

深圳大学实验报告

课程名称: Java 程序设计

实验项目名称: 必实验 5 GUI 高级应用

学院: 计算机与软件学院

专业: 计算机科学与技术

指导教师: 潘微科

报告人: 黎浩然 学号: 2018112061 班级: 1

实验时间: 2019 年 11 月 29 日 (周五) - 2019 年 12 月 11 日 (周三)

实验报告提交时间:_____

教务部制

实验目的与要求:

实验目的: 熟练掌握图形界面程序设计。

实验要求:

(1). 编写 Java 应用程序,实现英语单词测试的功能(“单机版中英文单词测试程序”),包含“中译英”和“英译中”两个功能。

中译英:

在主线程中创建一个 Frame 类型的窗口,在该窗口中再创建一个线程 giveWord。

线程 giveWord 每隔 10 秒钟输出一个中文单词和五个英文解释(分别为 A,B,C,D,E),可以有多个解释是正确的(也就是需要支持多选题);用户将答案输入到文本框中(按提交按钮提交结果);程序判断用户输入的答案是否正确(如果错选或漏选均得零分),并实时显示当前题目的正确答案、已经给出的题目的数量、用户答对的数量。

如此循环 15 次,包括随机选择的 10 个单选题(每题 5 分)和随机选择的 5 个多选题(每题 10 分),结束测试时给出最终成绩(满分 100 分)。

词库应至少包含 30 个中文单词和对应的正确的英文解释和错误的英文解释。要求使用图形用户界面。(70 分)

英译中:同上。(10 分)

在报告中附上程序截图、完整的运行结果截图(包括答对、答错的单选题和多选题)和详细的文字说明。(共计 80 分)

报告写作。要求:主要思路有明确的说明,重点代码有详细的注释,行文逻辑清晰可读性强,报告整体写作较为专业。**(20 分,这一项的评分,采用 20 分、10 分和 0 分三个级别)**

说明:

(1) 本次实验课作业满分为 100 分,占总成绩的比例(待定)。

(2) 本次实验课作业截至时间 2019 年 12 月 11 日（周三）23:59。

(3) 报告正文：请在指定位置填写，本次实验**需要单独提交源程序文件**。

(4) 个人信息：**WORD** 文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。

(5) 提交方式：截至时间前，请在 Blackboard 平台中提交。

(6) 发现抄袭（包括复制&粘贴整句话、整张图），该次作业记零分。

(7) 延迟提交，不得分；如有特殊情况，请于截止日期之后 **48 小时内**发邮件到 panweike@szu.edu.cn，并在邮件中注明课程名称、作业名称、姓名、学号等信息，以及特殊情况说明，我收到后会及时回复。

(8) 期末考试阶段补交无效。

(1). 编写 Java 应用程序,实现英语单词测试的功能(“单机版中英文单词测试程序”),包含“中译英”和“英译中”两个功能。

中译英:

在主线程中创建一个 Frame 类型的窗口,在该窗口中再创建一个线程 giveWord。

线程 giveWord 每隔 10 秒钟输出一个中文单词和五个英文解释(分别为 A,B,C,D,E),可以有多个解释是正确的(也就是需要支持多选题);用户将答案输入到文本框中(按提交按钮提交结果);程序判断用户输入的答案是否正确(如果错选或漏选均得零分),并实时显示当前题目的正确答案、已经给出的题目的数量、用户答对的数量。

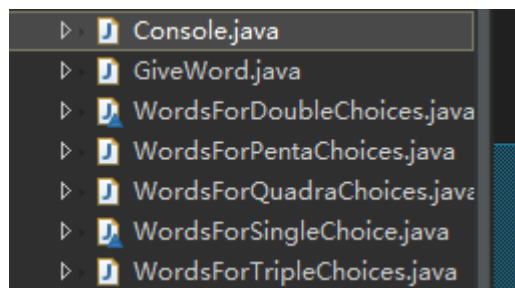
如此循环 15 次,包括随机选择的 10 个单选题(每题 5 分)和随机选择的 5 个多选题(每题 10 分),结束测试时给出最终成绩(满分 100 分)。

词库应至少包含 30 个中文单词和对应的正确的英文解释和错误的英文解释。要求使用图形用户界面。(70 分)

英译中:同上。(10 分)

在报告中附上程序截图、完整的运行结果截图(包括答对、答错的单选题和多选题)和详细的文字说明。(共计 80 分)

程序构思:



程序包括 7 个 public 类;其中 Console 类为图形用户界面(继承了 JFrame)类, GiveWord 类负责从五个类 WordsForSingleChoice, WordsForDoubleChoices, WordsForTripleChoices, WordsForQuadraChoices, WordsForPentaChoices 中提供单词,这五个类分别负责单选,二选,三选,四选及五选题(这也导致了程序代码非常多)

Console 类的核心代码:

```

private static final long serialVersionUID = 1L;
JLabel question;           //题目
JLabel labelA, labelB, labelC, labelD, labelE; //显示5个待选答案
JLabel results;            //正确答案
JTextField youranswer;     //文本框
JButton assure;            //确认提交答案按钮
private JLabel accuracy;   //正确率
private JLabel A,B,C,D,E;  //5个选择
private JLabel askforanswer; //请求答案
private JLabel score;      //当前分数

private int times;         //已答题目
private int singles;       //已答单选
private int multis;        //已答多选
private int corrects;      //正确题目
private int ansnum;        //正确答案
private String answer;     //正确答案
private String style = "英译中";
private int scores = 0;    //分数
boolean ismulti = false;   //是否为多选
private boolean running;   //是否在运行

```

同时实现了监听器：

```

public void actionPerformed(ActionEvent e) {
    String ans;           //用户正确答案
    JLabel tempans;       //对应label
    if(e.getSource()==assure) {
        String texts = youranswer.getText(); //获取用户答案
        int correct = 0;

        if((ismulti&&texts.length()>=2)||(!ismulti&&texts.length()==1)) { //先判断答案数目
            for (int i = 0; i < texts.length(); i++) {
                if (texts.charAt(i) == 'A') {
                    ans = labelA.getText();
                    tempans = labelA;
                } else if (texts.charAt(i) == 'B') {
                    ans = labelB.getText();
                    tempans = labelB;
                } else if (texts.charAt(i) == 'C') {
                    ans = labelC.getText();
                    tempans = labelC;
                } else if (texts.charAt(i) == 'D') {
                    ans = labelD.getText();
                    tempans = labelD;
                } else if (texts.charAt(i) == 'E') {
                    ans = labelE.getText();
                    tempans = labelE;
                } else {
                    break;
                }
            }
        }
    }
}

```

```

        assert ans != null;
        if (check(ans)) {
            assert tempans != null;
            tempans.setBackground(new Color(0, 255, 25)); //设置正确
            if ( (i+1==getAnsNum())&&(i+1==texts.length()) ) { //答案正确
                int flag = 1;
                for(int j = 0; j < texts.length(); j++) //重复选项不得分
                    for(int k = 0; k < texts.length(); k++)
                        if((k!=j)&&(texts.charAt(j)==texts.charAt(k)))
                            flag = 0;
                if(flag==1)
                    correct = 1;
            }
        } else {
            assert tempans != null;
            tempans.setBackground(new Color(255, 72, 72)); //设置正确答案为红色
            correct = 0; //答案错误直接跳出循环
            break;
        }
    }
}
if(correct==1) {
    corrects++;
    if(ismulti){
        scores+=10;
    }else{
        scores+=5;
    }
}
}
}

```

同时由上图可知，判断正确答案时使用了严格的策略，多选（包括一个选项选多次）、漏选均不得分，并且区分大小写。

GiveWord 类核心代码：

```

public void run() {
    while(console.getTimes()<=15) {
        ArrayList<Integer> indexs = new ArrayList<Integer>();
        if(console.getTimes()<15)
            console.assure.setEnabled(true);
        console.youranswer.setText(null);          //清空输入框
        console.results.setText(null);              //清空上一题答案
        console.setAccuracy();                      //显示正确率
        console.setColor();                        //恢复原来的颜色
        if(console.getTimes()==15){
            console.assure.setEnabled(false);      //结束时将提交按钮关闭
        }
        double random = Math.random();             //决定单选题还是多选题
        if(random>0.4) {
            if(console.getSingleTimes()==10) continue;
            console.addTimes();                    //增加总做题次数
            console.addSingle();                  //增加单选题做题次数
            console.ismulti = false;              //设置当前为单选题
            console.setAnsNum(1);                 //正确答案数是1
            HashMap<String, String> choices = single.getChoices();
            ArrayList<Map.Entry<String, String>> list =
                new ArrayList<Map.Entry<String, String>>(choices.entrySet());
            int index = (int) (Math.random()*100)%5; //随机获取一个为本题答案
            String style = console.getStyle();      //获取模式
            if(style.equals("英译中")) {          //测试模式为英译中
                console.question.setText(list.get(index).getKey());
                console.setAnswer(list.get(index).getValue());
                console.labelA.setText(list.get(0).getValue());
                console.labelB.setText(list.get(1).getValue());
                console.labelC.setText(list.get(2).getValue());
            }
        }
    }
}

```

```

        console.labelD.setText(list.get(3).getValue());
        console.labelE.setText(list.get(4).getValue());
    }
    else if(style.equals("中译英")) {            //测试模式为中译英
        console.question.setText(list.get(index).getValue());
        console.setAnswer(list.get(index).getKey());
        console.labelA.setText(list.get(0).getKey());
        console.labelB.setText(list.get(1).getKey());
        console.labelC.setText(list.get(2).getKey());
        console.labelD.setText(list.get(3).getKey());
        console.labelE.setText(list.get(4).getKey());
    }
    try {
        Thread.sleep(10000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
} else {
    if(console.getMultiTimes()==5) continue;
    int selects = (int) (Math.random()*100)%4; //决定二/三/四/五选
    if(selects==0) {                             //二选
        console.addTimes();                      //增加总做题次数
        console.addMulti();                     //增加多选题做题次数
        console.ismulti = true;                 //设置当前为多选题
        console.setAnsNum(2);                   //二选
        ArrayList<String> englishword1 = new ArrayList<String>(); //这是正确的
    }
}

```

```

ArrayList<String> chineseWord1 = new ArrayList<String>();
ArrayList<String> chineseWord2 = new ArrayList<String>();
for (int i = 0; i < 4; i++) { // 保证随机获得的4个样本不重复z只需要4个
    int index = getRandomForDouble();
    while(indexs.contains(index)) {
        index = getRandomForDouble();
    }
    indexs.add(index);
    englishWord1.add(_double.first_eng.get(index));
    englishWord2.add(_double.second_eng.get(index));
    chineseWord1.add(_double.first_chi.get(index));
    chineseWord2.add(_double.second_chi.get(index));
}
int index = (int) (Math.random()*100)%4; // 随机获取一个为本题答案
String style = console.getStyle();
if(style.equals("英译中")) {
    console.question.setText(englishWord1.get(index));
    console.setAnswer(chineseWord1.get(index)+" "+chineseWord2.get(index));
    ArrayList<String> list = new ArrayList<String>();
    list.add(chineseWord1.get(index));
    list.add(chineseWord2.get(index));
    for(int i=0;i<4;i++){
        if(i!=index){
            list.add(chineseWord1.get(i));
        }
    }
    Collections.shuffle(list); // 打乱选项
    console.labelA.setText(list.get(0));
    console.labelB.setText(list.get(1));
}

```

这个类这个下面还有几百行代码，因为测量是先判断是否单选，如果是多选，再判断是几选，而且每种情况代码相似

剩下 5 个类有相同的策略：

运行结果如下：

开始可以选择汉译英或英译汉



以英译汉为例：

单选正确时：

百词斩山寨版

waver Accuracy: 1/1

A vi.摇摆；犹豫不决 B n.魅力；妩媚

C n.论题，论点；论文 D n.有效，效力；正确

E a.破产的

得分：5 正确答案：vi.摇摆；犹豫不决

输入答案 确定

单选错误时

百词斩山寨版

perception Accuracy: 1/4

A n.集(合) B vt.告诉，揭示，泄露

C n.辨别；识别力；歧视 D n.处理；交易；和解

E n.感觉；概念；理解力

得分：5 正确答案：n.感觉；概念；理解力

输入答案 确定

在多选条件下，几选是随机的：

双选：

百词斩山寨版

adhere

Accuracy: 0/1

A

v.追随

B

v.坚持

C

地球

D

adj.充足

E

v.粘附

得分：0

正确答案：v.粘附,v.追随,v.坚持

输入答案

AE

确定

百词斩山寨版

table Accuracy: 1/2

A **n.书桌** B vt.遵守

C **n.丰富** D **n.桌子**

E vt.废除

得分：5 正确答案：n.书桌,n.桌子

输入答案 **ADC** 确定

百词斩山寨版

earth Accuracy: 5/7

A v.粘附 B **星球**

C **行星** D **地球**

E adj.充足

得分：35 正确答案：地球,星球,行星

输入答案 **BCD** 确定

百词斩山寨版

love

Accuracy: 4/4

A

喜爱

B

爱

C

喜欢

D

恋爱

E

adj.充足

得分：30

正确答案：爱,喜爱,喜欢,恋爱

输入答案

ABCD

确定

百词斩山寨版

love

Accuracy: 1/1

A

喜欢

B

喜好

C

恋爱

D

喜爱

E

爱

得分：10

正确答案：爱,喜欢,喜爱,恋爱,喜好

输入答案

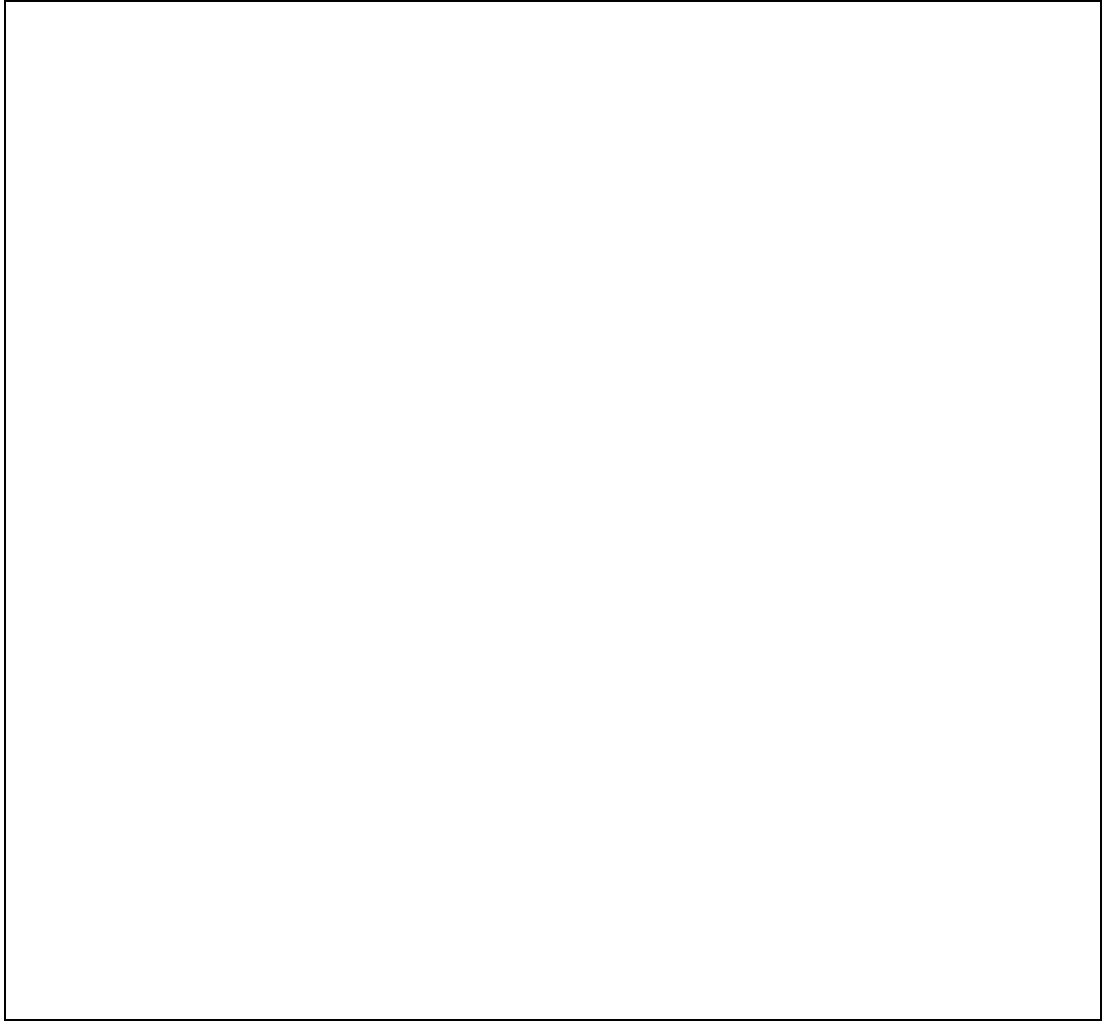
BCDE

确定

+++++

其他（例如感想、建议等等）。

由于未对代码进行优化，代码可能难以阅读



深圳大学学生实验报告用纸

指导教师批阅意见：

成绩评定：

指导教师签字：

2019 年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。