



合肥工业大学
HEFEI UNIVERSITY OF TECHNOLOGY

Python 程序设计报告

设计题目：基于Python的机器学习模型训练与科研工具初试

学生姓名：贺鑫帅

专 业：计算机科学与技术

班 级：计科23-3

学 号：2023217373

指导教师：马学森

完成日期：2024.12.20

（一）需求和规格说明

问题描述：这是一款帮助学生背单词的小软件。建立单词库，可从单词库中随机抽取单词，并随机隐去该单词中的一些字母，在屏幕上显示带空格的单词，用户对空格处的字母进行补全，程序判断填补是否正确，并统计正确率。

编程任务：

- (1) 建立单词库，并可以方便地对单词库进行增加、删除。
- (2) 随机读取一个单词。
- (3) 随机隐去单词中的一些字母，规则是：长度为 2~4 空一个字母，5~7 空二个字母，8~10 空三个字母，11 以上空四个字母。用随机数方式确定隐去哪几个位上的字母，并在屏幕上显示带空格单词。
- (4) 用户填充空格处的字母，程序判断填充是否正确。
- (5) 当用户结束游戏时，统计正确率，并输出相应的鼓励语句。

注：由于单词中空格不容易辨认，故将空格改为下划线。

（二）设计

1. 设计思想

本题主要考察链表的操作。我选取的数据存储方式是带头结点和尾指针的单链表，一开始看到题时觉得可以用顺序表存储数据，但是看到要求中要方便地对单词库进行删除，而单链表的删除操作更方便，于是选用单链表。需要用到的单链表操作有增加结点、删除结点、取某一位置的结点和打印所有结点数据。单词库以 txt 文件的形式存储，通过 `fstream` 类的函数进行读写操作。每个单词用数据类型 `string` 存储。通过随机函数抽取单词库中的单词，以及隐去每个单词中的 `n` 位字母。整个程序建立在 `switch` 菜单的大框架下，用户键入相应选项实现相应功能。

2. 设计表示

除了采用如下方式（注意：下面的描述是 C++ 类成员描述，要根据 Python 语言

的相应的修订)，还可以给出流程图，IPO 等各种设计表示。

(1) 结点结构：

结构体名称	成员数据类型	成员名	描述
node	string	data	存储的单词
	node*	next	下一个结点

(2) 类：

类名	成员类别	类型	成员名	描述
list	函数	int	length()	链表长度（结点数）
		string	get_word(int i)	取第 i 个结点的数据
		void	insert()	插入结点（键盘键入数据）
		void	insert_file(string str)	插入结点（从文件读取）
		void	delete_word()	删除结点
		void	print()	打印所有结点数据
		node*	get_head()	取指向头结点的指针
		node*	get_rear()	取尾指针
	数据	int	count	结点数
		node*	head	指向头结点的指针
		node*	rear	尾指针

(3) 函数：

数据类型	函数名称	描述
void	mainmenu()	显示主菜单（一级菜单）
void	loadmenu()	显示读取单词库后的菜单（二级菜单）
void	load()	读取单词库，并且包含 switch 函数，分别调用开始游戏、显示所有单词、添加单词、删除单词、返回主菜单的函数。
void	StartGame(list l)	开始游戏。用到 wordlist 类中 get_word(int i)函数来获取单词，并规定隐去字母规则，然后对字母进行隐去，最后统计结算并输出鼓励语句，细节算法见后。
void	DisplayWord(list l)	显示所有单词。用到 wordlist 类中 print()函数
void	AddWord(list l)	添加单词。用到 wordlist 类中的 insert()函数和 insert_file(string str)函数，其中前者是从键盘键入数据，后者是从文件中读取数据
void	DeleteWord(list l)	删除单词。用到 wordlist 类中的 delete_word()函数。

3. 核心算法

该程序的核心算法主要放在 `StartGame(list l)` 函数中。

开始游戏第一步是随机获取单词，先用随机函数 `srand()` 和 `rand()` 获取一个随机数 `i`，然后再用 `get_word(int i)` 获取链表中第 `i` 个结点的数据存入 `string` 类的 `str` 供修改，同时存入 `string` 类 `temp` 待最后比对使用。

第二步是根据题目设定隐去字母的规则，用 `if` 语句即可。

第三步是重点，随机抽取 `n` 位字母并隐去，同时要保存以便最后用户输入字母后进行比较。我设置了一个长度为 `n` 的整型数组 `a`，其值表示隐去的字母在单词中的位置（下标），即抽取的随机数；数组 `a` 中的下标表示第几次隐去字母（共 `n` 次），即第几次抽取随机数。另设置一个长度为单词长度的整型数组 `tempstr`，初值为 0，将其作为判断抽取的随机数是否重复的标记。然后开始抽取随机数，一共要抽取 `n` 次，每当取一次随机数，将其赋值给 `a` 数组，若 `tempstr[a[j]]` 为 1 时说明该随机数已经重复，则继续抽取知道不重复；若 `tempstr[a[j]]` 为 0，则说明该随机数未出现过，隐去单词上该为字母，并把 `tempstr[a[j]]` 标记为 1 表示该随机数已经出现。而后不重复的 `n` 个随机数已经抽取完成，但并不是按从小到大排序，而用户输入字母时是按从前往后的顺序输入，因此还需要将数组 `a` 排序，我采用了冒泡排序。至此游戏基本完成。

第四步等待用户输入，将用户输入的字母读入并与 `temp[a[j]]` 对比，并统计做题数和正确数。

第五步到游戏结束到选择菜单，选择继续游戏或者结算，若选择继续游戏则再次调用 `StartGame(list l)` 函数，否则就输出题目数，正确数，正确率和鼓励语句。

（三）用户手册

程序运行显示一级菜单，提示选择创建/读取单词库和退出，键入数字 1 或 2。

选择创建/读取单词库进入二级菜单，提示选择开始游戏，显示所有单词，添加单词，删除单词和返回主菜单，键入数字 1-5。

选择开始游戏后，根据题目提示空出字母的个数，依次连续输入字母，并按回车键提交，提交后会有正确或错误的提示，若错误则会显示正确答案。每题做完后都有继续或者结算两个选择，分别键入 1 或 2，若继续则继续答题，结算则会打印答题数，正确数，正确率和鼓励语句，随后按任意键返回主菜单。

选择显示所有单词后，打印单词库中所有单词（带序号），并提示选择返回上一级菜单和返回主菜单，键入数字 1 或 2。

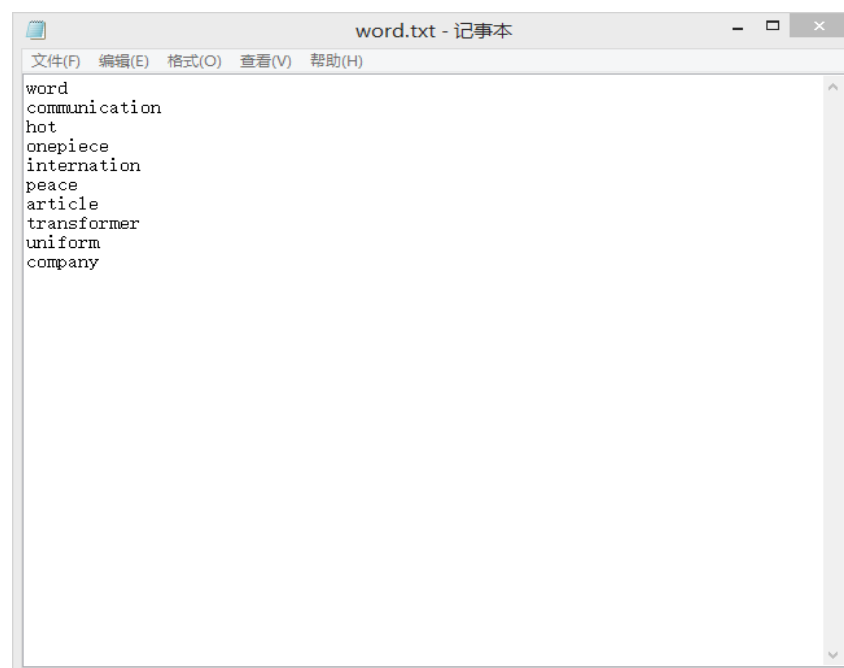
选择添加单词后，提示输入单词，用户直接输入单词并按回车键提交，显示添加成功，添加的单词默认放在单词库中最后一个。之后再提示继续添加，返回上一级和返回主菜单，键入数字 1-3。

选择删除单词后，先打印单词库中所有单词（带序号），提示用户输入要删除的单词的序号，用户键入相应数字后按回车键提交，显示删除成功。之后提示继续删除，保存并返回上一级和保存并返回主菜单，键入数字 1-3。

选择返回上一级即返回主菜单。

（四）调试及测试

1. 测试数据：



单词库

添加单词：debug,system,environment

删除单词：debug,onepiece,peace

2. 测试结果:



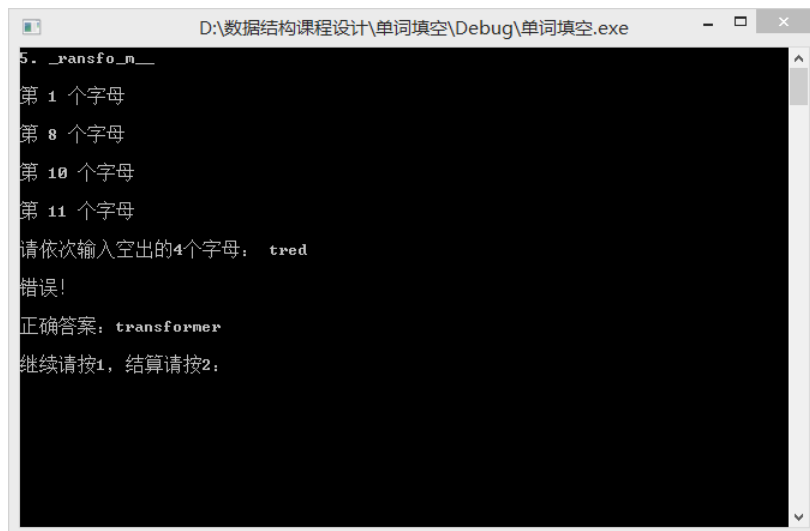
主菜单



二级菜单



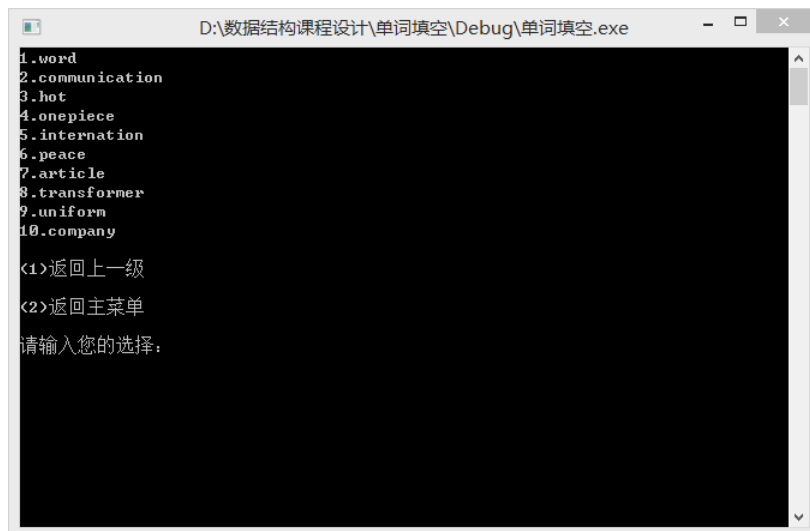
开始游戏 1



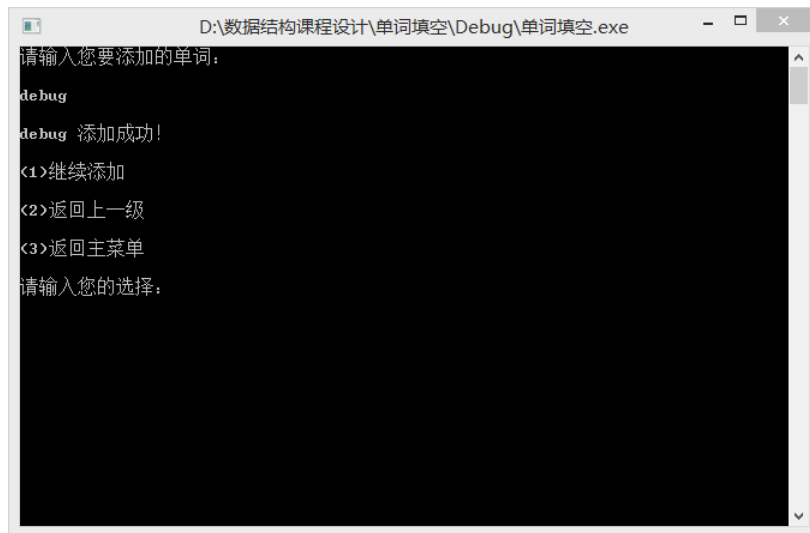
开始游戏 2



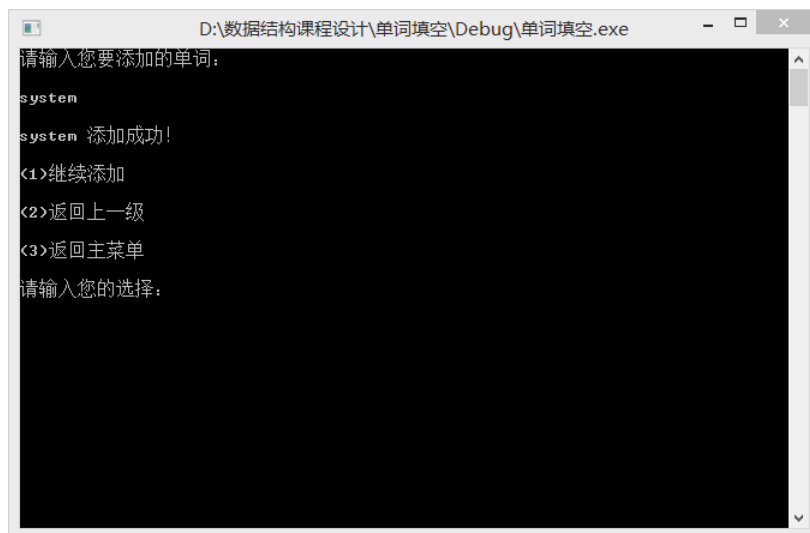
游戏结算



显示所有单词



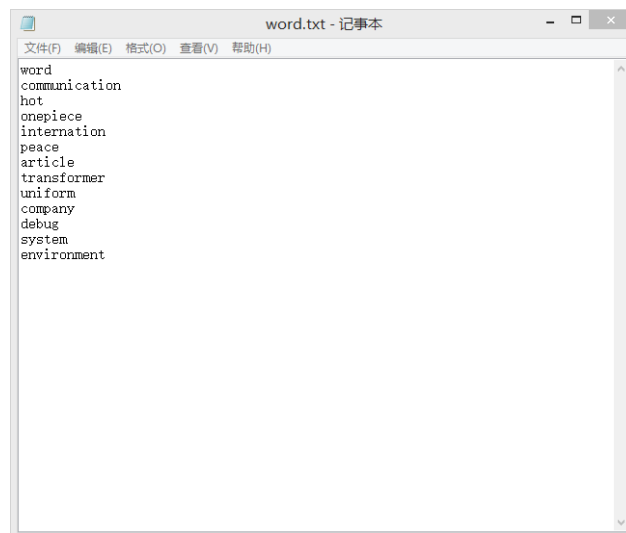
添加单词 1



添加单词 2



添加单词验证 1



添加单词验证 2

```
D:\数据结构课程设计\单词填空\Debug\单词填空.exe
1.word
2.communication
3.hot
4.onepiece
5.international
6.peace
7.article
8.transformer
9.uniform
10.company
11.debug
12.system
13.environment

请输入您要删除的单词的序号：11
删除成功！

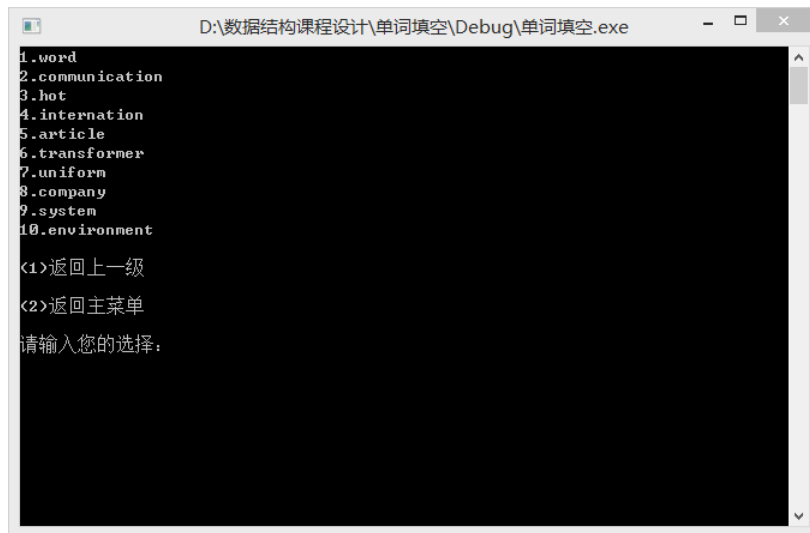
<1>继续删除
<2>保存并返回上一级
<3>保存并返回主菜单
请输入您的选择：
```

删除单词 1

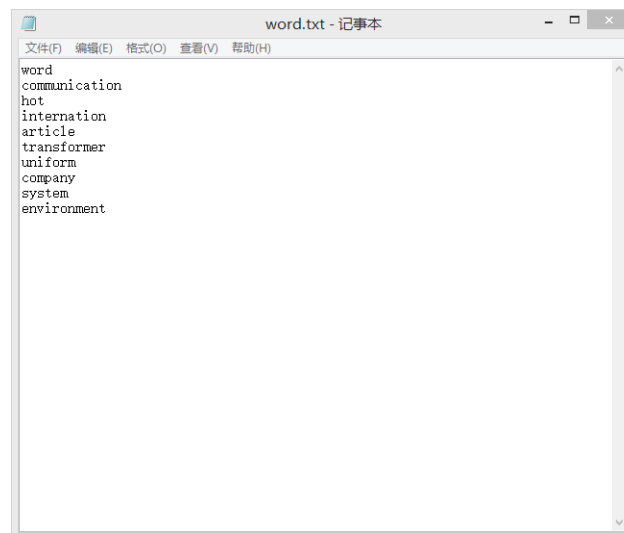
```
D:\数据结构课程设计\单词填空\Debug\单词填空.exe
1.word
2.communication
3.hot
4.onepiece
5.international
6.peace
7.article
8.transformer
9.uniform
10.company
11.system
12.environment

请输入您要删除的单词的序号：
```

删除单词 2



删除单词验证 1



删除单词验证 2

3. 进一步改进:

(1) 原本创建单词库我是另写了一个创建单链表来完成的，但是后来发现用键盘键入的插入结点就可以做到创建一个新的单链表，于是删掉了创建链表的操作。

(2) 目前这个程序只支持一个单词库，名称为 word.txt，但可以改进为让用户输入单词库的名字，创建多个单词库，读取任意单词库，但我没有查到让用户自定义名称读写文件，以及扫描当前路径文件的方法，于是这个想法搁浅了。

(3) 单链表中缺少了析构函数，只要写了析构函数就会出现随机的错误，原因没能够排查出来，于是删掉了析构函数。

(五) 感想

这道题并不难，但是算是首次完成一个综合题。我做这道题时是先把程序的功能拆分开来，比如读写文件，隐去字母，链表操作，一个功能写一个小程序，最后组合起来。分开做时都比较顺利的，但是合起来就出现了很多小问题，比如参数传递出错，变量命名不统一等等。最开始写的时候也没有考虑时间复杂度，只要能实现就好，写完之后才发现有些地方可以优化，比如抽取 n 个不重复随机数的算法，原先我写的就很麻烦，而后经过算法优化才有些改进。我觉得解决问题都会对我以后的程序设计有帮助，所以应当多多实践做题。

另外关于这道题，做完后我才想其实用顺序表代替单链表也可行，因为如果是顺序表，那么随机抽取单词的时候就省时间，按下标直接取出来就行，只是删除单词的时候可能移动次数比较多。相反的，使用单链表的话，每次抽取单词的时候都需要遍历单链表，直到找到序号为该随机数的结点为止，而删除单词时则比较省时间。真正使用这个程序的时候，删除单词的操作很少用到，但是随机抽取单词则是玩一次就要用一次，使用非常频繁，所以我觉得这样来看似乎使用顺序表也非常合理。这也让我明白以后要尽量在写程序之前先考虑好用户会如何使用这个程序，从使用者的角度出发。

附录：

1. 头文件 wordlist.h:

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
struct node
{
    string data;
    node *next;
};

class list
{
public:
    list();
```

```

~list();
int length();
string get_word(int i);           //取第 i 个单词
void insert();                    //插入结点（键盘键入）
void insert_file(string str);     //插入结点（从文件读取）
void delete_word();
void print();
node* get_head();
node* get_rear();
private:
    int count;
    node*head, *rear;
};

list::list()
{
    head = new node;
    head->next = NULL;
    rear = head;
    count = 0;
}

list::~list(){}

int list::length()
{
    return count;
}

```