**Java程序设计课程实训**

**驾考试题管理系统详细设计**

**版本1.0**

**编写者：赵晨阳**

**日期：2019年12月26日**

**版本历史：**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 文件名 | 文档格式 |
| 2019.10.26 | 1.0 | 驾考试题管理系统详细设计.doc | Word |

目录

[1 前言 4](#_Toc28270161)

[1.1 目的 4](#_Toc28270162)

[1.2 范围 4](#_Toc28270163)

[1.3 定义 4](#_Toc28270164)

[1.4 参考文献 4](#_Toc28270165)

[2 约定和规程 5](#_Toc28270166)

[2.1 设计标准 5](#_Toc28270167)

[2.2 文档标准 5](#_Toc28270168)

[2.3 命名约定 5](#_Toc28270169)

[3 总体设计 6](#_Toc28270170)

[3.1 系统描述 6](#_Toc28270171)

[3.2 总体结构 6](#_Toc28270172)

[3.2.1 系统结构设计 6](#_Toc28270173)

[3.2.2 数据库结构设计 7](#_Toc28270174)

[4 详细设计 8](#_Toc28270175)

[4.1 功能描述 8](#_Toc28270176)

[4.1.1 增加试题 8](#_Toc28270177)

[4.1.2 删除试题 9](#_Toc28270178)

[4.1.3 查询试题 10](#_Toc28270179)

[4.1.4 修改试题 10](#_Toc28270180)

[4.2 模块设计 11](#_Toc28270181)

[4.2.1 登入模块 11](#_Toc28270182)

[5 总结 16](#_Toc28270183)

# 前言

## 目的

本详细设计记录了如何通过数据库来管理一个驾校考试管理系统，实现的有增加一个试题，删除一个试题，修改一个试题，查询一个试题的功能。

本详细设计说明书的预期读者包括：用户。

## 范围

本文档包括项目的要求，实现概况，数据库的结构，以及设计思想。

## 定义

**表** 1.1 定义，缩写词表

|  |  |
| --- | --- |
| DERBY | Apache Derby |

## 参考文献

[1]《Java语言程序设计与数据结构（11版）》， 梁勇著。

[2]《数据库系统概论（第5版）》，王珊著。

# 约定和规程

## 设计标准

采用面向对象设计的设计方法，使用java语言描述，支持面向对象语言编程开发，如Java。

## 文档标准

文档使用Word编写，封面标题采用二号、加粗、宋体字体。章标题采用三号、加粗、宋体字体，节标题采用四号、加粗、宋体字体，正文内容采用小四、宋体字体，表格采用五号、宋体字体，表标和图标编号采用五号、宋体字体，代码段采用五号、宋体字体、按照Java编程规范编写。

## 命名约定

表2-1 命名约定

|  |  |  |
| --- | --- | --- |
| 类型 | 命名约定 | 扩展名 |
| 注释 | 代码中注释可以使用单行注释或多行注释，注释内容要与注释项对应，注释内容写在要解释的代码前，注释前有代码的，应与前面代码有一行或两行的间距 | NA |

# 

# 总体设计

## 系统描述

驾考试题管理系统旨在利用数据库来管理试题，试题分为判断题，单选题。每一种试题都可以增加一个试题，删除一个试题，修改一个试题，查询一个试题。删除、修改、查询是基于试题编号来查询的。并且用JAVAFX做了相应的页面设计，通过页面来进行所有的操作。

## 总体结构

### 系统结构设计

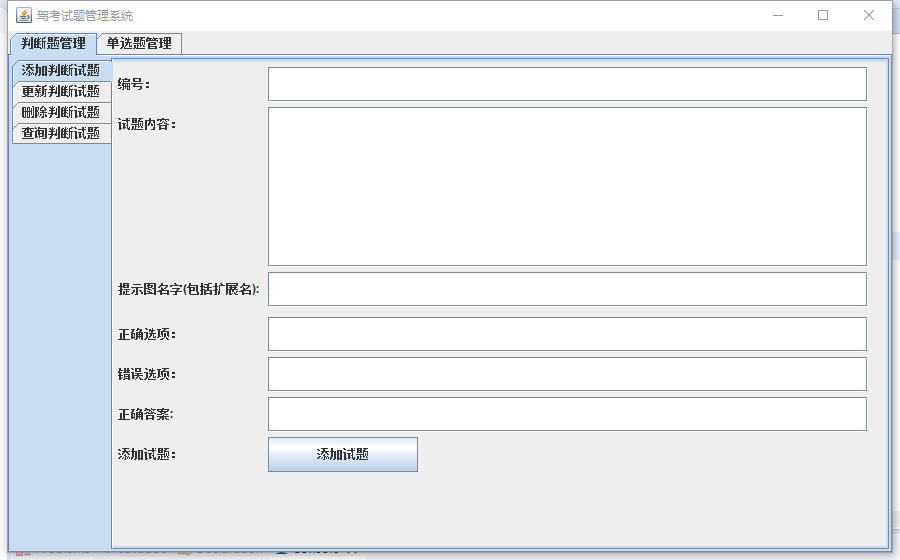


图1 模块框架图

设计一个类似上图的界面，并实现相应的功能。

下面是我设计的界面：



点击相应文字就可以进入相应的界面。

### 数据库结构设计

本系统设计了2个表，分别是：

#### 判断题表：

记录判断题。

|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | 数据类型 | 是键 | 注释 |
| 编号 | char(10) | true | 数字、字母下划线组成。 |
| 试题内容 | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 提示图名字 | varchar(20) | false | 可由数字、字母组成，理论上不低于6位 |
| 正确选项 | varchar(20) | false | 可由数字、字母组成，理论上不低于6位 |
| 错误选项 | varchar(20) | false | 可由数字、字母组成，理论上不低于6位 |
| 正确答案 | varchar(20) | false | 可由数字、字母组成，理论上不低于2位 |

#### 单选题表：

记录选择题。

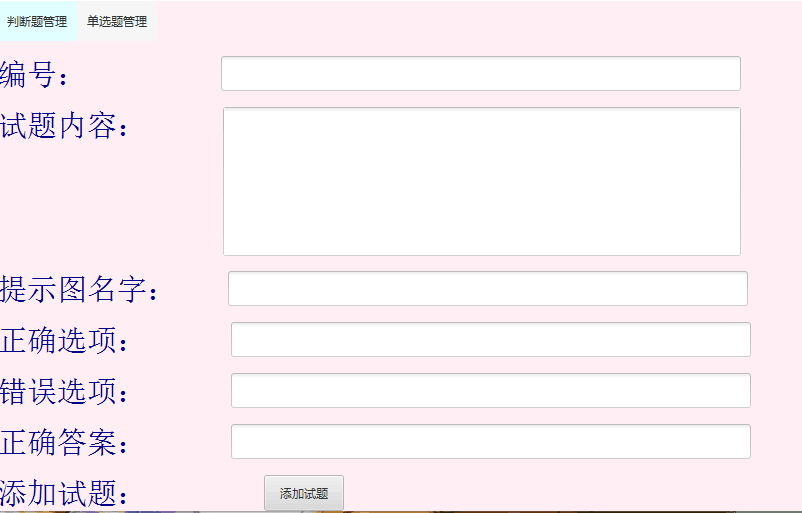
|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | 数据类型 | 是键 | 注释 |
| 编号 | char(10) | true | 数字、字母下划线组成。 |
| 试题内容 | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 提示图名字 | varchar(20) | false | 可由数字、字母组成，理论上不低于6位 |
| 选项A | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 选项B | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 选项C | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 选项D | varchar(200) | false | 可由数字、字母组成，理论上不低于6位 |
| 正确答案 | varchar(20) | false | 可由数字、字母组成，理论上不低于1位 |

# 详细设计

## 功能描述

### 增加试题

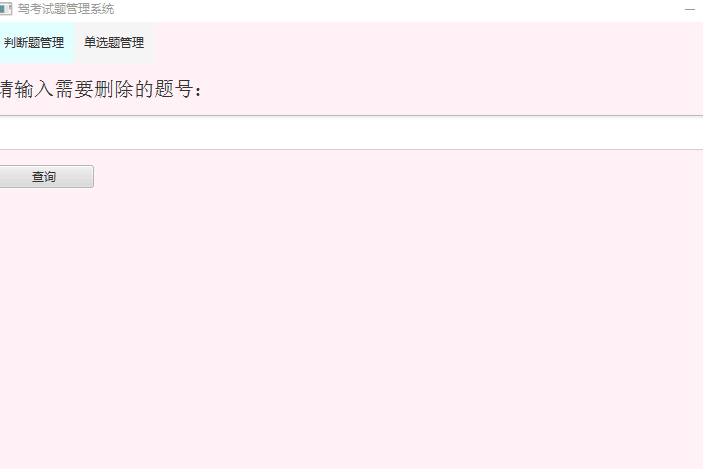
设计一个增加类，连接本地数据库，在界面上增加试题。需要注意的是，增加试题的编号不能与数据库中已有的重复，如果重复会报错。插入的任何一个数据不能为空，如果为空，会提示插入失败。

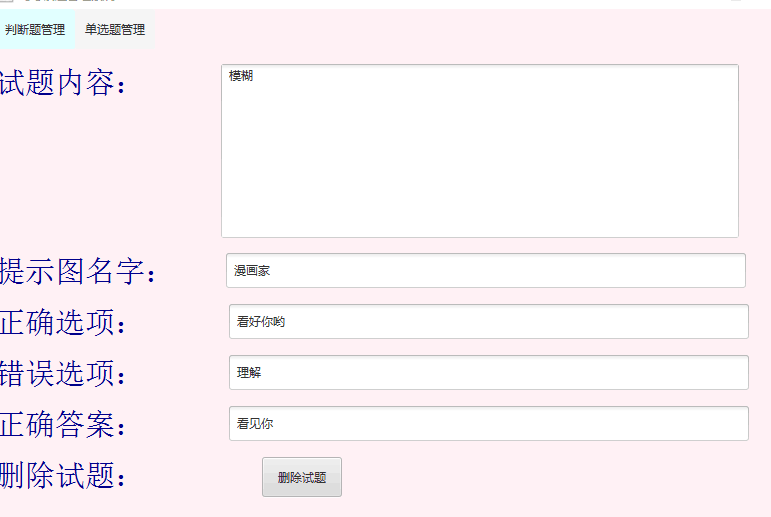


输入成功之后点击添加试题就可添加成功。

### 删除试题

设计一个删除类，连接本地数据库，在界面上删除试题。在删除之前可以输入试题编号，先查询确定在删除。





然后点击删除试题按钮即可完成删除。

### 查询试题

设计一个查询类，连接本地数据库，输入需要查询的试题编号，在界面上查询试题。如果不存在，则会给出提示，否则会显示出查询的结果。

### 修改试题

设计一个更新类，连接本地数据库，在界面上更新试题。输入试题编号进行更新。

具体界面跟删除差不多，点击下面更新试题按钮即可更新。

## 模块设计

### 登入模块

#### 功能描述

用户利用界面进行相应的操作，实现的功能包括对选择题、判断题的删除试题、添加试题、修改试题、查询试题。

#### 查询试题功能实现原理：

新建一个查询类，只要给其数据库的名字和相应的sql语句就可对数据库中的表进行相应的查询，将结果放到一个结果集中，就可以在相应的界面中显示出来。

#### 插入试题功能实现原理：

新建一个插入类，只要给其数据库的名字和相应的sql语句就可对数据库中的表进行相应的插入。需要注意的是插入的主码是唯一的，如果编号与现在数据库中的编号重复就需要判断一下，可以通过捕获一个异常来进行判断，并告诉用户插入是否成功。

#### 更新试题功能实现原理：

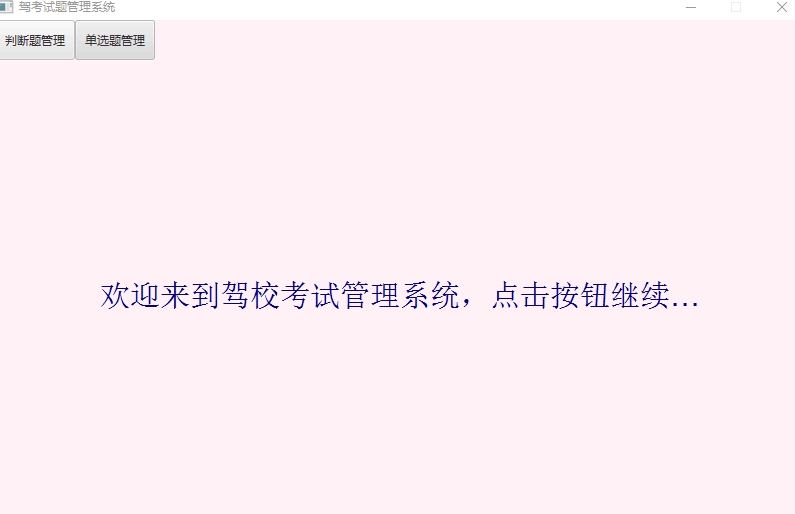
新建一个更新类，只要给其数据库的名字和相应的sql语句就可对数据库中的表进行相应的更新，更新用到了查询类，具体实现在界面上。需要特别注意sql语句的书写，因为要更改很多，所以需要特别注意，不能出错。

#### 删除试题功能实现原理：

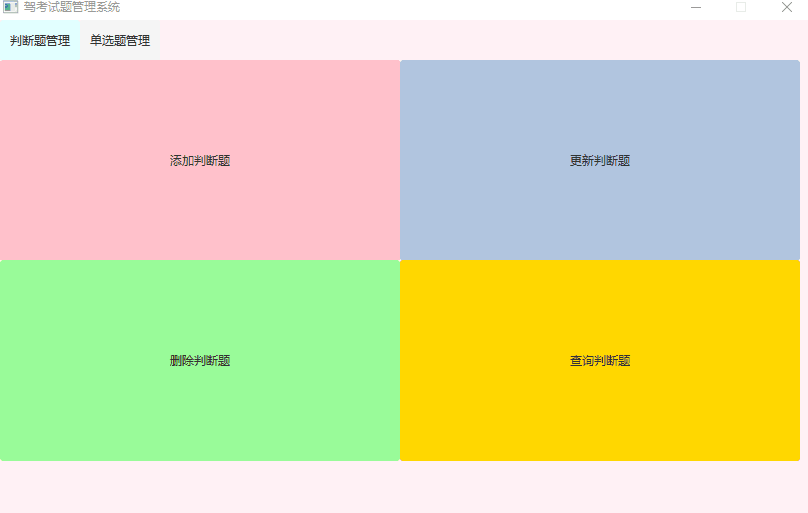
新建一个删除类，只要给其数据库的名字和相应的sql语句就可对数据库中的表进行相应的删除，更新用到了查询类，具体实现在界面上。

#### 界面设计

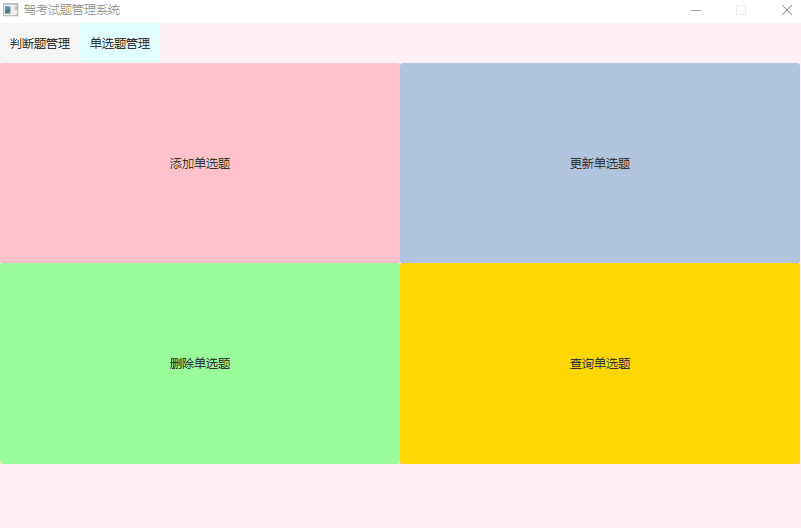
**总界面：**



**判断题界面:**



**单选题界面：**



**查询界面：**



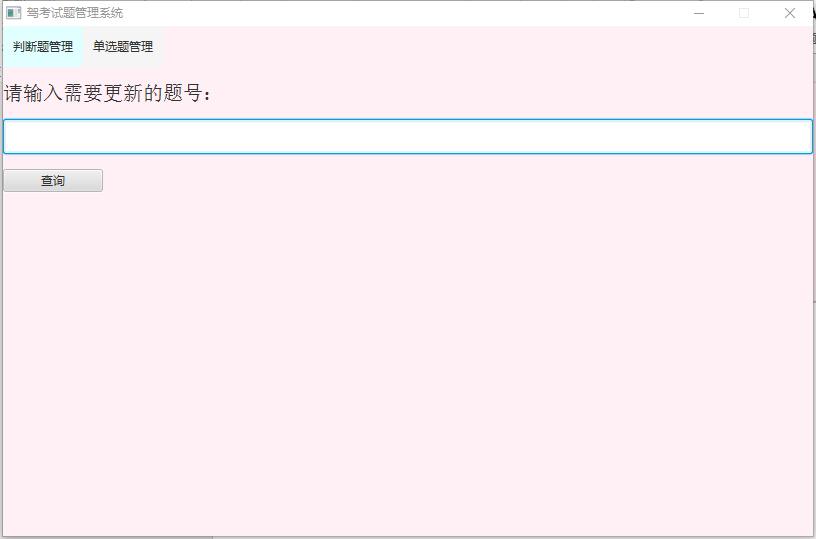
**插入界面：**



**删除界面：**



**更新界面：**



#### 程序设计

首先需要建表，该系统设计了6个类：插入类Insert、删除类Delete、更改类Update、查询类Query、界面类JavaFX、主类zcy。具体的实现在界面类里面。

# 总结

这是第一次接触数据库编程，用到derby数据库。虽然只是短短的一周，但真的学到了不少。对于这个驾考试题管理系统，应该实现的都实现了，界面设计重在简洁，利用JavaFX设计的界面，虽比不上Web界面，但也算是相对美观了。对于这个程序，我写完了才发现具体的实现都在JavaFX这个类（自己定义的，用来实现界面）中，导致该类的代码很长，在写代码之前没有做好规划，有一些可重用的代码并没有重用起来，导致代码中许多重复的部分，这样即不美观又不实用，希望下次写代码之前做好规划，分一下任务，让每一个类实现其功能，使代码简洁明了，方便观看。