实验报告一

|  |  |
| --- | --- |
| 姓名 | Fyl |
| 学号 | SA18225079 |
| 工具 | Win10；Dev C++； |

目录

目录

[1. 引言](#_Toc29454_WPSOffice_Level1) [3](#_Toc29454_WPSOffice_Level1)

[1.1. 标识](#_Toc31057_WPSOffice_Level2) [3](#_Toc31057_WPSOffice_Level2)

[1.2. 系统概述](#_Toc27497_WPSOffice_Level2) [3](#_Toc27497_WPSOffice_Level2)

[1.3. 文档概述](#_Toc5303_WPSOffice_Level2) [4](#_Toc5303_WPSOffice_Level2)

[2. 引用文件](#_Toc31057_WPSOffice_Level1) [4](#_Toc31057_WPSOffice_Level1)

[3. 测试结果概述](#_Toc27497_WPSOffice_Level1) [4](#_Toc27497_WPSOffice_Level1)

[3.1. 对被测试软件的总体评估](#_Toc3130_WPSOffice_Level2) [4](#_Toc3130_WPSOffice_Level2)

[3.2. 测试环境的影响](#_Toc13532_WPSOffice_Level2) [5](#_Toc13532_WPSOffice_Level2)

[3.3. 改进建议](#_Toc4383_WPSOffice_Level2) [5](#_Toc4383_WPSOffice_Level2)

[4. 详细的测试结果](#_Toc5303_WPSOffice_Level1) [5](#_Toc5303_WPSOffice_Level1)

[4.1. 等价类划分测试（NumberTransform-ecdiv）](#_Toc11983_WPSOffice_Level2) [6](#_Toc11983_WPSOffice_Level2)

[4.1.1. 测试用例设计](#_Toc27497_WPSOffice_Level3) [6](#_Toc27497_WPSOffice_Level3)

[4.2. 边界值测试（NumberTransform-boundary）](#_Toc24459_WPSOffice_Level2) [7](#_Toc24459_WPSOffice_Level2)

[4.2.1. 测试用例设计](#_Toc13532_WPSOffice_Level3) [8](#_Toc13532_WPSOffice_Level3)

[5. 测试记录](#_Toc3130_WPSOffice_Level1) [9](#_Toc3130_WPSOffice_Level1)

[6. 评价](#_Toc13532_WPSOffice_Level1) [10](#_Toc13532_WPSOffice_Level1)

[6.1. 能力](#_Toc29552_WPSOffice_Level2) [10](#_Toc29552_WPSOffice_Level2)

[6.2. 缺陷与限制](#_Toc13700_WPSOffice_Level2) [10](#_Toc13700_WPSOffice_Level2)

[6.3. 建议](#_Toc2481_WPSOffice_Level2) [10](#_Toc2481_WPSOffice_Level2)

[6.4. 结论](#_Toc32094_WPSOffice_Level2) [11](#_Toc32094_WPSOffice_Level2)

[7. 测试活动总结](#_Toc4383_WPSOffice_Level1) [11](#_Toc4383_WPSOffice_Level1)

[7.1. 人力消耗](#_Toc21354_WPSOffice_Level2) [11](#_Toc21354_WPSOffice_Level2)

[7.2. 物质资源消耗](#_Toc30977_WPSOffice_Level2) [11](#_Toc30977_WPSOffice_Level2)

[8. 注解](#_Toc11983_WPSOffice_Level1) [12](#_Toc11983_WPSOffice_Level1)

# 引言

## 标识

本文档适用系统：Windows 10；

本文档使用软件：NumberTransform.exe

## 系统概述

本文档测试“人民币数字大写转换”程序，具体功能如下：

1）中文大写金额数字应用壹、贰、叁、肆、伍、陆、柒、捌、玖、拾、佰、仟、万、亿、元、角、分、零、整(正)等字样。

2）中文大写金额数字到"元"为止的，在"元"之后，应写"整"(或"正")字，在"角"之后，可以不写"整"(或"正")字。

3）中文大写金额数字前应标明"人民币"字样，大写金额数字有"分"的，"分"后面不写"整"(或"正")字。

4）大写金额数字应紧接"人民币"字样填写，不得留有空白。

5）阿拉伯数字小写金额数字中有"0"时，中文大写应按照汉语语言规律、金额数字构成和防止涂改的要求进行书写。

## 文档概述

本文档为上述“人民币数字大写转换”程序的黑盒测试实验报告。

# 引用文件

·课程内容说明文档

·《计算机软件文档编制规范》——GB/T8567-2006

# **测试结果概述**

## **对被测试软件的总体评估**

软件当前已实现上述的基本功能。软件输入为：0-9的数字与小数点。软件的输出为汉字。程序为中科大软件学院18软设张思浩设计，开发使用Dev C++，编写使用C++程序设计语言。

程序现有以下缺陷：

①当输入非法数据如‘ xxx ’时，程序错误。

②程序对精度大于分位的数据进行处理时没有进行四舍五入；

③程序提示过于简单，未实现软件的图形化，测试时稍有不便。

## **测试环境的影响**

源程序在Mac环境下编写，测试在Win10环境下测试，因此对于测试在不同机种上的运行结果会有不全面性。

## **改进建议**

建议解决输入非数字数据时出现的问题；

建议实现软件的图形化，如使用MFC等；

建议实现高精度数据的四舍五入，使用判断取整等。

# **详细的测试结果**

黑盒测试又称功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。本次测试将采用等价类划分方法，边界值方法来进行测试。

## **等价类划分测试（NumberTransform-ecdiv）**

|  |  |
| --- | --- |
| 标识符 | NumberTransform-ecdiv |
| 测试项 | 用等价类划分方法来设计用例测试 NumberTransform.exe 的人民币大小写转换功能； |
| 方法原理 | 把所有可能的输入数据，即程序的输入域划分为若干个等价类（子集），然后从每一个子集中选取少数具有代表性的数据作为测试用例； |
| 测试环境要求 | Windows 10，Dev C++； |
| 预期输出 | 正确转换 或 提示错误； |
| 测试人 | 费玉玲 |

表4.1 等价类划分标识符

### 测试用例设计

按照以下表格进行等价类划分测试：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例 | 操作 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1-1 | 不输入数据，并进行转换 | - | 提示不执行 | 程序进入死循环 | Fail |
| 1-2 | 输入非法数据，并执行转换 | xxx | 提示不执行 | 程序无法执行，直接退出 | Fail |
| 1-3 | 输入多个小数点，并执行转换 | 1...1 | 提示不执行 | 人民币壹元 | Fail |
| 1-4 | 输入超大数据 | 100000000000000000000000000 | 提示不执行 | 程序无法执行，直接退出 | Fail |
| 1-5 | 输入负值 | -1 | 提示不执行 | 程序无法执行，直接退出 | Fail |
| 1-6 | 输入高精度数据 | 1004.135 | 人民币壹仟零肆元壹角四分 | 人民币壹仟零肆元壹角叁分 | Pass |
| 1-7 | 输入包含连续零的数据 |
| 1-8 | 输入不含零的数据 | 1234.56 | 人民币壹仟贰佰叁拾肆元伍角陆分 | 人民币壹仟贰佰叁拾肆元伍角陆分 | Pass |
| 1-9 | 输入角位是“0”，分位不是“0”的数据 | 1.01 | 人民币壹元零壹分 | 人民币壹元零壹分 | Pass |
| 1-10 | 输入前几位是“0”的数据 | 001223 | 人民币壹仟贰佰贰拾叁元整 | 人民币万零壹仟贰佰贰拾叁元整，程序继续运行 | Fail |
| 1-11 | 输入到元为止的数据 |
| 1-12 | 输入角位、分位都为“0”的数据 | 1.00 | 人民币壹元整 | 人民币壹元整 | Pass |

表4-2 等价类划分测试用例

## **边界值测试（NumberTransform-boundary）**

|  |  |
| --- | --- |
| 标识符 | NumberTransform-boundary |
| 测试项 | 用边界值划分方法来设计用例测试 NumberTransform.exe 的人民币大小写转换功能； |
| 方法原理 | 选取正好等于、 刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值。 |
| 测试环境要求 | Windows 10，Dev C++； |
| 预期输出 | 正确转换 或 提示错误； |
| 测试人 | 费玉玲 |

表4-3 边界值测试标识符

本条应由项目唯一标识符标识一个测试，并且分为以下几条描述测试结果。

### **测试用例设计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例 | 操作 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 2-1 | 输入Max value+1，并进行转换 | 10000000000000.00 | 提示不执行 | 人民币壹拾元整,程序继续执行 | Fail |
| 2-2 | 输入Max value，并执行转换 | 999999999 999.99 | 人民币玖仟玖佰玖拾玖亿玖仟玖佰玖拾玖万玖仟玖佰玖拾玖元玖角玖分 | 人民币玖仟玖佰玖拾玖亿玖仟玖佰玖拾玖万玖仟玖佰玖拾玖元玖角玖分 | Pass |
| 2-3 | 输入Min value，并执行转换 | 0.00 | 人民币零元整 | 人民币零元整 | Pass |
| 2-4 | 输入Min value-1，并进行转换 | -1 | 提示不执行 | 程序无法执行，直接退出 | Fail |

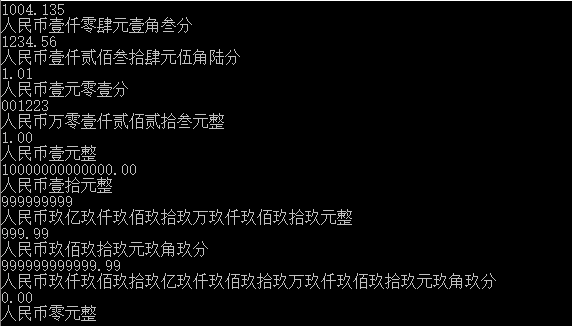
# **测试记录**

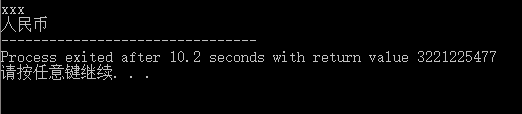
本次黑盒测试记录如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 测试日期 | 测试时间 | 测试地点 | 测试环境 |
| 等价类划分方法测试 | 2018.12.25 | 20：00 | 软件学院404 | Win 10 |
| 边界值方法测试 | 2018.12.25 | 21：00 | 软件学院404 | Win 10 |

表5.1 测试记录

本次黑盒测试部分截图：





# **评价**

## **能力**

基本实现了小写数字转换成人民币大写的功能。

## **缺陷与限制**

程序现有以下缺陷：

①当输入非法数据如‘ xxx ’时，程序无法执行；

②程序对精度大于分位的数据进行处理时没有进行四舍五入；

③未实现软件的图形化，程序提示太过简单，测试不便。

## **建议**

建议解决输入非数字数据时出现的问题；

建议实现软件的图形化，如使用 MFC等,

建议实现高精度数据的四舍五入，使用判断取整等

## **结论**

软件基本实现了“人民币大小写转换” 程序的需求，但是在输入一些非法字符时，没有相应提示，对于本次测验结果来说，通过半数的测试用例，基本合格。

# **测试活动总结**

## **人力消耗**

|  |  |
| --- | --- |
| 程序编写时间 | 1人\*1天 |
| 测试时间 | 1人\*1天 |
| 文档编写时间 | 1人\*1天 |
| 总人力消耗 | 3人天 |

表7.1 人力消耗

## **物质资源消耗**

|  |  |
| --- | --- |
| 程序编写消耗 | Mac pro |
| 程序测试消耗 | LENOVO ideapad 310S |

表7.2 物质资源消耗

# **注解**

原理注解：

等价类划分原理：等价类划分法是一种典型的、 重要的黑盒测试方法，它将程序所有可能的输入数据（有效的和无效的）划分成若干个等价类。然后从每个部分中选取具有代表性的数据当做测试用例进行合理的分类，测试用例由有效等价类和无效等价类的代表组成，从而保证测试用例具有完整性和代表性。利用这一方法设计测试用例可以不考虑程序的内部结构，以需求规格说明书为依据，选择适当的典型子集，认真分析和推敲说明书的各项需求，特别是功能需求，尽可能多地发现错误。等价类划分法是一种系统性的确定要输入的测试条件的方法。

边界值测试原理：边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界，针对各种边界情况设计测试用例。使用边界值分析方法设计测试用例，首先应确定边界情况。通常输入等价类与输出等价类的边界，就是应着重测试的边界情况。应当选取正好等于，刚刚大于，或刚刚小于边界的值做为测试数据，而不是选取等价类中的典型值或任意值做为测试数据。