

# 1 组手写代码 OCR 判分项目测试报告

撰写人：吴郑扬

## 1 引言

### 1.1 编写目的

编写本报告的目的是研究 Amending 软件的总体需求、背景。对开发结果、开发评价进行分析，得出经验与教训，本文档的预期读者为软件开发人员。

### 1.2 背景

该软件的名称为 Amending，该软件可以提高编程老师对手写代码题目批阅上的效率，该软件的测试环境为 Windows10。

### 1.3 术语和缩略词

无

### 1.4 参考资料

模板参考：[https://blog.csdn.net/qq\\_44528283/article/details/116062260](https://blog.csdn.net/qq_44528283/article/details/116062260)

## 2 测试用例设计

### (1) 打开文件路径操作

表 1：打开文件路径第一组测试用例

测试用例编号	Amending_Path_001
测试项目	打开文件路径操作
测试标题	输入文件路径的打开与记录
重要级别	高
预置条件	系统中存在该文件且文件后缀名为.input
输入	无
操作步骤	用户点击两个输入文件的打开按钮，分别选中两个提前预设定好的输入文件
预计输出	编辑框将会显示路径，且路径名记录到 Input_name1、Input_name2 中

表 2：打开文件路径第二组测试用例

测试用例编号	Amending_Path_002
测试项目	打开文件路径操作

测试标题	输出文件路径的打开与记录
重要级别	高
预置条件	系统中存在该文件且文件后缀名为.output
输入	无
操作步骤	用户点击两个输出文件的打开按钮，分别选中两个预设定好的输出文件
预计输出	编辑框将会显示路径，且路径名记录到 Output_name1、Output_name2 中

表 3：打开文件路径第三组测试用例

测试用例编号	Amending_Path_003
测试项目	打开文件路径操作
测试标题	答案文件路径的打开与记录
重要级别	高
预置条件	系统中存在该文件且文件后缀名为.correct
输入	无
操作步骤	用户点击两个答案文件的打开按钮，分别选中两个预设定好的答案文件
预计输出	编辑框将会显示路径，且路径名记录到 Correct_name1、Correct_name2 中

表 4：打开文件路径第四组测试用例

测试用例编号	Amending_Path_004
测试项目	打开文件路径操作
测试标题	存储代码文件路径的打开与记录
重要级别	高
预置条件	系统中存在该文件且文件后缀名分别为.txt、.cpp
输入	无
操作步骤	用户点击两个代码文件的打开按钮，分别选中两个新建好的空白 txt.cpp 文件
预计输出	编辑框将会显示路径，且路径名记录到 txt_name、cpp_name 中

## (2) 打分操作

表 5：打分第一组测试用例

测试用例编号	Amending_Score_001
测试项目	打分操作
测试标题	改变输入文件的内容观察对应输出文件的不同
重要级别	中
预置条件	在路径打开与选择中路径匹配
输入	改变系统中输入文件中的值
操作步骤	将输入文件中的内容由 1 2 3 4 改为 4 3 2 1 观察输出文件中内容的变化
预计输出	输出文件的内容由 $4+6i$ 、 $-2-2i$ 变为 $6+4i$ 、 $2+2i$

表 6：打分第二组测试用例

测试用例编号	Amending_Score_002
测试项目	打分操作
测试标题	改变答案文件的内容观察对应结果判决的不同
重要级别	中
预置条件	在路径打开与选择中路径匹配
输入	改变系统中答案文件中的值
操作步骤	将答案文件的内容由 4+6i、-2-2i 转变为 6+4i、2+2i
预计输出	结果输出框中将从该测试点通过转变为该测试点不通过

(3) 整体功能测试

表 7：整体功能测试 1

测试用例编号	Amending_001
测试项目	整体功能测试
测试标题	对运算符重载实现复数加减法题目进行测试
重要级别	中
输入	第一个输入文件写入：1 2 3 4；第二个输入文件写入：4 3 2 1 第一个答案文件写入：4+6i -2-2i；第二个答案文件写入：6+4i 2+2i
预计输出	第一个输出文件将显示：4+6i、-2-2i；第二个输出文件将显示：6+4i， 2+2i，.txt 文件记录下来了未修正的识别源码，.cpp 文件记录下来了修正后的源码，结果输出框将输出：第一个测试用例通过，第二个测试用例通过

表 8：整体功能测试 2

测试用例编号	Amending_002
测试项目	整体功能测试
测试标题	对任意手写大篇幅代码进行测试(以链表创建查询为例)进行测试
重要级别	中
输入	第一个输入文件写入：Bob 99 Cindy 100；第二个输入文件写入：Jack 73 Jessica 85。第一个答案文件写入：Bob 99 Cindy 100；第二个答案文件写入：Jack 73 Jessica 85。
预计输出	第一个输出文件将显示：Bob 99 Cindy 100；第二个输出文件将显示：Jack 73 Jessica 85，.txt 文件记录下来了未修正的识别源码，.cpp 文件记录下来了修正后的源码，结果输出框将输出：第一个测试用例通过，第二个测试用例通过

3 测试结果及分析

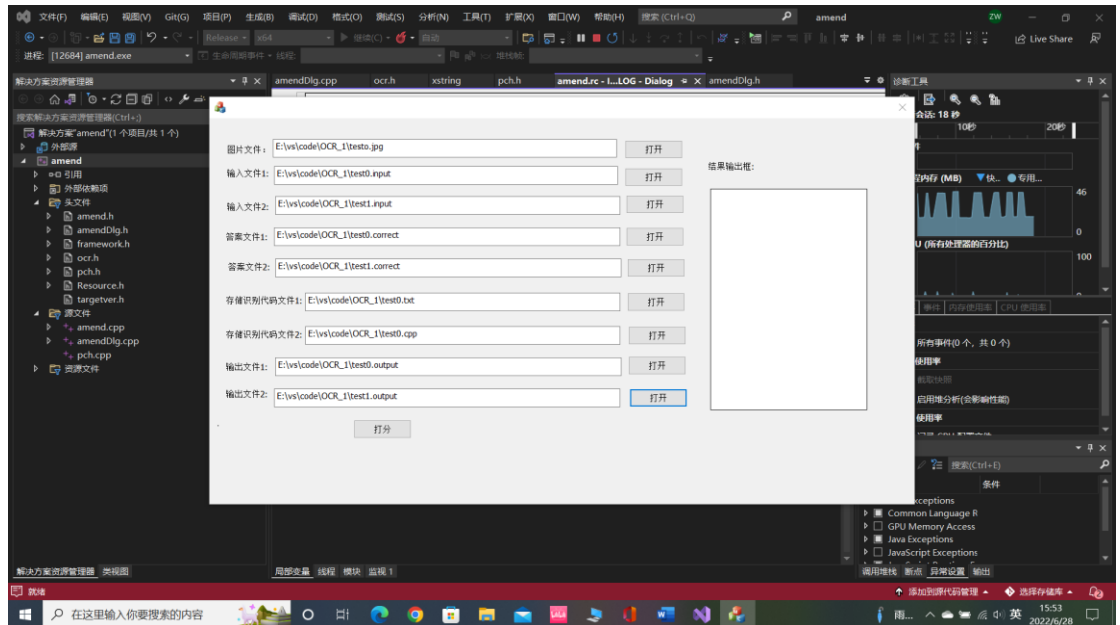
3.1 测试环境

该软件的测试环境为 Windows10 系统。

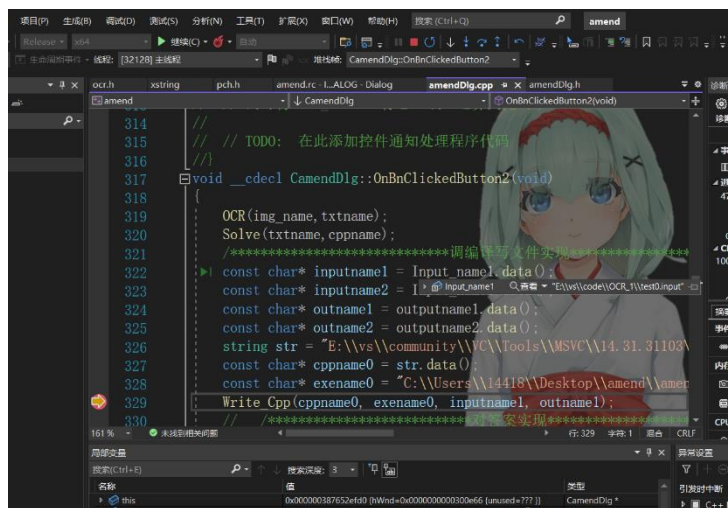
## 3.2 测试用例执行

### (1) 测试打开文件路径操作

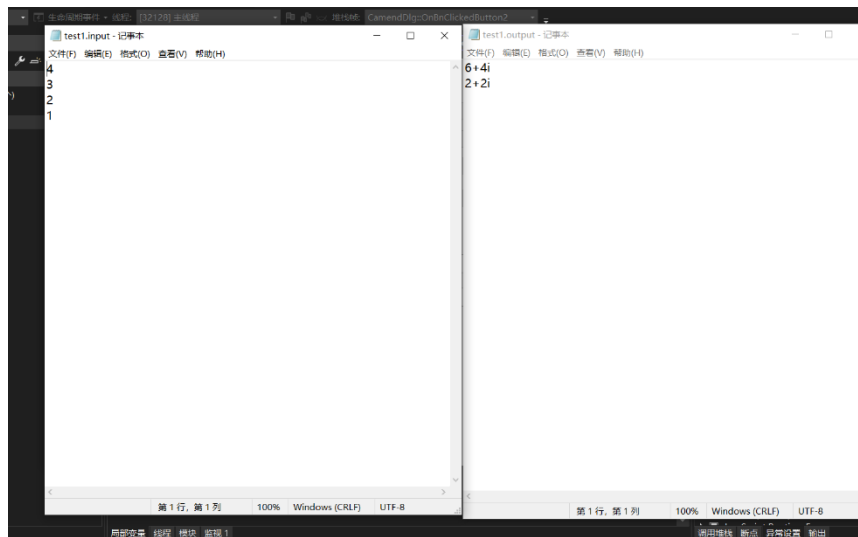
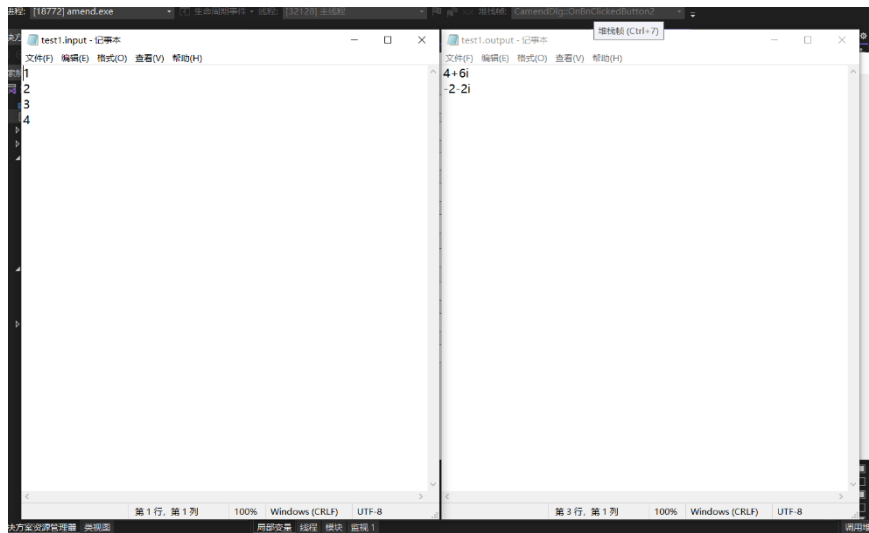
分别执行 Amending\_Path\_001 至 Amending\_Path\_005 操作如下图所示



从图中可以看到通过选中本地的文件，输入文件、输出文件、答案文件、存储代码文件以及图片文件均被记录在了编辑框中。

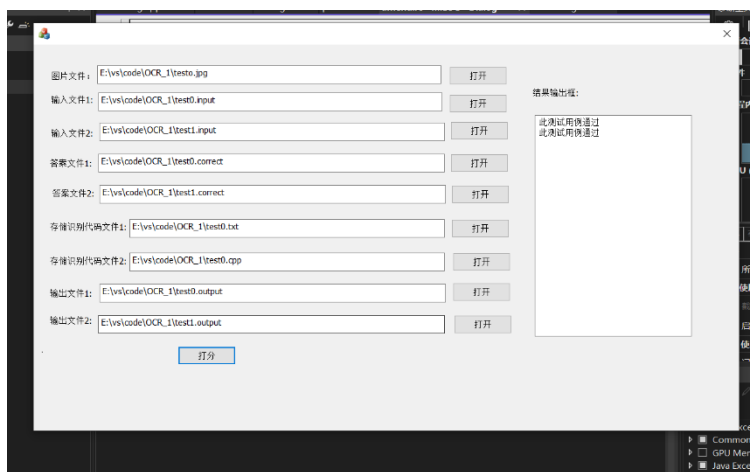


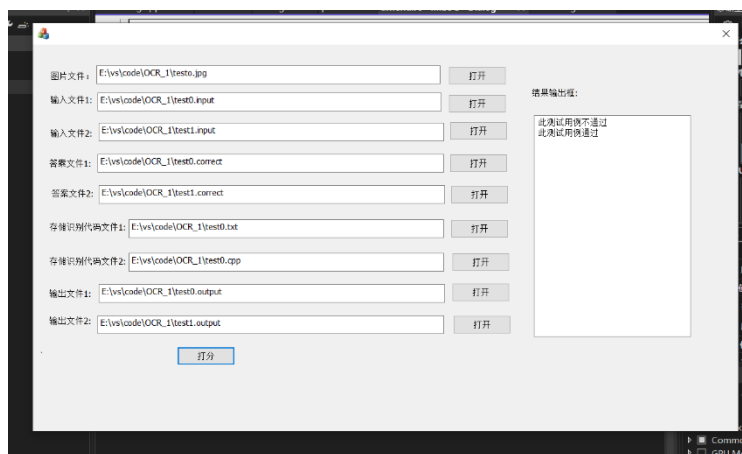




可以看出通过改变输入值，对应的输出值发生了变化，测试用例通过！

II) 执行 Amending\_Score\_002 操作，发生如下变化：





在结果输出框中由测试用例通过转变为测试用例不通过, 这说明测试用例 Amending\_Score\_002 通过!

### (3) 测试整体功能:

由以上可以看出针对于以运算符重载方式实现复数加减法的运算的可行性, Amending\_001 通过。但是当代码量达到一定程度以后, OCR 的结果将会发生错乱, 再加上手写的不确定性, 往往会产生微小错误, 而这一小错误就会使得程序无法编译, 而失去判卷功能。所以对于代码量较大的测试用例 Amending\_002 并没有通过。

## 3.3 结论及建议

(1) 该软件在一定条件下可以实现对简单手写代码的判决, 但当代码长度增加, 手写不标准等问题出现时, 会出现识别错误的问题。这是一个较难解决的问题, 解决此问题的方法只有提高 OCR 的识别率, 而这对我们来说是难以做到的, 因为我们也是调用百度的官方 OCR 接口。一旦识别率不能达到百分之百, 就会使得结果面临一个严峻的无解问题, 由 OCR 识别后生成的源文件, 如果因为语法错误没有通过编译。那么, 这些语法错误的产生存在两种可能的情况, 一是考生答卷上本身就有的语法错误; 二是 OCR 识别错误。如果 OCR 无法百分百识别正确, 那么当识别后生成的源文件无法通过编译时, 就无法确定语法错误属于上述两种情况中的哪一种。

(2) 建议: 目前利用该软件的比较好的办法是加入人工修正。当然软件也在不断完善之中。