

受理号：\_\_\_\_\_ 受理签字：\_\_\_\_\_

登记号：\_\_\_\_\_ 审查签字：\_\_\_\_\_

流水号	*****

计算机软件著作权登记申请表

软件基本信息	软件全称	Hollow 智能背单词软件				版本号	V1.0	
	软件简称					分类号	30209-8400	
	软件作品说明	<div><input checked="" type="radio"/> 原创 <input type="radio"/> 修改(含翻译软件、合成软件) <input type="checkbox"/> 修改软件须经原权利人授权 <input type="checkbox"/> 原有软件已经登记     • 原登记号：     • 修改（翻译或合成）软件作品说明：</div>						
开发完成日期		2020 年 03 月 27 日						
发表状态		<input type="radio"/> 已发表 <input checked="" type="radio"/> 未发表						
开发方式		<input checked="" type="radio"/> 独立开发 <input type="radio"/> 合作开发 <input type="radio"/> 委托开发 <input type="radio"/> 下达任务开发						
著作权人	姓名或名称	类别	证件类型	证件号码	国籍	省份/城市	园区	
	Hollow Man	自然人	居民身份证	*****	中国	*****		

流水号	*****
-----	-------

权利说明	权利取得方式	<input checked="" type="radio"/> 原始取得 <input type="radio"/> 继受取得（ <input type="radio"/> 受让 <input type="radio"/> 承受 <input type="radio"/> 继承） <input type="checkbox"/> 该软件已登记（原登记号：__） <input type="checkbox"/> 原登记做过变更或补充（变更或补充证明编号：__）		
	权利范围	<input checked="" type="radio"/> 全部 <input type="radio"/> 部分（ <input type="checkbox"/> 发表权 <input type="checkbox"/> 署名权 <input type="checkbox"/> 修改权 <input type="checkbox"/> 复制权 <input type="checkbox"/> 发行权 <input type="checkbox"/> 出租权 <input type="checkbox"/> 信息网络传播权 <input type="checkbox"/> 翻译权 <input type="checkbox"/> 应当由著作权人享有的其他权利）		
软件鉴别材料	<input checked="" type="radio"/> 一般交存	提交源程序前连续的30页和后连续的30页； 提交任何一种文档的前连续的30页和后连续的30页； <input checked="" type="radio"/> 一种文档 <input type="radio"/> ____种文档		
	<input type="radio"/> 例外交存	<input type="radio"/> 使用黑色宽斜线覆盖，页码为： <input type="radio"/> 前10页和任选连续的50页 <input type="radio"/> 目标程序的连续的前、后各30页和源程序任选连续的20页		
软件功能和技术特点	硬件环境	CPU: Intel Core i7-8550U 1.80GHZ, RAM: 8GB, 硬盘: SSD 128GB 机械 1TB		
	软件环境	Windows 10 Version 1909, Java 8		
	编程语言	Java	源程序量	953
	主要功能和技术特点	1. 进入主界面时，首先程序会智能判断程序目录下是否存在 word.txt 词库。 2. 在存在词库的前提下，程序将会自动智能加载 history.dat 来加载你上次的进度，并且进入主界面后自动采用记忆新词模式。 3. 程序主界面会提示你当前模式，当前进度，和已经背过的各类单词数量，还有倒计时剩余的时间。 4. 程序智能使用倒计时进度条，首先你拥有 15 秒的时间记忆新词。 15 秒过去后，程序自动隐藏英文释义，隐藏进度条和倒计时提示标签，同时显示输入框和提交按钮，你需要输入正确的单词。如果输入错误，程序将自动创建 wrong.txt 记录你被错的单词程序将继续让你输入，直到输入正确的答案为止，自动进入下一个单词的记忆。 6. 程序界面显示了已背诵单词和错误及收藏单词。		

流水号	*****
-----	-------

申请办理方式		<input checked="" type="radio"/> 由著作权人申请 <input type="radio"/> 由代理人申请		
申请人信息	姓名或名称	Hollow Man	电话	*****
	详细地址	*****	邮编	*****
	联系人	Hollow Man	手机	*****
	E-mail	*****	传真	
代理人信息	申请人委托下述代理人办理登记事宜，具体委托事项如下：			
	姓名或名称		电话	
	详细地址		邮编	
	联系人		手机	
	E-mail		传真	
申请人认真阅读了填表说明，准确理解了所需填写的内容，保证所填写的内容真实。				
申请人签章：				
2020 年 05 月 25 日				

流水号	*****
-----	-------

证书份数	1份正本	
请确认所需要的计算机软件著作权登记证书副本份数。登记证书正本和副本数量之和不能超过软件著作权人的数量。		
<b>提交申请材料清单</b>		
<b>申请材料类型</b>	<b>申请材料名称</b>	
申请表	打印签字或盖章的登记申请表	一份 <u>4</u> 页
软件鉴别材料	软件源程序	一份 <u>21</u> 页
	软件文档(1)	一份 <u>8</u> 页
	软件文档(2)	一份 ____ 页
身份证明文件	申请人身份证明复印件	一份 <u>1</u> 页
	代理人身份证明复印件	一份 ____ 页
权利归属证明文件	软件转让合同或协议	一份 ____ 页
	承受或继承证明文件	一份 ____ 页
其他材料		一份 ____ 页
		一份 ____ 页
		一份 ____ 页
		一份 ____ 页

**填写说明：**

请按照提示要求提交有关申请材料，并在提交申请材料清单中准确填写实际交存材料页数。若提示中没有的，请填写材料名称及其页数。该页是申请表的组成部分与申请表一并打印提交。

```
01 // Word.java
02
03 import java.io.BufferedInputStream;
04 import java.io.BufferedOutputStream;
05 import java.io.BufferedWriter;
06 import java.io.DataInputStream;
07 import java.io.DataOutputStream;
08 import java.io.File;
09 import java.io.FileInputStream;
10 import java.io.FileOutputStream;
11 import java.io.FileReader;
12 import java.io.IOException;
13 import java.io.InputStream;
14 import java.io.LineNumberReader;
15 import java.io.OutputStreamWriter;
16 import java.util.Scanner;
17
18 class Word
19 {
20     // 第一个记录 word.txt, 第二个记录 wrong.txt, 第三个记录 mark.txt
21     private long[] pstline;
22     private long wdline;
23     private long wrline;
24     private long mkline;
25     private String enword;
26     private String cntrans;
27
28     public Word() throws IOException
29     {
30         File f1 = new File("history.dat");
31
32         if (f1.exists())
33             pstline = readRecord();
34         else
35             pstline = new long[3];
36
37         enword = "";
38         cntrans = "";
39
40         File wd = new File("word.txt");
41         File wr = new File("wrong.txt");
42         File mk = new File("mark.txt");
43
44         if (wd.exists())
45         {
46             LineNumberReader lnr = new LineNumberReader(new FileReader
47             (wd));
48             lnr.skip(Long.MAX_VALUE);
49             wdline = lnr.getLineNumber();
50             lnr.close();
51         }
52     }
53 }
```

```
50     }
51     else
52     {
53         wdline = 0;
54     }
55
56     if (wr.exists())
57     {
58         LineNumberReader lnr = new LineNumberReader(new FileReader
59 (wr));
60         lnr.skip(Long.MAX_VALUE);
61         wrline = lnr.getLineNumber();
62         lnr.close();
63     }
64     else
65         wrline = 0;
66
67     if (mk.exists())
68     {
69         LineNumberReader lnr = new LineNumberReader(new FileReader
70 (mk));
71         lnr.skip(Long.MAX_VALUE);
72         mkline = lnr.getLineNumber();
73         lnr.close();
74     }
75     else
76         mkline = 0;
77
78     }
79
80     public String getEnWord()
81     {
82         return enword;
83     }
84
85     public String getCnTrans()
86     {
87         return cntrans;
88     }
89
90     public boolean setPstLine(long word, long wrong, long mark)
91     {
92         if (word <= wdline && wrong <= wrline && mkline <= mkline)
93         {
94             pstline[0] = word;
95             pstline[1] = wrong;
96             pstline[2] = mark;
97
98             return true;
99         }
100     }
```

```
98         else
99             return false;
100     }
101
102     public boolean setPstLine(long[] record)
103     {
104         if (record.length == 3)
105         {
106             if (record[0] <= wdline && record[1] <= wrline && record[
107 2] <= mkline)
108             {
109                 pstline = record;
110                 return true;
111             }
112             else
113                 return false;
114         }
115         else
116             return false;
117     }
118
119     public long getwordLine()
120     {
121         return pstline[0];
122     }
123
124     public long getwrongLine()
125     {
126         return pstline[1];
127     }
128
129     public long getmarkLine()
130     {
131         return pstline[2];
132     }
133
134     public long getwordTotalLine() throws IOException
135     {
136         File wd = new File("word.txt");
137         if (wd.exists())
138         {
139             LineNumberReader lnr = new LineNumberReader(new FileReade
140 r(wd));
141             lnr.skip(Long.MAX_VALUE);
142             wdline = lnr.getLineNumber();
143             lnr.close();
144         }
145         else
```

```
146         wdline = 0;
147     return wdline;
148 }
149
150 public long getwrongTotalLine() throws IOException
151 {
152     File wr = new File("wrong.txt");
153     if (wr.exists())
154     {
155         LineNumberReader lnr = new LineNumberReader(new FileRead
r(wr));
156
157         lnr.skip(Long.MAX_VALUE);
158         wrline = lnr.getLineNumber();
159         lnr.close();
160     }
161     else
162         wrline = 0;
163     return wrline;
164 }
165
166 public long getmarkTotalLine() throws IOException
167 {
168     File mk = new File("mark.txt");
169     if (mk.exists())
170     {
171         LineNumberReader lnr = new LineNumberReader(new FileRead
r(mk));
172
173         lnr.skip(Long.MAX_VALUE);
174         mkline = lnr.getLineNumber();
175         lnr.close();
176     }
177     else
178         mkline = 0;
179     return mkline;
180 }
181
182 public boolean readWord(String choice)
183 {
184     File file;
185     int choicenum;
186
187     if (choice.equals("word"))
188     {
189         file = new File(choice + ".txt");
190         choicenum = 0;
191     }
192     else if (choice.equals("wrong"))
193     {
```



```
194         file = new File(choice + ".txt");
195         choicenum = 1;
196     }
197     else if (choice.equals("mark"))
198     {
199         file = new File(choice + ".txt");
200         choicenum = 2;
201     }
202     else
203         return false;
204     Scanner sc = null;
205     try
206     {
207         sc = new Scanner(file);
208
209         for (long i = 0; i < pstline[choicenum] * 2; i++)
210             sc.next();
211
212         if (sc.hasNext()) {
213             enword = sc.next();
214             cntrans = sc.next();
215             pstline[choicenum]++;
216         }
217         else {
218             sc.close();
219             return false;
220         }
221         sc.close();
222
223         return true;
224     }
225     catch (Exception e)
226     {
227         sc.close();
228         return false;
229     }
230 }
231
232 public boolean judge(String input, boolean r) throws IOException
233 {
234     if (input.equals(enword))
235     {
236         return true;
237     }
238     else
239     {
240         if (r)
241         {
242             FileOutputStream fos = new FileOutputStream(new File(
243 "wrong.txt"), true);
```

```
243         OutputStreamWriter osw = new OutputStreamWriter(fos,
"GBK");
244         BufferedWriter bw = new BufferedWriter(osw);
245
246         bw.write(enword + " " + cntrans + "\n");
247
248         bw.close();
249         osw.close();
250         fos.close();
251     }
252     wrline++;
253
254     return false;
255 }
256 }
257
258 public void mark() throws IOException
259 {
260     FileOutputStream fos = new FileOutputStream(new File("mark.tx
t"), true);
261     OutputStreamWriter osw = new OutputStreamWriter(fos, "GBK");
262     BufferedWriter bw = new BufferedWriter(osw);
263
264     bw.write(enword + " " + cntrans + "\n");
265     bw.close();
266     osw.close();
267     fos.close();
268
269     mkline++;
270 }
271
272 public void saveRecord() throws IOException
273 {
274     FileOutputStream out1 = new FileOutputStream("history.dat");
275     BufferedOutputStream out2 = new BufferedOutputStream(out1);
276     DataOutputStream out = new DataOutputStream(out2);
277
278     out.writeLong(pstline[0]);
279     out.writeLong(pstline[1]);
280     out.writeLong(pstline[2]);
281
282     out.close();
283 }
284
285 public long[] readRecord() throws IOException
286 {
287     InputStream in1 = new FileInputStream("history.dat");
288     BufferedInputStream in2 = new BufferedInputStream(in1);
289     DataInputStream in = new DataInputStream(in2);
290     long record[] = new long[3];
```

```
291
292     record[0] = in.readLong();
293     record[1] = in.readLong();
294     record[2] = in.readLong();
295
296     in.close();
297
298     return record;
299 }
300 }
301
302 // MainFrame.java
303
304 import java.awt.Component;
305 import java.awt.Dimension;
306 import java.awt.Font;
307 import java.awt.GridBagConstraints;
308 import java.awt.GridBagLayout;
309 import java.awt.HeadlessException;
310 import java.awt.event.ActionEvent;
311 import java.awt.event.ActionListener;
312 import java.awt.event.WindowAdapter;
313 import java.awt.event.WindowEvent;
314 import java.io.File;
315 import java.io.IOException;
316 import java.util.logging.Level;
317 import java.util.logging.Logger;
318
319 import javax.swing.BorderFactory;
320 import javax.swing.Box;
321 import javax.swing.BoxLayout;
322 import javax.swing.DefaultListModel;
323 import javax.swing.JButton;
324 import javax.swing.JFrame;
325 import javax.swing.JLabel;
326 import javax.swing.JList;
327 import javax.swing.JMenu;
328 import javax.swing.JMenuBar;
329 import javax.swing.JMenuItem;
330 import javax.swing.JOptionPane;
331 import javax.swing.JPanel;
332 import javax.swing.JProgressBar;
333 import javax.swing.JScrollPane;
334 import javax.swing.JTextField;
335 import javax.swing.ListSelectionModel;
336
337 public class MainFrame {
338     private Word word;
339     private JFrame f;
340     private JMenuBar mMenuBar;
```

```
341     private JMenu mMenuSelect, mMenuHelp, mMenuReview;
342     private JMenuItem mItemReviewW, mItemReviewM, mItemAbout, mItemHelp, mItemContinue, mItemRefresh, mItemWord;
343     private JPanel topPanel, bottomPanel, middlePanel;
344     private JButton judgeAns, skip, mark;
345     private JLabel en, cn, pgwd, pgwr, pg, pgmk, tm, md;
346     private DefaultListModel<String> listModel;
347     private JList<String> sourceList;
348     private JLabel sourceLabel, nLabel;
349     private JPanel panelContainer;
350     private JTextField tf1;
351     private JProgressBar jProgressBar;
352     private DefaultListModel<String> targetListModel;
353     private int count, countm;
354     private ProgressBarThread progressBarThread;
355
356     class ProgressBarThread extends Thread
357     {
358         private int curTime = 15;
359         boolean go = true;
360
361         public void end()
362         {
363             go = false;
364         }
365
366         public void run()
367         {
368             judgeAns.setEnabled(false);
369             nLabel.setVisible(false);
370             tf1.setEnabled(false);
371             jProgressBar.setVisible(true);
372             tm.setVisible(true);
373
374             while (go)
375             {
376                 if (curTime >= 0)
377                 {
378                     try
379                     {
380                         tm.setText("时间剩余: " + curTime + " 秒");
381                         jProgressBar.setValue(curTime--);
382                         sleep(1000);
383                     }
384                     catch (InterruptedException ex)
385                     {
386                         Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
387                     }
388                 }
389             }
390         }
391     }
```

```
389         else
390         {
391             judgeAns.setEnabled(true);
392             nLabel.setVisible(true);
393             tf1.setEnabled(true);
394             jProgressBar.setValue(0);
395             en.setVisible(false);
396             jProgressBar.setVisible(false);
397             tm.setVisible(false);
398             go = false;
399             jProgressBar.setValue(15);
400         }
401     }
402 }
403 }
404
405 public MainFrame() throws IOException
406 {
407     count = 0;
408     countm = 0;
409     word = new Word();
410     panelContainer = new JPanel();
411     topPanel = new JPanel();
412     listModel = new DefaultListModel<String>();
413
414     sourceLabel = new JLabel("已背诵单词: ");
415     judgeAns = new JButton("提交");
416     skip = new JButton("跳过");
417     mark = new JButton("收藏");
418     cn = new JLabel("中文");
419     en = new JLabel("英文");
420     nLabel = new JLabel("请输入答案: ");
421     md = new JLabel("模式: 记忆新词");
422
423     if (word.getwordTotalLine() == 0)
424         pg = new JLabel("进度 : 0%");
425     else
426         pg = new JLabel("进度 :
427 " + 100.0 * word.getwordLine() / word.getwordTotalLine() + "%");
428     pgwd = new JLabel("已背单词:
429 " + word.getwordLine() + "/" + word.getwordTotalLine());
430     pgwr = new JLabel("已背错词:
431 " + word.getwrongLine() + "/" + word.getwrongTotalLine());
432     pgmk = new JLabel("已背收藏词:
433 " + word.getmarkLine() + "/" + word.getmarkTotalLine());
434     tm = new JLabel("时间剩余: 15 秒");
435     f = new JFrame("Hollow 智能背单词软件 V1.0");
436
437     tf1 = new JTextField();
```

```
435         tf1.setHorizontalAlignment(JTextField.CENTER);
436
437         en.setFont(new Font(null, Font.PLAIN, 32));
438         cn.setFont(new Font(null, Font.PLAIN, 16));
439         tm.setFont(new Font(null, Font.PLAIN, 16));
440         md.setFont(new Font(null, Font.PLAIN, 16));
441
442         jProgressBar = new JProgressBar();
443         jProgressBar.setMaximum(15);
444         jProgressBar.setPreferredSize(new java.awt.Dimension(146, 10)
445 );
446         progressBarThread = new ProgressBarThread();
447         mMenBar = new JMenuBar();
448
449         mMenuSelect = new JMenu("选项");
450         mMenuHelp = new JMenu("帮助");
451         mMenuReview = new JMenu("复习");
452         mItemReviewW = new JMenuItem("背错的单词");
453         mItemReviewM = new JMenuItem("收藏的单词");
454         mItemAbout = new JMenuItem("关于");
455         mItemHelp = new JMenuItem("使用说明");
456         mItemContinue = new JMenuItem("继续上次");
457         mItemRefresh = new JMenuItem("重新开始");
458         mItemWord = new JMenuItem("记忆新词(默认)");
459
460         targetListModel = new DefaultListModel<String>();
461     }
462     public void run()
463     {
464         topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.Y_AXIS))
465 ;
466         topPanel.add(md);
467         topPanel.add(pg);
468         topPanel.add(pgwd);
469         topPanel.add(pgwr);
470         topPanel.add(pgm);
471         topPanel.add(tm);
472
473         middlePanel = new JPanel();
474         middlePanel.setLayout(new BoxLayout(middlePanel, BoxLayout.X_
475 AXIS));
476         sourceLabel.setAlignmentY(Component.TOP_ALIGNMENT);
477         sourceLabel.setBorder(BorderFactory.createEmptyBorder(4, 5, 0
478 , 5));
479         sourceList = new JList<String>(listModel);
480         sourceList.setSelectionMode(ListSelectionModel.MULTIPLE_INTER
481 VAL_SELECTION);
482         sourceList.setVisibleRowCount(5);
483     }
```

```
480         JScrollPane sourceListScroller = new JScrollPane(sourceList);
481         sourceListScroller.setPreferredSize(new Dimension(120, 80));
482         sourceListScroller.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
483         sourceListScroller.setAlignmentY(Component.TOP_ALIGNMENT);
484
485         JPanel sourceListPanel = new JPanel();
486         sourceListPanel.setLayout(new BoxLayout(sourceListPanel, BoxLayout.X_AXIS));
487         sourceListPanel.add(sourceLabel);
488         sourceListPanel.add(sourceListScroller);
489         sourceListPanel.setAlignmentY(Component.TOP_ALIGNMENT);
490         sourceListPanel.setBorder(BorderFactory.createEmptyBorder(0,
491 0, 0, 30));
491         middlePanel.add(sourceListPanel);
492         progressBarThread.start();
493
494         JPanel buttonPanel = new JPanel();
495         buttonPanel.add(Box.createRigidArea(new Dimension(15, 15)));
496         buttonPanel.setAlignmentY(Component.TOP_ALIGNMENT);
497         buttonPanel.setBorder(BorderFactory.createEmptyBorder(15, 5,
498 15, 5));
498         middlePanel.add(buttonPanel);
499
500         JLabel targetLabel = new JLabel("错误及收藏单词: ");
501         targetLabel.setAlignmentY(Component.TOP_ALIGNMENT);
502         targetLabel.setBorder(BorderFactory.createEmptyBorder(4, 5, 0
503 , 5));
503
504         JList<String> targetList = new JList<String>(targetListModel)
505 ;
505         targetList.setSelectionMode(ListSelectionModel.MULTIPLE_INTERVAL_SELECTION);
506         targetList.setVisibleRowCount(5);
507
508         JScrollPane targetListScroller = new JScrollPane(targetList);
509         targetListScroller.setPreferredSize(new Dimension(120, 80));
510         targetListScroller.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
511         targetListScroller.setAlignmentY(Component.TOP_ALIGNMENT);
512
513         JPanel targetListPanel = new JPanel();
514         targetListPanel.setLayout(new BoxLayout(targetListPanel, BoxLayout.X_AXIS));
515         targetListPanel.add(targetLabel);
516         targetListPanel.add(targetListScroller);
517         targetListPanel.setAlignmentY(Component.TOP_ALIGNMENT);
518         targetListPanel.setBorder(BorderFactory.createEmptyBorder(0,
519 30, 0, 0));
519         middlePanel.add(targetListPanel);
```

```
520         bottomPanel = new JPanel();
521         bottomPanel.setLayout(new BoxLayout(bottomPanel, BoxLayout.Y_
AXIS));
522         bottomPanel.add(jProgressBar);
523
524         bottomPanel.add(en);
525         bottomPanel.add(Box.createVerticalStrut(10));
526         bottomPanel.add(cn);
527         bottomPanel.add(Box.createVerticalStrut(10));
528         bottomPanel.add(nLabel);
529         bottomPanel.add(Box.createVerticalStrut(10));
530         bottomPanel.add(tf1);
531         bottomPanel.add(Box.createVerticalStrut(10));
532         buttonPanel = new JPanel();
533         buttonPanel.setLayout(new BoxLayout(buttonPanel, BoxLayout.X_
AXIS));
534         buttonPanel.add(skip);
535         buttonPanel.add(Box.createHorizontalGlue());
536         buttonPanel.add(judgeAns);
537
538         buttonPanel.add(Box.createHorizontalGlue());
539         buttonPanel.add(mark);
540         bottomPanel.add(Box.createVerticalStrut(10));
541         bottomPanel.add(buttonPanel);
542         bottomPanel.add(Box.createVerticalStrut(10));
543         panelContainer.setLayout(new GridBagLayout());
544
545         GridBagConstraints c1 = new GridBagConstraints();
546         c1.gridx = 0;
547         c1.gridy = 0;
548         c1.weightx = 1.0;
549         c1.weighty = 1.0;
550
551         c1.fill = GridBagConstraints.BOTH;
552         panelContainer.add(topPanel, c1);
553
554         GridBagConstraints c2 = new GridBagConstraints();
555         c2.gridx = 0;
556         c2.gridy = 1;
557         c2.weightx = 1.0;
558         c2.weighty = 0;
559         c2.fill = GridBagConstraints.HORIZONTAL;
560         panelContainer.add(bottomPanel, c2);
561
562         GridBagConstraints c3 = new GridBagConstraints();
563         c3.gridx = 0;
564         c3.gridy = 2;
565         c3.weightx = 1.0;
566         c3.weighty = 0;
567         c3.fill = GridBagConstraints.HORIZONTAL;
```



```
568         panelContainer.add(middlePanel, c3);
569         mMenuReview.add(mItemReviewW);
570         mMenuReview.add(mItemReviewM);
571         mMenuSelect.add(mItemWord);
572         mMenuSelect.add(mMenuReview);
573         mMenuSelect.add(mItemContinue);
574         mMenuSelect.add(mItemRefresh);
575         mMenuHelp.add(mItemHelp);
576         mMenuHelp.add(mItemAbout);
577         mMenBar.add(mMenuSelect);
578         mMenBar.add(mMenuHelp);
579
580         // 监听各个菜单项的动作并调用相应函数
581         mItemAbout.addActionListener(new ActionListener()
582         {
583             public void actionPerformed(ActionEvent e)
584             {
585                 JOptionPane.showMessageDialog(null,
586                     "作者: \n                      Hollow Man \n 联系邮箱:
587                     \n *****",
588                     "关于", JOptionPane.INFORMATION_MESSAGE);
589             }
590         });
591
592         mItemHelp.addActionListener(new ActionListener()
593         {
594             public void actionPerformed(ActionEvent e)
595             {
596                 JOptionPane.showMessageDialog(null, "本程序帮助你记忆单
597                 词。 \n 词库存放在软件目录下 word.txt 文件中 \n 你拥有 15 秒钟记忆单词的时间, 然后你
598                 必须拼写出单词!",
599                 "使用说明", JOptionPane.INFORMATION_MESSAGE);
600             }
601         });
602
603         mItemReviewW.addActionListener(new ActionListener()
604         {
605             public void actionPerformed(ActionEvent e)
606             {
607                 if (md.getText().equals("模式: 复习错词"))
608                     JOptionPane.showMessageDialog(null, "当前已在该模式
609                     下!", "提示", JOptionPane.INFORMATION_MESSAGE);
610                 else
611                 {
612                     if (word.readWord("wrong"))
613                     {
614                         countm = 0;
615                     }
616                 }
617             }
618         });
```

```
611         count = 0;
612         en.setText(word.getEnWord());
613         cn.setText(word.getCnTrans());
614         progressBarThread.end();
615         progressBarThread = new ProgressBarThread();
616         progressBarThread.start();
617         md.setText("模式: 复习错词");
618         JOptionPane.showMessageDialog(null, "成功切换
至复习错词模式!", "成功", JOptionPane.INFORMATION_MESSAGE);
619     }
620     else
621     {
622         JOptionPane.showMessageDialog(null, "你已经背
完了所有错词, 重新开始吧!", "没有单词了", JOptionPane.ERROR_MESSAGE);
623     }
624 }
625 }
626 });
627
628 mItemWord.addActionListener(new ActionListener()
629 {
630     public void actionPerformed(ActionEvent e)
631     {
632         if (md.getText().equals("模式: 记忆新词"))
633             JOptionPane.showMessageDialog(null, "当前已在该模式
下!", "提示", JOptionPane.INFORMATION_MESSAGE);
634         else
635         {
636             if (word.readWord("word"))
637             {
638                 countm = 0;
639                 count = 0;
640                 en.setText(word.getEnWord());
641                 cn.setText(word.getCnTrans());
642                 progressBarThread.end();
643                 progressBarThread = new ProgressBarThread();
644                 progressBarThread.start();
645                 md.setText("模式: 记忆新词");
646                 JOptionPane.showMessageDialog(null, "成功切换
至记忆新词模式!", "成功", JOptionPane.INFORMATION_MESSAGE);
647             }
648             else
649             {
650                 JOptionPane.showMessageDialog(null, "你已经背
完了所有新词, 重新开始吧!", "没有单词了", JOptionPane.ERROR_MESSAGE);
651             }
652         }
653     }
654 });
```

```
655

656         mItemReviewM.addActionListener(new ActionListener()
657         {
658             public void actionPerformed(ActionEvent e)
659             {
660                 if (md.getText().equals("模式: 复习收藏词"))
661                     JOptionPane.showMessageDialog(null, "当前已在该模式
662 下!", "提示", JOptionPane.INFORMATION_MESSAGE);
663                 else
664                 {
665                     if (word.readWord("mark"))
666                     {
667                         countm = 0;
668                         count = 0;
669                         en.setText(word.getEnWord());
670                         cn.setText(word.getCnTrans());
671                         progressBarThread.end();
672                         progressBarThread = new ProgressBarThread();
673                         progressBarThread.start();
674                         md.setText("模式: 复习收藏词");
675                         JOptionPane.showMessageDialog(null, "成功切换
676 至复习收藏词模式!", "成功", JOptionPane.INFORMATION_MESSAGE);
677                     }
678                     else
679                     {
680                         JOptionPane.showMessageDialog(null, "你已经背
681 完了所有收藏词, 重新开始吧!", "没有单词了", JOptionPane.ERROR_MESSAGE);
682                     }
683                 }
684             }
685         });

686         mItemRefresh.addActionListener(new ActionListener()
687         {
688             public void actionPerformed(ActionEvent e)
689             {
690                 if (md.getText().equals("模式: 复习收藏词"))
691                 {
692                     if (word.setPstLine(word.getwordLine(), word.getw
693 rongLine(), 0))
694                     {
695                         progressBarThread.end();
696                         try {
697                             newWord("mark");
698                         }
699                         catch (IOException e1)
700                         {
701                             //
702                         }
703                     }
704                 }
705             }
706         });
```

```
698             JOptionPane.showMessageDialog(null, "文件
读取错误!", "错误", JOptionPane.ERROR_MESSAGE);
699         }
700         JOptionPane.showMessageDialog(null, "成功!
", "重新开始", JOptionPane.INFORMATION_MESSAGE);
701     }
702     else
703         JOptionPane.showMessageDialog(null, "失败!
", "重新开始", JOptionPane.ERROR_MESSAGE);
704     }
705     else if (md.getText().equals("模式: 复习错词"))
706     {
707         if (word.setPstLine(word.getwordLine(), 0, word.g
etmarkLine()))
708         {
709             progressBarThread.end();
710             try
711             {
712                 newWord("wrong");
713             }
714             catch (IOException e1)
715             {
716                 JOptionPane.showMessageDialog(null, "文件
读取错误!", "错误", JOptionPane.ERROR_MESSAGE);
717             }
718             JOptionPane.showMessageDialog(null, "成功!
", "重新开始", JOptionPane.INFORMATION_MESSAGE);
719         }
720         else
721             JOptionPane.showMessageDialog(null, "失败!
", "重新开始", JOptionPane.ERROR_MESSAGE);
722     }
723     else if (md.getText().equals("模式: 记忆新词"))
724     {
725         if (word.setPstLine(0, word.getwrongLine(), word.
getmarkLine()))
726         {
727             progressBarThread.end();
728             try
729             {
730                 newWord("word");
731             } catch (IOException e1)
732             {
733                 JOptionPane.showMessageDialog(null, "文件
读取错误!", "错误", JOptionPane.ERROR_MESSAGE);
734             }
735             JOptionPane.showMessageDialog(null, "成功!
", "重新开始", JOptionPane.INFORMATION_MESSAGE);
736         } else
```

```
737             JOptionPane.showMessageDialog(null, "失败!",  
738 "重新开始", JOptionPane.ERROR_MESSAGE);  
739         }  
740     });  
741     mItemContinue.addActionListener(new ActionListener()  
742     {  
743         public void actionPerformed(ActionEvent e)  
744         {  
745             try {  
746                 if (word.setPstLine(word.readRecord()))  
747                 {  
748                     progressBarThread.end();  
749                     if (md.getText().equals("模式: 复习收藏词"))  
750                         newWord("mark");  
751                     else if (md.getText().equals("模式: 复习错词"))  
752                         newWord("wrong");  
753                     else if (md.getText().equals("模式: 记忆新词"))  
754                         newWord("word");  
755                     JOptionPane.showMessageDialog(null, "成功!",  
756 "继续上次", JOptionPane.INFORMATION_MESSAGE);  
757                 }  
758                 else  
759                     JOptionPane.showMessageDialog(null, "失败! 未知原因!", "继续上次", JOptionPane.ERROR_MESSAGE);  
760             }  
761             catch (IOException e2)  
762             {  
763                 JOptionPane.showMessageDialog(null, "失败! 历史文件不存在!", "继续上次", JOptionPane.ERROR_MESSAGE);  
764             }  
765         }  
766     });  
767     judgeAns.addActionListener(new ActionListener()  
768     {  
769         public void actionPerformed(ActionEvent e)  
770         {  
771             try  
772             {  
773                 judgeWord();  
774             }  
775             catch (HeadlessException e1)  
776             {  
777                 JOptionPane.showMessageDialog(null, "设备不支持!", "错误", JOptionPane.ERROR_MESSAGE);  
778             }  
779             catch (IOException e1)  
780             {  
781                 JOptionPane.showMessageDialog(null, "文件写入错误!", "错误", JOptionPane.ERROR_MESSAGE);  
782             }  
783         }  
784     });  
785 }
```

```
781         }
782     }
783 });
784 skip.addActionListener(new ActionListener()
785 {
786     public void actionPerformed(ActionEvent e)
787     {
788         progressBarThread.end();
789         progressBarThread = new ProgressBarThread();
790         progressBarThread.start();
791         judgeAns.setEnabled(true);
792         nLabel.setVisible(true);
793         tf1.setEnabled(true);
794         en.setVisible(true);
795         listModel.addElement(word.getEnWord());
796
797         if (md.getText().equals("模式: 复习收藏词"))
798             try
799             {
800                 newWord("mark");
801             }
802             catch (IOException e1)
803             {
804                 JOptionPane.showMessageDialog(null, "文件读取
805 错误!", "错误", JOptionPane.ERROR_MESSAGE);
806             }
807         else if (md.getText().equals("模式: 复习错词"))
808             try
809             {
810                 newWord("wrong");
811             }
812             catch (IOException e1)
813             {
814                 JOptionPane.showMessageDialog(null, "文件读取
815 错误!", "错误", JOptionPane.ERROR_MESSAGE);
816             }
817         else if (md.getText().equals("模式: 记忆新词"))
818             try
819             {
820                 newWord("word");
821             }
822             catch (IOException e1)
823             {
824             }
825         }
826 });
827 mark.addActionListener(new ActionListener()
828 {
```

```
829         public void actionPerformed(ActionEvent e)
830         {
831             try
832             {
833                 pgmk.setText("已背收藏词:
" + word.getmarkLine() + "/" + word.getmarkTotalLine());
834             }
835             catch (IOException e2)
836             {
837                 JOptionPane.showMessageDialog(null, "文件读取错误!
", "错误", JOptionPane.ERROR_MESSAGE);
838             }
839             if (countm == 0)
840             {
841                 try
842                 {
843                     word.mark();
844                 } catch (IOException e1)
845                 {
846                     JOptionPane.showMessageDialog(null, "文件写入
错误!", "错误", JOptionPane.ERROR_MESSAGE);
847                 }
848                 targetListModel.addElement(word.getEnWord());
849                 countm++;
850             }
851         }
852     });
853
854     f.setSize(600, 440);
855     f.setContentPane(panelContainer);
856     f.setLocationRelativeTo(null);
857     f.setJMenuBar(mMenBar);
858     f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
859     f.setVisible(true);
860     f.setResizable(false);
861
862     f.addWindowListener(new WindowAdapter()
863     {
864         public void windowClosing(WindowEvent e)
865         {
866             try
867             {
868                 word.saveRecord();
869             } catch (IOException e1)
870             {
871                 JOptionPane.showMessageDialog(null, "保存记录失败!
", "退出", JOptionPane.ERROR_MESSAGE);
872             }
873             super.windowClosing(e);
874         }
875     });
876 }
```

```
875     });
876     File file = new File("word.txt");
877     if (!file.exists())
878     {
879         progressBarThread.end();
880         JOptionPane.showMessageDialog(null, "请在软件目录下放置
word.txt 文件!", "词库不存在", JOptionPane.ERROR_MESSAGE);
881         System.exit(0);
882     }
883     if (word.readWord("word"))
884     {
885         en.setText(word.getEnWord());
886         cn.setText(word.getCnTrans());
887     } else
888     {
889         progressBarThread.end();
890         JOptionPane.showMessageDialog(null, "你已经背完了所有单词,
重新开始或者更新词库吧!", "没有单词了", JOptionPane.ERROR_MESSAGE);
891     }
892 }
893
894 public void newWord(String choice) throws IOException
895 {
896     pg.setText("进度 :
" + 100.0 * word.getwordLine() / word.getwordTotalLine() + "%");
897     pgwd.setText("已背单词:
" + word.getwordLine() + "/" + word.getwordTotalLine());
898     pgwr.setText("已背错词:
" + word.getwrongLine() + "/" + word.getwrongTotalLine());
899     pgmk.setText("已背收藏词:
" + word.getmarkLine() + "/" + word.getmarkTotalLine());
900
901     if (word.readWord(choice))
902     {
903         countm = 0;
904         count = 0;
905         en.setText(word.getEnWord());
906         cn.setText(word.getCnTrans());
907         progressBarThread.end();
908         progressBarThread = new ProgressBarThread();
909         progressBarThread.start();
910     }
911     else
912     {
913         JOptionPane.showMessageDialog(null, "你已经背完了所有单词,
重新开始或者更新词库吧!", "没有单词了", JOptionPane.ERROR_MESSAGE);
914     }
915 }
916
917 public void judgeWord() throws HeadlessException, IOException
```



```
918     {
919         pgwr.setText("已背错词:
" + word.getwrongLine() + "/" + word.getwrongTotalLine());
920         boolean r = false;
921         if (count == 0)
922             r = true;
923         if (word.judge(tf1.getText(), r))
924         {
925             tf1.setText("");
926             judgeAns.setEnabled(true);
927             nLabel.setVisible(true);
928             tf1.setEnabled(true);
929             en.setVisible(true);
930             listModel.addElement(word.getEnWord());
931             if (md.getText().equals("模式: 复习收藏词"))
932                 newWord("mark");
933             else if (md.getText().equals("模式: 复习错词"))
934                 newWord("wrong");
935             else if (md.getText().equals("模式: 记忆新词"))
936                 newWord("word");
937             count = 0;
938         }
939         else
940         {
941             if (count == 0)
942                 targetListModel.addElement(word.getEnWord());
943             count++;
944             JOptionPane.showMessageDialog(null, "请重试!", "拼写错误
", JOptionPane.INFORMATION_MESSAGE);
945         }
946     }
947
948     public static void main(String[] args) throws IOException
949     {
950         MainFrame window = new MainFrame();
951         window.run();
952     }
953 }
```

## 目录

一、	引言.....	2
二、	软件概述.....	2
1.	软件结构.....	2
2.	功能.....	2
三、	运行环境.....	2
1.	硬件环境.....	2
2.	软件环境.....	3
四、	技术细节.....	3
1.	Word 类.....	3
2.	MainFrame 类.....	3
五、	用户指南.....	3

## 一、 引言

随着学习英语的热情不断高涨，各种不同的手机客户端背单词软件 App，如“百词斩”、“扇贝单词”等涌现了出来。但是，对于托福，GRE 等这一类考试来说，其是通过电脑端进行考试，使用手机进行单词背诵不能契合这一特点，方便用户备考。

我设计的这款背单词软件采用 Java 面向对象程序设计，使用 Swing 包设计 UI 界面，实现电脑客户端的背单词软件。

## 二、 软件概述

### 1. 软件结构

```
.
  history.dat
  MainFrame.java
  mark.txt
  Word.class
  Word.java
  word.txt
  wrong.txt
```

### 2. 功能

本程序具有以下功能：

1. 进入主界面时，首先程序会智能判断程序目录下是否存在 word.txt 词库，如果不存在则会智能报错。
2. 在存在词库的前提下，程序将会自动智能加载 history.dat 来加载你上次的进度，并且进入主界面后自动采用记忆新词模式。
3. 程序主界面会提示你当前模式，当前进度，和已经背过的各类单词数量，还有倒计时剩余的时间。
4. 程序智能使用倒计时进度条，首先你拥有 15 秒的时间记忆新词。15 秒过去后，程序自动隐藏英文释义，隐藏进度条和倒计时提示标签，同时显示输入框和提交按钮，你需要输入正确的单词。如果输入错误，程序将自动创建 wrong.txt 记录你被错的单词程序将继续让你输入，直到输入正确的答案为止，自动进入下一个单词的记忆。
5. 在任何状态下，你都可以收藏和跳过单词。点击收藏按钮，程序会创建 mark.txt 来记录你的收藏单词。
6. 你可以通过菜单栏中的选项标签，选择你想要的记忆模式，比如复习错词，智能背单词程序的实现复习收藏词。
7. 程序界面显示了已背诵单词和错误及收藏单词，你可以随时进行查看。

## 三、 运行环境

### 1. 硬件环境

本软件的编写与测试运行在作者本人的个人笔记本电脑上。由于条件限制，作者未测试在其它硬件环境下的运行效果，理论上符合运行 Java 8 虚拟机硬件条件的，有 1G 以上的 RAM 和硬盘存储空间的所有 PC 机上都能运行本程序。

作者本人个人笔记本电脑配置：

**CPU: Intel Core i7-8550U 1.80GHZ**

**RAM: 8GB**

**硬盘: SSD 128GB 机械 1TB**

## 2. 软件环境

因为本程序为 Java 语言编写，因而凡是支持 Java 8 虚拟机的操作系统都可以编译运行本程序。作者的编译运行环境如下：

**Windows 10 Version 1909 (64 位)**

**JDK 8 Update 251 (64 位)**

## 四、 技术细节

### 1. Word 类

此类抽象词库的基本功能，在此类下，我设定了 `psline` 数组记录三种单词模式的进度数据，`wdline` 变量记录词库 `word.txt` 中单词的个数，`wrlne` 变量记录 `wrong.txt` 中单词的个数，`mkline` 变量记录词库 `mark.txt` 中单词的个数，`enword` 变量记录当前单词的英文，`cntrans` 变量记录当前单词的中文。

为了体现封装性，我将这些变量都设为 `private`，我还提供了这些变量必要 `get/set` 方法，构造方法提供这些数据的初始化值。

下面我将介绍此类中的主要方法：

1. `public boolean readWord(String choice)`: 此方法根据传入参数从三种单词模式选择一种模式，并更据进度得到单词，写入 `enword`，`cntrans` 变量中。返回值为布尔类型，采用了异常处理，用于判断获取是否成功。

2. `public boolean judge(String input, boolean r)`: 此方法用于判断输入的单词是否正确，第二个传入参数用于设定如果错误是否将该单词写入 `wrong.txt` 中。

3. `public void mark()` 用于收藏单词，写入 `mark.txt`。

4. `public void saveRecord()` 用于记录当前进度，写入 `history.dat` 中。

5. `public long[] readRecord()` 用于读取进度文件 `history.dat`，并自动加载到变量里。

### 2. MainFrame 类

该类主要是图形界面的框架编写，采用 `Swing` 图形包进行设计，`BoxLayout` 布局管理，根据逻辑性和通常使用习惯编写而成，主要分为以下几个部分。

1. 内嵌 `ProgressBarThread extends Thread`，用于进度条的显示。

2. `public void run()`方法，提供程序运行的框架设置，对不同的主界面控件进行设定。

3. `public void newWord(String choice)` 沿袭了 `word` 类中的 `readWord` 方法，采用逻辑判断等使其更符合 UI 程序的要求。

4. `public void judgeWord()` 同样沿袭了 `word` 类中的 `judge` 方法，采用逻辑判断等使其更符合 UI 程序的要求。

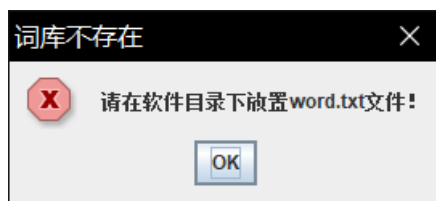
## 五、 用户指南

首先确保软件目录下存在词库文件 `word.txt`，词库中应该以左边英文单词，右边中文释义的形式存放，中英文之间可以以空格或者 `Tab` 分隔开，词库可以直接从 `Excel` 中导出。

示例如下：

English	中文
always	ad.总是，一直；永远
ability	n.能力；能耐，本领
able	a.有能力的；出色的
accomplish	vt.达到(目的)；完成
accommodation	n.招待设备；预定铺位
accommodate	vt.容纳；供应，供给

如果软件目录下没有 word.txt 词库文件，则会提示以下错误：



否则，软件便会正常启动，读取 history.dat 中的记录数据，默认进入“记忆新词”模式，在软件左上角你可以看到当前模式，背诵进度以及已背单词、错词、收藏词的统计，你有 15 秒钟的时间记忆词汇：



在倒计时期间，你可以随时点击跳过，该单词将被写入软件左下方“已背诵单词”中：



如果 15 秒倒计时已经过去，如果你认为单词较简单，你可以直接点击跳过，或者在输入框中输入单词来检验自己的记忆效果：



如果你输入的单词正确，将直接进入下一个单词的记忆，否则将会提示错误，该单词也将进入软件右下角“错误及收藏单词”中，并且写入 wrong.txt 文件中。

输入错误的单词：



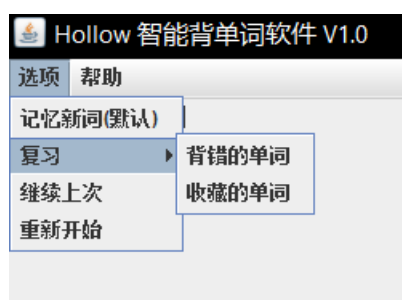
提示错误：



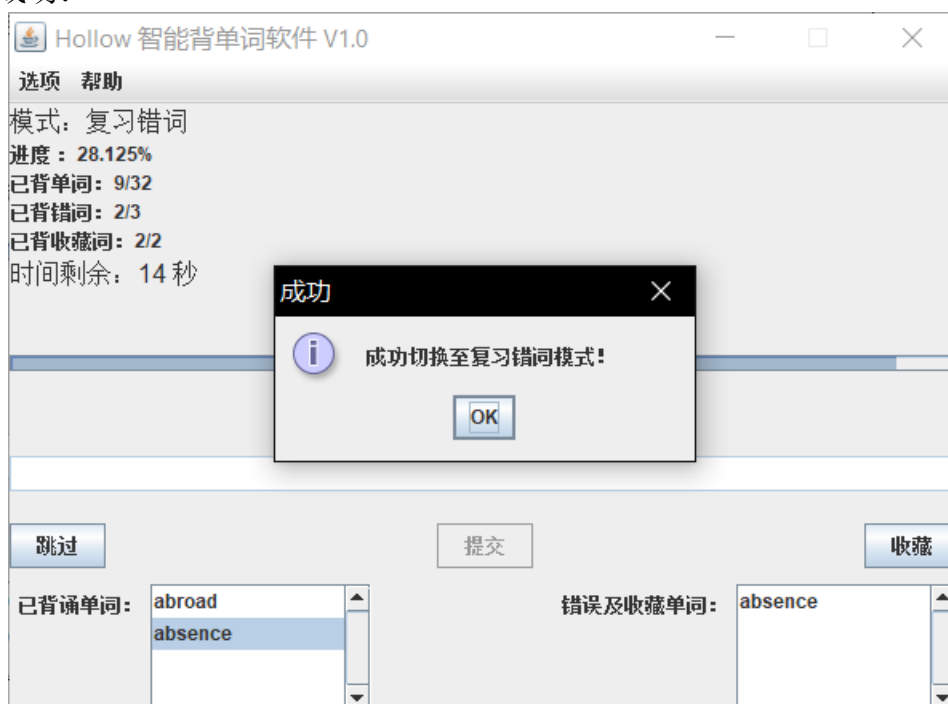
重试输入正确的单词，提交后：



你还可以点击左上角菜单栏中的选项选择模式:



提示切换成功:





如果此库中的单词已经背诵完成，请扩充 word.txt，或者点击选项中的重新开始清除背诵进度：



点击重新开始：



另外你还可以查看帮助菜单栏下的“使用说明”，获得一个简单的使用提示，以及“关于”，获得作者的信息。

