**附件2：课程设计（项目实践）报告书模板**



数学与计算机学院

**课 程 设 计 报 告 书**

**所属课程名称 《JAVA程序设计》课程设计**

**题 目 “电子词典软件”设计与开发**

**专业班级 418软件工程**

**学生姓名 王文涛 XXX XXX**

**指导教师 余宏**

**2019年 06月 15日**

**豫章师范学院课程设计（论文）任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学生姓名 |  | |  | 王文涛 |  | |  |
| 设计题目 | “电子词典软件”设计与开发 | | | | | | |
| 设  计  技  术  参  数 | 1、java语言  2、Derby数据库  3、MVC设计模式 | | | | | | |
| 设  计  要  求 | 1、课程设计要求：  要求每个小组利用Java基础和数据库（Derby）的综合应用，单独完成一个基于GUI的电子词典数据管理系统。每个小组成员必须至少单独完成项目的各个功能模块。  2、设计项目：  ①查询  具有：全查询、根据单词查询、随机查询若干个单词的功能模块。  ②添加  具有添加单词信息的功能。  ③修改  具有按英文单词修改其中文解释的功能。  ④删除  具有按英文单词删除记录的功能 | | | | | | |
| 工  作  量 | 写明在本次课程设计期间、本小组分工中主要完成什么工作 | | | | | | |
| 工  作  计  划 | 1. 第一天 数据库设计与创建 2. 第二天 项目分析、设计相关的类 3. 第三天 实现项目连接数据库、实现增加操作 4. 第四天 实现项目修改和删除操作   5、 第五天 实现项目查询功能 | | | | | | |
| 参  考  资  料 | 【序号】作者.书名.出版社.出版年月 （按此格式写） | | | | | | |
| 指导教师签字 | |  | | 教研室主任签字 | |  | |

年 月 日 （此日期一般填写课程设计开始的第一天）

|  |
| --- |
| **软件的详细设计：**  **一、软件功能**  设计GUI界面的电子词典软件。具体要求如下：  ①使用内置Derby数据库。在数据库中使用表存储单词和该单词的翻译解释，例如，“sun, 太阳” “moon, 月亮”等。  ②通过GUI界面管理电子词典。可以向单词库添加单词，可以修改单词库中的单词，可以删除单词库中的单词。  ③通过GUI界面查询单词。可以查询一个，随机查询若干个或全部单词。  程序运行效果如图所示：     1. **数据模型设计**   **2.1 根据系统设计要求在数据模型部分编写了以下类。**  ●CreateDatabaseAndTable类：负责创建数据库和表。  ●Word类：负责封装单词。  ● ConnectDatabase类：负责连接数据库。  ● AddWord类：负责向表中添加单词。  ● UpdateWord类：负责修改表中的单词。  ● DelWord类：负责删除表中的单词。  ● QueryOneWord类：负责查询表中的一个单词。  ● QueryAllWord类：负责查询表中的全部单词。  ● RandomQueryWord类：负责随机查询表中的单词。  **2.2 数据模型部分涉及的主要类的UML图：**    **2.3数据库相关的类**  根据设计要求建立名字为MyEnglishBook的数据库，在该库中建立名字为word\_table的表，其结构如下：  (word varchar(50) primary key, meaning varchar(200))  1)封装单词数据的Word类  在数据库设计中需要用类来封装表的基本结构，这对于后续代码的设计是非常有利的。  下列Word类封装word\_table表的结构。  **Word.java**  **package data;**  **public class Word{**  **String englishWord;**  **String meaning;**  **public void setEnglishWord(String englishWord){**  **this.englishWord = englishWord;**  **}**  **public String getEnglishWord() {**  **return englishWord;**  **}**  **public void setMeaning(String meaning){**  **this.meaning = meaning;**  **}**  **public String getMeaning() {**  **return meaning;**  **}**  **}**  **2)创建数据库表**  通过创建类CreateDatabaseAndTable的对象来创建MyEnglishBook数据库，并在数据库中创建表word\_table用于存储单词库中的单词信息。如果数据库MyEnglishBook不存在，则创建该DB,并建立连接。如果数据库已经存在，那么不再创建，而是直接与其连接。  运行环境会在当前应用程序所在目录下建立名字是MyEnglishBook的文件夹作为Derby数据库，该子目录下存放着和该数据库相关的配置文件。  **CreateDatabaseAndTable.java**    **3)连接数据库的类**  由于后续很多类的实例都需要连接数据库，因此将连接数据库的有关代码封装到ConnectDatabase类中，其他需要连接数据库并进行相关操作的类只要扩展该类就可以使用连接数据库的代码。    **2.4 添加、更新和删除单词的类**  1）添加单词的类  AddWord类的实例使用int insertWord(Word w)方法向word\_table表添加单词信息。  注：本类继承自ConnectDatabase类,目的是可以直接调用该类的方法connectDatabase()连接DB，以及可以直接引用该类的成员变量：Connection类型的变量con.  **AddWord.java**  **2)更新单词的类**  UpdateWord类的实例使用int updateWord(Word w)方法更新word\_table表中的单词信息。  UpdateWord.java （详见后面）  **3)删除单词的类**  DelWord类的实例使用int delWord(Word w)方法删除word\_table表中的单词。  DelWord.java  **2.5 查询单词的类**  1）查询一个单词的类  QueryOneWord类的实例使用Word queryOneWord(Word w)方法查询word\_table表中的一个单词信息。  注意：该方法的形式参数和返回值均为Word类的对象，但参数w对象只需要使用其englishWord字段的信息，即，根据w对象的englishWord字段，返回该对象的完整信息，包括meaning字段的信息。  2）查询全部单词的类  QueryAllWord类的实例使用Word[] queryAllWord()方法查询word\_table表中全部单词。  注意： 因为word\_table表中的单词有多个，所以查询方法的返回值是一个Word类型的数组。  rs.last();  rs.getRow()//获取记录结果集的记录行数  rs.beforeFirst()  3）随机查询单词的类  RandomQueryWord类的实例使用Word[ ] randomQueryWord()方法随机查询word\_table表中的单词。  Math.min(int a, int b) //取a与b中的较小者；  在1—max之间取 k个不同的随机数：  int [] getRandomNumber(int max,int k) //详见后页  **三、简单测试**  按照源文件中的包语句将相关的Java源文件保存到以下目录中：  D:\2000\data  编译各个源文件，例如：  D:\2000>javac data/CreateDatabaseAndTable.java  也可以编译全部源文件：  D:\2000>javac data/\*.java  用前面数据模型框架中的类构建一个简单的应用程序，测试电子词典功能，即在命令行表述对象的行为过程，如果表述成功，那么就为后面的GUI程序设计提供了很好的对象功能测试，在后续的GUI设计中，重要的工作仅仅是为某些对象提供视图界面，并处理相应的界面事件而已。  将AppTest.java源文件按照包名保存到以下目录中：  D:\2000\test  编译源文件：  D:\2000>javac test/ AppTest.java  运行AppTest类（运行效果如图）：    **四、视图设计**  **4.1 视图部分总体设计**  设计GUI程序除了使用数据模型框架内的类以外，需要使用javax.swing包提供的视图（也称Java Swing框架）以及处理视图上触发的界面事件。与AppTest.java中的简单测试相比，GUI程序可以提供更好的用户界面，完成软件的设计要求。  GUI部分设计的类如下：  AddWordView类：其实例为添加单词提供视图。  UpdateWordView类：其实例为更新单词提供视图。  DelWordView类：其实例为删除单词提供视图。  QueryOneWordView类：其实例为查询一个单词提供视图。  QueryAllWordView类：其实例为查询全部单词提供视图。  RandomQueryView类：其实例为随机查询单词提供视图。  IntegrationView类：其实例将其它视图集成为一个视图。  HandleAddWord类：其实例处理AddWordView视图上的界面事件。  HandleUpdateWord类：其实例处理UpdateWordView视图上的界面事件。  HandleDelWord类：其实例处理DelWordView视图上的界面事件。  HandleQueryOneWord类：其实例处理QueryOneWordView视图上的界面事件。  HandleQueryAllWord类：其实例处理QueryAllWordView视图上的界面事件。  HandleRandomQuery类：其实例处理RandomQueryView视图上的界面事件。  4.2 主要类的UML图    **4.3视图相关类**  **1）添加、更新和删除视图**  （1） AddWordView  AddWordView类是Jpanel类的子类，其实例提供了添加单词的视图，用户可以在视图提供的文本框中输入要添加的单词，然后单击提交按钮，如图所示红线框内部分。（参看课本的BOX容器的例题）    AddWordView.java（详见后面）  （2）UpdateWordView  UpdateWordView类是Jpanel类的子类，其实例提供了更新单词的视图，用户可以在视图提供的文本框中输入要更新的单词，然后单击提交按钮。    （3）DelWordView类  DelWordView类是Jpanel类的子类，其实例提供了删除单词的视图，用户可以在视图提供的文本框中输入要删除的单词，然后单击提交按钮。    2）查询视图  （1） QueryOneWordView类：是Jpanel类的子类，其实例提供了一个单词的视图，用户可以在视图提供的文本框中输入要查询的单词，然后单击提交按钮查询某个单词。  注：可以采用BorderLayout布局，使用该布局的north和center部分。    （2） QueryAllWordView类：是Jpanel类的子类，其实例提供了查询全部单词的视图，用户可以在视图提供的查询全部单词的按钮来查询全部的单词。    （3）RandomQueryView  该类是Jpanel类的子类，其实例提供了随机查询的视图，用户可以在视图提供的文本框中输入要随机查询的单词个数，然后单击提交按钮。    （4）IntegrationView类：  该类是JFrame类的子类，其实例使用JTabbedPane将各个视图集成到当前IntegrationView窗体中。  注：参看课本关于JTabbedPane的例子；  注：new JTabbedPanne(JTabbedPane.LEFT)  //标签在左侧，默认实在顶部。    4.4 事件监听器  事件监听器负责处理视图上触发的用户见面事件，以便完成相应的任务。  1）处理添加、更新和删除视图上触发的用户界面事件。  （1） HandleAddWord类：实现了ActionListener接口，其实例负责处理用户单击按钮触发的ActionEvent事件。当用户在添加单词界面（AddWordView视图）单击提交按钮后由HandleAddWord的实例负责将单词添加到数据库的表中。  HandleAddWord.java  其它的监视器类（略）  五、GUI程序  按照源文件中的包语句将相关源文件保存到以下目录中：  D:\2000\view  编译各个源文件，例如：  D:\2000>javac view/IntegrationView.java  也可以编译多个源文件：  D:\2000>javac view/\*.java  把前面数据模型和视图设计部分的类看做一个小框架，下面用框架中的类编写GUI应用程序，完成软件的设计要求。    将AppWindow.java源文件按照报名保存在相应目录，并编译和运行：  D:\2000>javac gui/AppWindow.java  D:\2000>java gui.AppWindow   1. **程序发布**   用户可以使用jar.exe命令制作JAR文件来发布软件。  ★清单文件  编写以下清单文件（用记事本保存时需要将保存类型选择为“所有文件（\*.\*）”）:  wb.mf  Manifest-Version: 1.0  Main-Class: gui.AppWindow  Created-By: 1.8  将wb.mf 保存在D:\2000，即保存在包名所代表的目录上一层目录中。  ★用批处理文件发布程序  使用JAR命令创建JAR文件：  D:\2000>jar cfm EB.jar wb.mf data/\*.class view/\*.class gui/\*.class  其中，参数c 表示要生成一个新的JAR文件，f表示要生成的JAR文件的名字，m表示清单文件的名字。如果没有任何错误提示，将在D:\2000>目录下产生一个名字为EB.jar的文件。  编写以下EB.bat，用记事本板寸为类型“所有文件（\*.\*）”。  EB.bat  path . \jre\bin  pause  javaw –jar EB.jar  然后将该批处理文件、EB.jar、数据库文件夹及JRE(即调试程序使用的JDK安装目录下的jre子目录)保存在一个文件夹中，如名字是2000的文件夹中，双击该EB.bat即可运行程序。（前提是安装了JRE环境）  **七、附：程序详细设计**  （各个类的详细代码）  **AddWord.java**  package data;  import java.sql.\*;  public class AddWord extends ConnectDatabase{  int isOK ;  public int insertWord(Word word) {  connectDatabase();  try {  String SQL ="insert into word\_table values(?,?)";  PreparedStatement sta = con.prepareStatement(SQL);  //从左向右数第1个通配符?的值是word.getEnglishWord():  sta.setString(1,word.getEnglishWord());  //从左向右数第2个通配符?的值是word.getMeaning():  sta.setString(2,word.getMeaning());  isOK = sta.executeUpdate();  con.close();  }  catch(SQLException e) {  isOK = 0; //word\_table表中word字段是主键，即不允许单词重复  }  return isOK;  }  } |

|  |
| --- |
| **课程设计总结** |

|  |
| --- |
| 课程设计是我们的专业课，我也学的格外认真，可是在设计过程中依然出现了很多的问题，但通过看书，上网查询等方式，不断的学习、解决、提高，设计出的网页不论是外观还是内容，都在不断的进步、改善。可以说在自己动手，不断实践的过程中，网页设计水平得到了很大的提高。  关于本次课程设计，我们的设计虽然还不够完善，许多功能因为时间和本身的知识水平有限无法实现，不过，至少在平时的学生基础之上已经有了很大的提高，我坚信它必将会对我们的未来走上工作岗位产生积极的影响。  虽然我们的课程设计在规定的时间内按时完成了任务，但是，由于知识和经验的欠缺，整个系统的开发功能还不算太完善，在设计的过程中也遇到了很多困难，比如 由于早期的需求分析不充分，给后来系统的具体实现带来了很大的困难，所以中间做了一部分的无用功，但通过我们的不懈努力和老师、同学的帮助，设计才得以圆 满按时的完成，所以仅凭个人的力量想设计出一个完美的系统几乎是不可能的。  通过这次课程设计使我们都更加懂得并亲身体会到了理论与实际相结合的重要性，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从实践中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在设计的过程中遇到很多问题，可以说是困难重重，并且在设计的过程中发现了自己的很多不足之处，发现自己对之前所学过的知识理解得不够深刻，掌握得不够牢固，有待加强。 |

**课程设计小组任务分工情况**

|  |
| --- |
|  |

**豫章师范学院课程设计评审意见表**

|  |
| --- |
| 指导教师评语：  成绩：  指导教师：（若有企业老师，则要两个老师都签名）  年 月 日 |
| 课题小组评语：  成绩：  评阅人：（一般是组长或其他同学）  年 月 日 |
| 课程设计总成绩：（依据总成绩=平时成绩30%+作品40%+课程设计报告30%） |
| 课题小组成员成绩：  年 月 日 |