附件1：

《数据结构与算法课程设计》

项目实施进度管理表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | | |  | | | |
| 项目负责人 | | |  | 班级及学号 | |  |
| 同组姓名 | | |  | 班级及学号 | |  |
|  | 班级及学号 | |  |
| **一、项目实施与合作交流内容** | | | | | | |
| 1. **项目实施任务时间安排** | | | | | | |
| 具体实施情况记录 | | | | | | |
| 阶段 | 实施时间 | 实施内容 | | | 完成情况及呈现形式 | |
| 1 |  |  | | |  | |
| 2 |  |  | | |  | |
| 3 |  |  | | |  | |
| 4 |  |  | | |  | |
| 5 |  |  | | |  | |
| 6 |  |  | | |  | |

项目负责人（签字）：

同组（签字）：

年 月 日

附件2：

**《数据结构与算法课程设计》总结报告写作要求及示例**

**一、报告要求：**

1. 各位同学要严格按照要求写报告，A4 纸打印，**用学校统一印发的课程设计报告封面封装**（报告封皮各班班长到学院办公室领取）。
2. 报告中不要求附全部的程序代码，但主要代码在详细设计中说明。
3. 报告上的课程设计名称和指导教师按照数据结构与算法课程设计计划书中指定的填写。
4. 报告字体及字号说明：中文使用宋体，西文使用Times New Roman。一级标题使用四号加黑，其它次级标题使用小四加黑，正文使用五号不加黑（注：不要附代码）。
5. 刻盘说明：统一刻录光盘保存文档和源码，每班一张（班长或学委负责），卷标：数据结构与算法课程设计（\*\*2018-\*班），如：数据结构与算法课程设计（计科2018-1班）。根目录中每个同学一个文件夹，文件夹中保存源代码及报告电子文档，目录名为学号加姓名如：2018081539-刘德华。
6. 报告提交：课程设计报告由各班长或学习委员收齐后，交给各班的课程设计指导教师。**报告提交时间：12月21日（周一）前**。具体提交地点和方式，由各班指导教师确定。

**二、报告格式**

课程设计题目：**小型英汉电子词典**

1．问题描述

题目内容、基本要求、提高要求。

2．需求分析

软件的基本功能、输入/输出形式、测试数据要求。

3．概要设计

抽象数据类型、主程序流程及模块调用关系。

4．详细设计

实现概要设计的数据类型、主程序以及其它模块的算法描述。

5．编码与调试分析

编码与调试过程中遇到的问题及解决的办法，还存在哪些没有解决的问题？

6．使用说明

简要说明程序运行操作步骤。

7．测试结果

准备典型的测试数据和测试方案，包括正确的输入及输出结果和含有错误的输入及输出结果。

8．团队协作和自学内容

自主学习了哪些新知识及主要知识点描述。

**三、示例**

**课程设计题目：**计算表达式的值

**1、问题描述**

设计一个英汉电子词典，支持查找、插入、删除等功能。

基本要求：用AVL树去实现字典的基本操作，查找单词、插入单词（插入时，先查找，找不到则插入，找到则提示用户）、删除单词（删除时，先查找，找到则删除，找不到则提示用户）等。字典是按字母顺序排列的，不能用顺序查找，插入或删除单词后，要保持字典的有序性。

考核要求：

1. 如果采用线性结构且无序，成绩为不及格。
2. 选择合适的数据结构，达到基本要求，成绩酌情为优秀。
3. 鼓励设计图形用户界面

**2．需求分析**

软件的基本功能：由键盘输入英文单词和汉字释意;鼠标点击或回车来实现增删查

输入/输出形式：用户可以通过输入框，根据输入提示。

输入形式：

①英文单词任意

②汉字及符号任意

输出形式：

①word:输入的1

②translation:对应的翻译

测试数据要求：任意，字典残缺，部分单词需要先插入后才可以查询

**3．概要设计**

**（1）抽象数据类型**：

实现了一个 **英文-中文字典**，支持用户通过图形界面进行以下操作：

* 查询单词的翻译。
* 插入新的单词及其翻译。
* 删除已有单词。
* 将字典数据保存到文件中，并从文件加载数据。

通过 **AVL 树** 作为核心数据结构，实现了对字典单词的高效存储、查找、插入和删除操作。根据题目的要求，考虑用AVL树类型比较适合。

**（2）模块调用关系**:

### 本程序中函数包括：main函数，AVL节点类，AVLTree类及其需要的函数（左旋右旋函数），DictionaryAppGUID图形化界面类，loadWordsFromFiles函数(从文件中读取单词及翻译插入AVL树中)等。

函数调用关系如下：

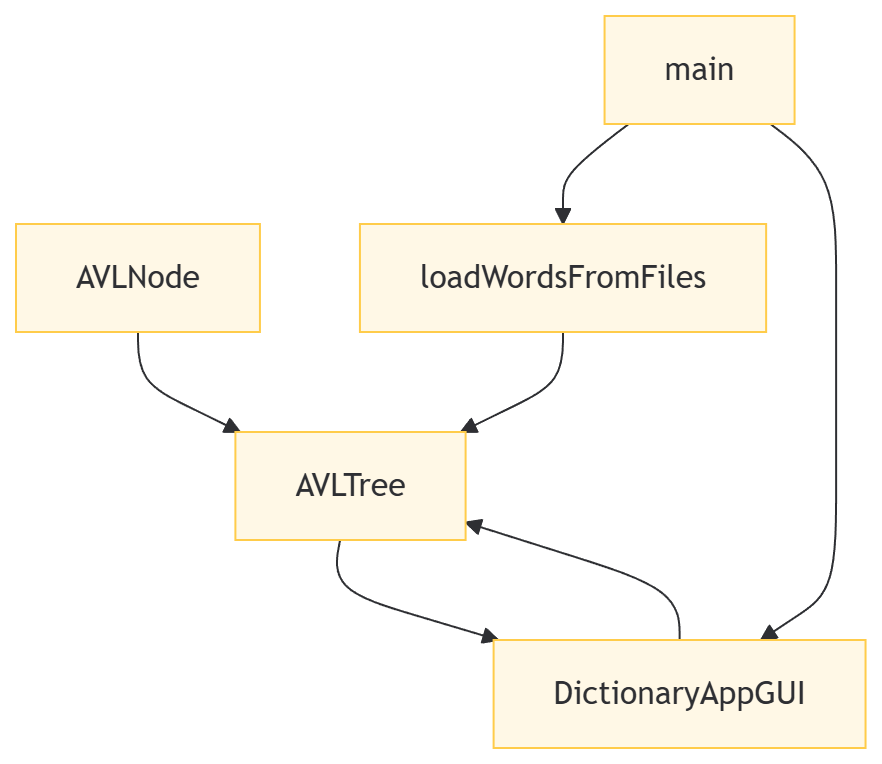


图2：函数调用关系

**4．详细设计**

**（1）实现概要设计的数据类型：**

**采用AVL树**

**static class AVLNode {//AVL节点**

**String word; //单词**

**String translation; //翻译**

**AVLNode left, right; //左右子树**

**int height; //节点的高度**

**}**

**static class AVLTree { //AVL树**

**private AVLNode root; //记录根节点**

**}**

**（2）主程序以及主要模块的算法描述：**

**主函数具体代码**：

public static void main(String[] args) {

AVLTree tree = loadWordsFromFiles("C:/Users/LEGION/Desktop/新建文件夹/02/src/python-code/生成/课设17/word\_source");

DictionaryAppGUI app = new DictionaryAppGUI(tree);

app.show();

}

先是从文件中读取插入树中，然后启动GUI页面

**（3）其它模块的算法描述**

**查询函数**

功能: search 函数通过递归在 AVL 树中查找指定单词，利用 AVL 树的性质（左子树小于根节点，右子树大于根节点）来决定继续向左子树还是右子树递归。它的查找效率是对数级别的，适合于大规模数据的查找操作。

· 用户在输入框中输入单词并点击 **查找** 按钮。

· 程序获取用户输入的单词并去除首尾空格。

· 程序调用 tree.search(tree.getRoot(), word) 查找该单词。

* 如果找到了，显示单词及其翻译。
* 如果没有找到，显示 "Word not found!" 提示框。

**算法的效率与特点**

· **对数时间复杂度：** 查找、插入和删除操作的时间复杂度为 O(log n)，适合大规模字典数据。

· **自平衡：** 确保树的高度始终保持平衡，避免了性能下降。

· **节省空间：** 相比散列表，AVL树在内存使用上更加高效。

· **有序性：** 支持按字母顺序遍历，便于排序和范围查询。

· **避免重复：** 保证字典中的单词不重复。

· **灵活扩展：** 可以支持更多高级操作，如前缀查找、补全等。

**5、编码与调试分析**

**编码与调试过程中遇到的问题及解决办法**：

【问题一】由于使用了类模板，所以在 VC 编译器中不能把类和实现函数分开。

* 解决办法：既然你是在使用 Java，那么你不需要担心模板类分离的问题。Java 不需要显式实例化模板类，也没有类似 VC 编译器的限制。你可以按照常规方式将所有类的实现放在一个文件或多个文件中。

【问题二】程序在判断表达式输入形式有误时，考虑情况不周全。

解决办法：尽可能多的将表达式有误的情况考虑在内。以下为现已考虑到并解决的问题：①表达式中出现非数字或非运算符的其他字符； ②表达式中括号不匹配。

【问题三】单字母变量的表达式计算问题，解决方法是使用字母到值的映射，并在计算过程中解析字母作为变量。

解决此问题的核心代码：

**private double getVariableValue(char variable) throws Exception {**

**if (variableValues.containsKey(variable)) {**

**return variableValues.get(variable);**

**} else {**

**throw new Exception("Variable " + variable + " not defined.");**

**}**

**}**

**// 计算表达式值，假设表达式已预先处理并简化为单字母或数值操作**

**public double evaluateExpression(String expression) throws Exception {**

**// 假设这里直接解析并计算表达式**

**if (Character.isLetter(expression.charAt(0))) {**

**return getVariableValue(expression.charAt(0)); // 替换字母为其对应的值**

**}**

**return Double.parseDouble(expression); // 如果是数字，直接返回**

**}**

**}**

**2.待解决问题**：

**无**

**6、使用说明**

进入菜单，根据提示输入框进行输入：

(a).若要查询的单词,”abraham”出现如测试结果（1）所示翻译；

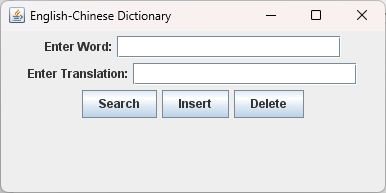
(b).若要退出程序，直接点击右上角X号；

(c).若增加单词及对应翻译，在输入框中输入，然后点击insert即可。

(d).若删除数据，无需添加翻译，直接输入单词点击delete；将会和对应的翻译一同删除

**7、测试结果**

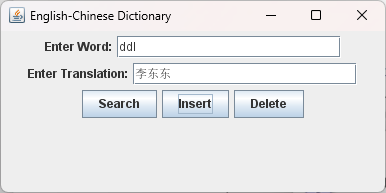
（1）基础界面：

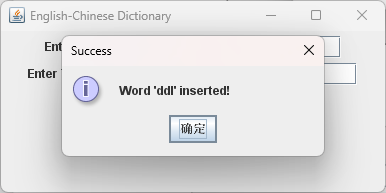


（2）查询显示结果：



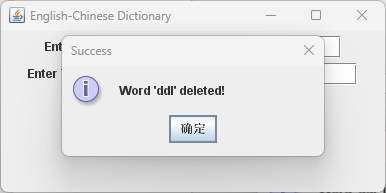
（3）增加数据：





（4）删除数据：

直接输入单词，删除对应的数据



**8、团队协作和自学内容**

在设计和实现字典功能时，我深入学习了AVL树（自平衡二叉搜索树）及其基本操作（插入、删除、搜索等）。AVL树的核心思想是通过平衡因子的计算，确保树的高度差不超过1，从而提高查询效率。在代码实现中，我掌握了如何使用递归实现树的插入、删除、旋转等操作，这对字典的高效查询和维护提供了坚实的支持。

在此过程中，我还学到了如何处理树的平衡操作，如左旋和右旋，确保树在每次插入或删除节点后保持平衡。

由于字典包含大量的单词和翻译数据，我在优化数据存储时，采用了按字母分文件存储的方法，使得字典数据能够被分割成多个小文件，这样可以提高字典查询的效率，并避免单个文件过大导致的性能问题。

参考书：

[1]《数据结构与算法分析：C语言描述》（作者：Mark Allen Weiss）

[2]《Java数据结构和算法》（作者：Robert Lafore）

[3]《Java Swing编程实战》（作者：Richard Reese, Robert Freeman, James Fox, Angela Biesiek）

附件3：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **《数据结构与算法课程设计》考核表** | | | | | |
| 考核项目 | 考核内容 | 支撑的课程目标 | 考核依据与方法 | 占比 | 分数 |
| 平时成绩 | 资料查阅及合理性分析 | 课程目标1 | 1. 资料调研、查阅充分，能分析其合理性（9-10分） 2. 资料调研、查阅比较充分，能分析其合理性（7~8分） 3. 资料调研、查阅一般充分，能做合理性分析但不够具体（5~6） 4. 进行了资料查阅，但没有分析合理性（3~4分） 5. 没有查阅资料（0~2分） | 10% |  |
| 团队协作 | 课程目标4 | 1. 团队协作充分，在组内承担了重要任务（9~10分） 2. 团队协作充分，在组内承担了主要任务（7~8分） 3. 团队协作比较充分，在组内承担了一些任务（5~6分） 4. 有团队协作，但在团队中没有发挥作用（3~4分） 5. 只是组建了团队，没有协作（0~2分） | 10% |  |
| 代码验收 | 数据结构与算法分析分析、方案设计合理性 | 课程目标2 | 1. 设计方案论证充分，方案具体合理可行（16~20分） 2. 设计方案论证充分，方案具体，基本合理（12~15分） 3. 设计方案简单论证，方案基本合理，但不够具体（8~11分） 4. 设计方案无论证，设计方案基本合理但不具体（4~7分） 5. 设计方案不合理或存在原则性错误（0~3分） | 20% |  |
| 程序设计及调试、维护、交互性 | 课程目标3 | 1. 程序设计完整，测试充分，功能齐全，软件交互性很好（16~20分） 2. 程序设计比较完整，测试充分，功能比较齐全，软件交互性较好（12~15分） 3. 程序设计基本完整，测试比较，功能一般，交互性一般（8~11分） 4. 程序设计基本完整，测试不充分，功能不齐全，交互性一般（4~7） 5. 程序设计不完整，功能不齐全，结果有误（0~3分） | 20% |  |
| 答辩 | 问题分析与表述 | 课程目标5 | 1. 答辩过程中能够准确回答老师提出的问题（16~20分） 2. 答辩过程中能够比较准确回答老师提出的问题（12~15分） 3. 答辩过程中能够回答老师提出的问题，但有的问题回答比较模糊（8~11分） 4. 答辩过程中能够少部分准确回答老师提出的问题（4~7分） 5. 答辩过程中不能够准确回答老师提出的问题（0~3分） | 20% |  |
| 报告 | 课程设计报告内容及规范性 | 课程目标6 | 1. 报告数据充分，论述规范，分析恰当合理（16~20分） 2. 报告数据充分，论述比较规范，分析不够详细（12~15分） 3. 报告数据比较充分，缺乏结果分析（8~11分） 4. 报告数据不充分，缺乏结果分析（4~7分） 5. 报告中缺乏数据和结果分析（0~3分） | 20% |  |
| **成绩** | | | | **分数** |  |
| **指导教师（签名）：**  **年 月 日** | | | | | |