

# 背单词应用(WordMaster)

## 测试记录

测试员：11302010054王可嘉

2014.6.8

# 目 录

<b>I 引言</b>	<b>3</b>
1.1 编写目的	3
1.2 文档范围	3
1.3 读者对象	3
1.4 术语解释	3
<b>2 JUnit单元测试</b>	<b>3</b>
2.1 测试环境	3
2.2 测试对象	3
2.3 测试用例	4
2.3.1 <i>TestWordModel</i>	4
2.3.2 <i>TestWordBaseModel</i>	5
<b>3 其他单元测试</b>	<b>11</b>
3.1 测试环境	11
3.2 测试对象	11
3.3 测试用例	12

# I 引言

本测试记录文档是针对背单词应用(WordMaster)单元测试而编写的，包含JUnit单元测试以及其他单元测试的说明、测试过程说明以及测试结果记录等。

## I.1 编写目的

本测试记录文档的目的是保证软件测试的质量，把人为的因素减小到最小。同时将软件测试以文档的形式保存，便于以后测试的更新和软件的维护，也方便项目人员的交流。

## I.2 文档范围

针对背单词应用，文档范围控制在对背单词应用使用过程中的所有单元测试用例的范围之内。

## I.3 读者对象

测试人员，相关项目人员。

## I.4 术语解释

- WordMaster：背单词软件名称。
- 词库：通过单词开头字母划分的不同单词范围。例如，以A字母开头的为A词库，单词abandon、abbreviation都在词库A中。
- 起始单词：指定词库中的某个单词，以当前单词作为背诵的起点。

# 2 JUnit单元测试

本节详细描述了JUnit单元测试的测试用例说明、测试结果记录以及测试过程说明。其中测试用例的具体代码请参见源代码文件夹，JUnit代码中有详细注释。

## 2.1 测试环境

系统测试环境为装配了JRE(Java Runtime Environment)的操作系统以及Eclipse集成开发环境，测试辅助工具为JUnit单元测试

## 2.2 测试对象

在本次项目中，软件设计使用MVC框架，其中Model部分的功能代码由JUnit单元测试来进行测试，软件UI界面部分使用其他单元测试方法。Model部分包含的Java类有：

- ProcessModel.java：记录程序运行当中的状态，与应用UI界面相关，无需JUnit测试。

- **ReadDat.java**: 记录词典路径，不含有任何函数，无需JUnit测试。
- **UserModel.java**: 记录使用背单词用户的基本信息，作为可扩展的功能而保留。目前待实现，不含有任何函数，无需JUnit测试。
- **WordBaseModel.java**: 包含主要业务逻辑，主要功能聚集一体的关键Model类。内涵多个函数，需对每个函数进行JUnit单元测试。
- **WordModel.java**: 记录单词的基本属性，包含中文释义，英文拼写，是否背正确。只有对于相应属性的set和get函数，需要JUnit测试。

2.3 测试用例

对2.1中的每个测试对象编写测试用例说明、测试结果记录以及测试过程说明。

2.3.1 TestWordModel

- 测试对象（类）：**WordModel.java**
- 测试对象描述：记录单词的基本属性，包含中文释义，英文拼写，是否背正确。对于每一个属性设有set和get函数。

表1 TestWordModel测试用例详述（共3项）	
测试对象（函数）： <b>setChineseMean(String chinese)</b> , <b>getChineseMean()</b>	总编号：1-01
测试用例： <b>testSetGetChineseMean()</b>	编号：01
测试过程说明：对某一单词的中文释义进行设定和获取，查看输出值与预期是否一致	
输入： <b>String</b> “放弃”	
预期输出： <b>String</b> “放弃”	
实际输出： <b>String</b> “放弃”	
测试结果：成功	
测试对象（函数）： <b>setEnglishMean(String english)</b> , <b>getEnglishMean()</b>	总编号：1-02
测试用例： <b>testSetGetEnglishMean()</b>	编号：02

表1 TestWordModel测试用例详述（共3项）

测试过程说明：对某一单词的英文释义进行设定和获取，查看输出值与预期是否一致

输入：String “abandon”

预期输出：String “abandon”

实际输出：String “abandon”

测试结果：成功

测试对象（函数）：setIfCorrect(String ifcorrect) ,  
getIfCorrect()

总编号：1-03

测试用例：testSetGetIfCorrect()

编号：03

测试过程说明：单词背诵状态分为三种状态：0表示从未背过，1表示背诵正确，2表示背诵错误。其中针对一个单词只要至少有一次背诵正确，其状态均为1。本次测试用例中，对某一单词背诵状态进行设定和获取，查看输出值与预期是否一致

输入：String “0”

预期输出：String “0”

实际输出：String “0”

测试结果：成功

### 2.3.2 TestWordBaseModel

- 测试对象（类）：WordBaseModel.java
- 测试对象描述：包含主要业务逻辑，主要功能聚集一体的关键Model类。内涵多个函数，需对每个函数进行JUnit单元测试。

表2 TestWordBaseModel测试用例详述（共16项）

测试对象（函数）：readWord(String temp)

总编号：2-01

测试用例：testReadWord\_dic()

编号：01

测试过程说明：给定一行从词典获取的词条，调用被测函数，测试解析该行单词是否为相应的中文释义、英文拼写，以及自动设置该单词背诵状态为未背过（“0”）

表2 TestWordBaseModel测试用例详述（共16项）

输入：String "abroad adv.国外，海外"

预期输出：String "adv.国外，海外" "abroad" "0"

实际输出：String "adv.国外，海外" "abroad" "0"

测试结果：成功

测试对象（函数）：readWord(String temp)

总编号：2-02

测试用例：testReadWord\_rec()

编号：02

测试过程说明：给定一行从某一词库record文件中获取的词条，调用被测函数，测试解析该行单词是否为相应的中文释义、英文拼写，以及该单词背诵状态（“0”，“1”或“2”）

输入：String "abroad adv.国外，海外#0"

预期输出：String "adv.国外，海外" "abroad" "0"

实际输出：String "adv.国外，海外" "abroad" "0"

测试结果：成功

测试对象（函数）：inputFromDictionary(String letter)

总编号：2-03

测试用例：testInputFromDictionary()

编号：03

测试过程说明：给定某一词库（一个字母），测试从词典里取到该词库的所有单词，存入wordlist，测试内容包括测试单词数目和测试某一index处的单词的中文释义、英文拼写以及该单词背诵状态

输入：String "Z"

预期输出：int 11, String String "n.活动，尖啸声" "zip" "0"

实际输出：int 11, String String "n.活动，尖啸声" "zip" "0"

测试结果：成功

测试对象（函数）：setCurrentWord(String word)

总编号：2-04

测试用例：testSetCurrentWord\_true()

编号：04

测试过程说明：测试设置当前起始单词，给定一个正确单词拼写，判断拼写正确的情况

输入：String "zip"

表2 TestWordBaseModel测试用例详述（共16项）

预期输出: true

实际输出: true

测试结果: 成功

测试对象(函数): setCurrentWord(String word)

总编号: 2-05

测试用例: testSetCurrentWord\_false1()

编号: 05

测试过程说明: 测试设置当前起始单词, 给定一个错误单词拼写(字母), 判断拼写正确的情况

输入: String "ziilip"

预期输出: false

实际输出: false

测试结果: 成功

测试对象(函数): setCurrentWord(String word)

总编号: 2-06

测试用例: testSetCurrentWord\_false2()

编号: 06

测试过程说明: 测试设置当前起始单词, 给定一个错误单词拼写(数字), 判断拼写正确的情况

输入: String "123"

预期输出: false

实际输出: false

测试结果: 成功

测试对象(函数): judgeWord(String word)

总编号: 2-07

测试用例: testJudgeWord\_false()

编号: 07

测试过程说明: 判断用户在背单词过程中根据中文释义输入单词正确与否, 拼写错误的情况zip,全局变量相应的变化是否正确

输入: String "ziilip"

预期输出: int[] expecteds = {1, 10, 0, 1}

实际输出: int[] actuals = {1, 10, 0, 1}

表2 TestWordBaseModel测试用例详述（共16项）

测试结果：成功

测试对象（函数）：nextWord()

总编号：2-08

测试用例：testNextWord()

编号：08

测试过程说明：测试下一个单词是否使得index加一

输入：null

预期输出：int 6

实际输出：int 6

测试结果：成功

测试对象（函数）：judgeWord(String word)

总编号：2-09

测试用例：testJudgeWord\_true()

编号：09

测试过程说明：判断用户在背单词过程中根据中文释义输入单词正确与否，拼写正确的情况zipcode,全局变量相应的变化是否正确

输入：String "zipcode"

预期输出：int[] expecteds = {2, 9, 1, 1}

实际输出：int[] actuals = {2, 9, 1, 1}

测试结果：成功

测试对象（函数）：setCountWord(int count)

总编号：2-10

测试用例：testSetCountWord\_false()

编号：10

测试过程说明：设置背单词数目，大于允许数量，函数应该返回false。允许数量的计算方法为当前选定的起始单词至该词库最后一个单词直接的单词数目。

输入：int 6

预期输出：false

实际输出：false

测试结果：成功

测试对象（函数）：setCountWord(int count)

总编号：2-11



表2 TestWordBaseModel测试用例详述（共16项）

测试用例：testSetCountWord\_true1()

编号：11

测试过程说明：设置背单词数目，恰好等于允许数量，函数应该返回true。允许数量的计算方法为当前选定的起始单词至该词库最后一个单词直接的单词数目。

输入：int 5

预期输出：true

实际输出：true

测试结果：成功

测试对象（函数）：setCountWord(int count)

总编号：2-12

测试用例：testSetCountWord\_true2()

编号：12

测试过程说明：设置背单词数目，小于允许数量，函数应该返回false。允许数量的计算方法为当前选定的起始单词至该词库最后一个单词直接的单词数目。

输入：int 4

预期输出：true

实际输出：true

测试结果：成功

测试对象（函数）：findRecordText(String letter)

总编号：2-13

测试用例：testFindRecordText\_true()

编号：13

测试过程说明：测试在制定目录寻找目标record文件，找到的情况。本测试中手动写一个record文件，再调用被测函数进行查找，测试结束后将临时文件删除

输入：String "A"

预期输出：true

实际输出：true

测试结果：成功

测试对象（函数）：findRecordText(String letter)

总编号：2-14

表2 TestWordBaseModel测试用例详述（共16项）

测试用例：testFindRecordText\_false()

编号：14

测试过程说明：测试在制定目录寻找目标record文件，未找到的情况

输入：String "E"

预期输出：false

实际输出：false

测试结果：成功

测试对象（函数）：outputRecord()

总编号：2-15

测试用例：testOutputRecord()

编号：15

测试过程说明：测试将最新结果写入相应的record文件。这里手动设置一个词库letter，然后调用被测函数将当前背单词情况写入该词库的record文件。测试用例读取record文件中的内容与预期进行一一比对判断是否一致

输入：String "Z"

预期输出：String[] expecteds = {"11", "2", "9", "1", "1", "6"}; String[] expecteds2 = {"zeal n.热情，热忱#0", "zealous adj.热情的，热心的#0", "zebra n.斑马#0", "zero n.零，零度#0", "zinc n.锌#0", "zip n.活动，尖啸声#2", "zipcode n.邮政编码#1", "zipper n.拉链#0", "zone n.地带，区域，区#0", "zoo n.动物园#0", "zoology n.动物学#0"};

实际输出：String[] actuals = {"11", "2", "9", "1", "1", "6"}; String[] actuals2 = {"zeal n.热情，热忱#0", "zealous adj.热情的，热心的#0", "zebra n.斑马#0", "zero n.零，零度#0", "zinc n.锌#0", "zip n.活动，尖啸声#2", "zipcode n.邮政编码#1", "zipper n.拉链#0", "zone n.地带，区域，区#0", "zoo n.动物园#0", "zoology n.动物学#0"};

测试结果：成功

测试对象（函数）：inputFromRecord(String letter)

总编号：2-16

测试用例：testInputFromRecord()

编号：16

表2 TestWordBaseModel测试用例详述（共16项）

测试过程说明： 给定某一词库（一个字母），测试从指定record文件里取到该词库的所有单词，存入wordlist，并且将该词库的统计信息读出并计入相应变量。测试内容包括测试从record中读取的统计信息是否正确

输入：String "Z"

预期输出：int[] expecteds = {11, 2, 9, 1, 1, 6};

实际输出：int[] actuals = {11, 2, 9, 1, 1, 6};

测试结果：成功

## 3 其他单元测试

本节详细描述了除JUnit之外单元测试的测试用例说明、测试结果记录以及测试过程说明。

### 3.1 测试环境

系统测试环境为装配了JRE(Java Runtime Environment)的操作系统。

### 3.2 测试对象

在本次项目中，软件设计使用MVC框架，其中View和Controller的部分由JUnit单元测试较为困难，在本节主要利用其它直观的方法来进行界面和交互相关的单元测试。

- 登录界面（初始界面）
- 选择词库界面
- 选择起始单词界面
- 选择数量界面
- 背单词界面
- 本次统计信息界面
- 词库统计信息界面

### 3.3 测试用例

对3.1中的测试对象编写测试用例说明、测试结果记录以及测试过程说明。

- 测试对象（界面）：背单词应用中的所有界面
- 测试对象描述：在使用过程中涉及到的所有界面，即View。通过手动设置Controller，对程序的响应情况进行观察分析。

表3 UI界面测试用例详述（共7项）

测试对象（界面）：登录界面（初始界面）	总编号：3-01
预制条件：程序正常启动	编号：01
测试用例说明：启动程序，显示登录界面。在本次项目要求中无需登录也可进入应用。在本界面中右上角包含“设置”、“最小化”、“关闭”按钮，其中设置按钮的触发会调用主页换肤功能。直接点击“进入应用”按钮可以进入下一界面	
输入：无	
操作1：点击“进入应用”	
响应1：程序跳转进入下一界面	
操作2：点击“设置”	
响应2：界面背景图片更换	
操作3：点击“退出”或点击右上角“关闭”按钮	
响应3：程序退出	
测试结果：成功	
测试对象（界面）：选择词库界面	总编号：3-02
预制条件：程序登录成功，进入应用	编号：02
测试用例说明：由登录界面点击“进入应用”按钮后跳转至本页面。本页面中央设有26个字母所代表的26个词库，可以选择其中任一词库进行背单词。界面下方有三个按钮，分别是“查看统计信息”，“准备测试”，“退出”。其中“查看统计信息”跳转到词库的统计信息界面，“准备测试”跳转到选择起始单词界面，“返回”跳转至登录界面（初始界面）。	
输入：无	
操作1：点击“查看统计信息”	

表3 UI界面测试用例详述（共7项）

响应1：程序跳转进入统计信息界面

操作2：选择某一词库，点击“准备测试”

响应2：程序跳转进入选择起始单词界面

操作3：点击“返回”按钮

响应3：程序跳转至登录界面（初始界面）

测试结果：成功

测试对象（界面）： 选择起始单词界面

总编号： 3-03

预制条件：词库成功选择并导入

编号： 03

测试用例说明：由选择词库界面（选定词库后）点击“准备测试”按钮后跳转至本页面。本页面中央设有三个单选选项，分别是“继续测试”，“从头开始”，“自定义起始词汇”。其中“继续测试”继续上一次背诵的单词，“从头开始”从所选词库的第一个单词开始，选择“自定义起始词汇”显示“输入起始单词”的文本框。在文本框内输入合乎规范的单词，则从所输单词开始。选择完毕后，点击“开始测试”按钮，进入选择数量界面。

输入：无 或 起始单词

操作1：选择“继续测试”或“从头开始”，点击“开始测试”按钮

响应1：程序跳转进入选择数量界面

操作2：选择“自定义起始词汇”，输入正确起始单词

响应2：程序跳转进入选择数量界面

操作3：选择“自定义起始词汇”，输入不正确起始单词

响应3：程序跳转进入选择数量界面

测试结果：成功

测试对象（界面）： 选择数量界面

总编号： 3-04

预制条件：成功选择了起始词汇

编号： 04

表3 UI界面测试用例详述（共7项）

测试用例说明：由选择起始单词界面（选定起始单词后）点击“开始测试”按钮后，经进度条加载页面过渡至本页面。本页面设有“请选择背单词数量”提示的文本框。在文本框内输入数字作为选定的单词数量。点击“开始测试”按钮，进入背单词界面。

输入：背单词数量

操作1：点击“开始测试”按钮

响应1：程序跳转进入背单词界面

测试结果：成功

测试对象（界面）：背单词界面

总编号：3-05

预制条件：成功选择了背单词数量

编号：05

测试用例说明：由选择数量界面（输入数量后）点击“开始测试”按钮后跳转至本页面。本页面设有一个文本框，文本框上方为之前所选起始单词的中文释义。在文本框内输入中文释义对应的单词拼写，点击“下一个”按钮，系统做出判断。若拼写正确则直接进入下一个单词的测试，若不正确，显示错误信息，点击下一个后进入下一个单词的测试。界面右下方显示所剩余的单词数量。

输入：单词的英文拼写

操作1：拼写正确，点击“下一个”按钮

响应1：程序跳转进入下一个单词的背单词界面，右下角剩余单词数减少一个

操作2：拼写不正确，点击“下一个”按钮

响应2：程序在当前界面显示错误信息，用户点击“下一个”按钮后跳转进入下一个单词的背单词界面，右下角剩余单词数减少一个

测试结果：成功

测试对象（界面）：本次统计信息界面

总编号：3-06

预制条件：背完所选单词数量的单词

编号：06

测试用例说明：背完所有单词后，界面下方出现“完成”按钮，点击该按钮后跳转到本次统计信息界面。界面显示本次背单词流程的统计信息，包含背单词总数量，正确数量，错误数量，正确率

输入：无

表3 UI界面测试用例详述（共7项）

操作1：在最后一个单词界面完成后点击“完成”按钮

响应1：程序跳转进入本次信息统计界面，显示相关信息

测试结果：成功

测试对象（界面）：词库统计信息界面

总编号：3-07

预制条件：在选择词库界面成功选择了某一词库

编号：07

测试用例说明：由选择词库界面（选定词库后）点击“查看统计信息”按钮后跳转至本页面。本页面显示所选词库的统计信息，包含词库名，词库内单词总数量，已背单词数量，正确数量，错误数量，正确率

输入：无

操作1：在选择词库界面选定某一词库后点击“查看统计信息”

响应1：程序跳转进入词库统计界面，显示相关信息

测试结果：成功