + (num+1) + "题");

data[num-1][6] = selectAnswer;

num = num + 1;

} else {

data[num-1][6] = selectAnswer;

JOptionPane.showMessageDialog(null, "这是最后一题！","提示",JOptionPane.INFORMATION\_MESSAGE);

}

} else if (str.equals("提交")) {

GradeView gradeView = new GradeView(this.data);

gradeView.setVisible(true);

}

}

# 6系统测试

## 6.1 登录模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试标题 | 登录模块测试 | | | |
| 用例目的 | 测试用户输入用户名与密码是否正确，从而进行后续相应功能的操作。 | | | |
| 测试项 | 测试内容 | 测试步骤 | 预期结果 | 结论 |
| 1 | 登  录 | 只输入用户名或密码，点击登录 | 弹窗提示用户域名或密码不能为空 | 通过 |
| 2 | 输入错误的用户名或密码，点击登录 | 弹窗提示用户名或密码错误 | 通过 |
| 3 | 输入正确的用户名和密码 | 登录成功，跳出新页面 | 通过 |

## 6.2 录入试题模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试标题 | 录入试题模块测试 | | | |
| 用例目的 | 测试用户输入录入试题时是否正确，验证输入内容是否符合要求 | | | |
| 测试项 | 测试内容 | 测试步骤 | 预期结果 | 结论 |
| 1 | 录  入 | 未输入题目 | 弹窗提示题目不能为空 | 通过 |
| 2 | 未输入某个选项 | 弹窗提示选项不能为空 | 通过 |
| 3 | 未输入正确答案 | 弹窗提示答案不能为空 | 通过 |

## 6.3 删除试题模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试标题 | 删除试题模块测试 | | | |
| 用例目的 | 测试用户删除试题时的操作是否正确 | | | |
| 测试项 | 测试内容 | 测试步骤 | 预期结果 | 结论 |
| 1 | 删  除 | 试题号不存在 | 弹窗提示试题不存在 | 通过 |
| 2 | 试题号存在 | 弹窗提示删除成功 | 通过 |

## 6.4 修改试题模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试标题 | 修改试题模块测试 | | | |
| 用例目的 | 测试用户修改试题时的操作 | | | |
| 测试项 | 测试内容 | 测试步骤 | 预期结果 | 结论 |
| 1 | 修  改 | 查询时输入不存在的题号 | 弹窗提示用户该试题不存在 | 通过 |
| 2 | 查询时输入存在的题号 | 弹窗提示用户修改成功 | 通过 |

## 6.5 生成试题模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试标题 | 生成试题模块测试 | | | |
| 用例目的 | 测试用户生成试题的相关操作 | | | |
| 测试项 | 测试内容 | 测试步骤 | 预期结果 | 结论 |
| 1 | 生  成  试  题 | 未输入生成的试题数 | 弹窗提示用户未输入试题数 | 通过 |
| 2 | 未选择选项 | 弹窗提示用户未选择选项 | 通过 |
| 3 | 当为最后一题时继续点击下一题 | 提示这是最后一题 | 通过 |

# 7 总结与展望

这次开发总体适用了面向对象的编程设计思想，通过封装继承多态，完成了此次的单项选择题标准化考试系统。GUI界面适用了Java swing，数据库使用的是MySQL，通过给不同GUI界面中的按钮添加点击事件，实现相应的功能，如查看全部试题的题库，录入新的试题，删除不需要的试题，以及修改有错误的试题，同时还包含根据输入的试题数自动生成试卷，最后做打完再自动阅卷，实现了系统的基本功能。

在实训过程中当然也遇到了很多问题，如GUI界面的设计，为了解决这个问题，采用了windowbuilder插件的使用，它使界面设计更加方便快捷，顺利解决了这个GUI界面的设计问题。

在这一次的实训中，我再一次熟悉使用了Java进行开发，使我更加理解了面向对象的编程思想，用类进行实体封装属性与方法，通过各种类的创建与调用，完成由整体到局部的开发。更加了解了一个小型项目的开发流程，从简单到复杂，将问题拆分。通过这次实训，是我更加坚信，认识重复性的工作没有什么是程序不能完成的，程序能够帮助我们处理大量重复、有规律可循的事情。同时也让我意识到了在生活中不管遇到什么问题，只要坚定低努力下去，探索适合的方法去解决，困难总压不倒我们，我们总能在这个过程中收货很多。

# 参考文献

[1]徐红宇,吴轲.  [基于B/S结构的模拟考试系统的设计与实现](https://kns.cnki.net/kcms/detail/detail.aspx?filename=DZRU201518164&dbcode=CJFD&dbname=CJFD2015&v=TD9MwUYQaxMw2JRIoKyLKNjkwDOPtwhFhSfrlOxPpa2QwquL8cTOpRJyWD7G8u8c)[J]. 电子技术与软件工程. 2015(18)

[2]帅跃飞. 基于 J2EE 的法院培训系统设计与实现[D]. 四川:西南科技大学,2018.

[3]马帅. 论 MySQL 数据库教程开设的必要性[J]. 现代交际, 2016(13):252.

[4]辛甜.  [基于B/S结构的学生信息管理系统的设计与实现](https://kns.cnki.net/kcms/detail/detail.aspx?filename=SDDZ201509068&dbcode=CJFD&dbname=CJFD2015&v=rn8wyCiFew3Jf-YcXv_GVsCmOC_oOCB81xZmoxfxkFRoubcGicOKb63yF7UXBSE-)[J]. 信息技术与信息化. 2015(09)

[5]张莹,蔡晓真.  [多功能通用题库系统](https://kns.cnki.net/kcms/detail/detail.aspx?filename=FZDZ705.004&dbcode=CJFD&dbname=CJFD1997&v=MSuHo45CsQX38ctDBFdrtTCNzcT34x8gvn5FDOtPZ3WbMXxpQw4MRWzuLocqyZfE)[J]. 福州大学学报(自然科学版). 1997(05)

# 附录：程序的源代码

工具类

package www.utils;

import java.util.Enumeration;

import javax.swing.AbstractButton;

import javax.swing.ButtonGroup;

public class GroupButtonUtils {

public String getSelectedButtonText(ButtonGroup buttonGroup) {

for (Enumeration buttons = buttonGroup.getElements(); buttons.hasMoreElements();) {

AbstractButton button = (AbstractButton) buttons.nextElement();

if (button.isSelected()) {

return button.getText();

}

}

return null;

}

}

启动类：

**package** www.main;

**import** www.gui.Login;

**public** **class** MainSystem {

**public** **static** **void** main(String[] args) {

Login login = **new** Login();

login.setVisible(**true**);

}

}

主界面：

package www.gui;

import java.awt.\*;

import java.sql.\*;

import javax.swing.\*;

import www.listener.ListenerEvent;

public class Home extends JFrame {

private JPanel contentPane;

public JTextField enterNum;

public LookExercise dialog1;

public EnterExercise dialog2;

public GenerateTest dialog3;

/\*\*

\* Create the frame.

\*/

public Home() {

Object[][] data;

// 连接数据库

try

{

Class.forName("com.mysql.jdbc.Driver"); //加载MYSQL JDBC驱动程序

System.out.println("成功加载Mysql驱动程序!");

}

catch (Exception e1)

{

System.out.print("加载Mysql驱动程序时出错!");

e1.printStackTrace();

}

try

{

Connection connect = DriverManager.getConnection( "jdbc:mysql://127.0.0.1:3306/jdbcTest?useSSL=false","root","123456");

//连接URL为 jdbc:mysql//服务器地址/数据库名 ，后面的2个参数分别是登陆用户名和密码

System.out.println("成功连接Mysql服务器!");

Statement stmt = connect.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM exercises;");

data = new Object[100][6];

while (rs.next())

{

int num = rs.getInt("id");

data[num-1][0] = rs.getString("content");

data[num-1][1] = rs.getString("answer1");

data[num-1][2] = rs.getString("answer2");

data[num-1][3] = rs.getString("answer3");

data[num-1][4] = rs.getString("answer4");

data[num-1][5] = rs.getString("right\_answer");

}

connect.close();

System.out.println("关闭连接");

// Home frame = new Home(data);

// frame.setVisible(true);

}

catch (Exception e)

{

System.out.print("获取数据错误!");

e.printStackTrace();

}

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 691, 469);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

// 窗口居中

setLocationRelativeTo(null);

ListenerEvent listener = new ListenerEvent(this);//监听器

// 系统名称

JLabel systemName = new JLabel("单项选择题标准化考试系统");

systemName.setForeground(Color.BLUE);

systemName.setFont(new Font("宋体", Font.PLAIN, 24));

systemName.setBounds(206, 30, 302, 40);

contentPane.add(systemName);

// 录入试题

JButton addExercise = new JButton("录入试题");

addExercise.addActionListener(listener);//注册事件

addExercise.setFont(new Font("宋体", Font.PLAIN, 27));

addExercise.setBounds(140, 167, 150, 56);

contentPane.add(addExercise);

// 所有试题

JButton lookExercise = new JButton("所有试题");

lookExercise.addActionListener(listener);//注册事件

lookExercise.setFont(new Font("宋体", Font.PLAIN, 27));

lookExercise.setBounds(140, 90, 150, 56);

contentPane.add(lookExercise);

// 删除试题

JButton deleteExercise = new JButton("删除试题");

deleteExercise.addActionListener(listener);//注册事件

deleteExercise.setFont(new Font("宋体", Font.PLAIN, 27));

deleteExercise.setBounds(386, 90, 150, 56);

contentPane.add(deleteExercise);

// 生成试题

JButton generateTest = new JButton("生成试题");

generateTest.addActionListener(listener);//注册事件

generateTest.setFont(new Font("宋体", Font.PLAIN, 27));

generateTest.setBounds(453, 255, 155, 56);

contentPane.add(generateTest);

// 试题抽取提示

JLabel labelTest = new JLabel("试题抽取数：");

labelTest.setFont(new Font("宋体", Font.PLAIN, 27));

labelTest.setBounds(92, 263, 174, 40);

contentPane.add(labelTest);

// 试题抽取输入

enterNum = new JTextField();

// enterNum.setText("请输入试题数");

enterNum.setToolTipText("");

enterNum.setBounds(253, 268, 174, 40);

contentPane.add(enterNum);

enterNum.setColumns(10);

// 退出系统

JButton exitSystem = new JButton("退出系统");

exitSystem.addActionListener(listener);//注册事件

exitSystem.setFont(new Font("宋体", Font.PLAIN, 25));

exitSystem.setBounds(272, 340, 155, 50);

contentPane.add(exitSystem);

// 修改试题

JButton deleteExercise\_1 = new JButton("修改试题");

deleteExercise\_1.addActionListener(listener);//注册事件

deleteExercise\_1.setFont(new Font("宋体", Font.PLAIN, 27));

deleteExercise\_1.setBounds(386, 167, 150, 56);

contentPane.add(deleteExercise\_1);

// 录入试题对话框

dialog2 = new EnterExercise();

dialog2.setVisible(false);

}

}