

实验一 黑盒测试

1. 引言

1.1 标识

本文档适用于以下测试环境

系统：Windows 11

CPU: AMD Ryzen 9 7940HX

1.2 测试系统概述

本文档测试程序“找零钱”，功能如下：

- i. 输入付款和商品价格，输出找给顾客的零钱有几张几元的货币
- ii. 商品价格、付款金额应为整数且小于等于 100 元
- iii. 商品价格、付款金额应为阿拉伯数字
- iv. 找零货币有 1、2、10、20、50 六种面值

2. 测试需求

2.1 输入要求

- i. 确保输入的付款金额、商品价格是整数
- ii. 确保输入的付款金额、商品价格不超过 100 元不低于 0 元
- iii. 确保输入的付款金额、商品价格是阿拉伯整型数字

2.2 输出正确性

正确输入时，程序应能够计算出最佳的找零组合，也即计算出找零使用的最少货币张数以及每张面值

3. 测试结果

3.1 等价类划分

标识符	Test-charge
测试项	用等价划分方法来设计测试程序的找零钱功能的用例
方法原理	将程序的输入域划分为若干个等价类，然后从每个等价域中选取少量代表性数据作为测试用例

测试环境	Windows 11, VS Code
预期输出	正确的找零方法或者输出错误提示
测试人	赵乐君

表 3.1 等价类划分标识符

3.1.1 测试用例设计

测试用例	操作	输入数据	预期结果	实际输出	状态
1-1	不输入数据	-	非法输入	非法输入	Pass
1-2	输入字符串	付款金额 =aaaaa 商品价格 =10	非法输入	非法输入	Pass
1-3	输入超过上限数据	付款金额 =10000 商品价格 =10	非法输入	非法输入	Pass
1-4	输入低于下限数据,	付款金额= -1 商品价格 =10	非法输入	非法输入	Pass
1-5	输入数据为浮点型数字	付款金额 =1.5 商品价格 =10	非法输入	非法输入	Pass
1-6	输入数据有前置 0	付款金额 =0001 商品价格 =10	非法输入	非法输入	Pass
1-7	付款金额小于商品价格	付款金额=1 商品价格=2	非法输入	非法输入	Pass
1-8	输入合法数据	付款金额 =20 商品价格 =10	找零一张十元纸币	应找零 1 张 10 元纸币	Pass
1-9	输入合法数据	付款金额 =55 商品价格 =40	找零一张十元纸币、一张五元纸币	应找零 1 张 10 元纸币、1 张 5 元纸币	Pass
1-10	输入合法数据	付款金额 =50 商品价格 =37	找零一张十元纸币、一张两元纸币、一张一元纸币	应找零 1 张 10 元纸币、1 张 2 元纸币、1 张 1 元纸币	Pass

1-11	输入合法数据	付款金额=99 商品价格=88	找零一张十元纸币、一张一元纸币	应找零 1 张 10 元纸币、1 张 1 元纸币	Pass
------	--------	--------------------	-----------------	--------------------------	------

表 3.2 等价类划分测试

3.2 边界值分析

边界值分析法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。

使用边界值分析法设计测试用例，首先应确定边界情况，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。

3.2.1 测试用例设计

测试用例	操作	输入数据	预期结果	实际输出	状态
2-1	输入最大金额	付款金额=100 商品价格=100	无需找零	无需找零	Pass
2-2	输入最大金额加一	付款金额=101 商品价格=100	非法输入	非法输入	Pass
2-3	输入最小金额	付款金额=0 商品价格=0	无需找零	无需找零	Pass
2-4	输入最小金额-1	付款金额=-1 商品价格=0	非法输入	非法输入	Pass
2-5	找零 0 元	付款金额=1 商品价格=1	无需找零	无需找零	Pass
2-6	找零 1 元	付款金额=2 商品价格=1	找零一张一元纸币	找零 1 张 1 元纸币	Pass
2-7	找零 2 元	付款金额=3 商品价格=1	找零一张两元纸币	找零 1 张 2 元纸币	Pass
2-8	找零 4 元	付款金额=5 商品价格=1	找零两张两元纸币	找零 2 张 2 元纸币	Pass
2-9	找零 5 元	付款金额=10 商品价格=5	找零一张五元纸币	找零 1 张 5 元纸币	Pass
2-10	找零 6 元	付款金额=7 商品价格=1	找零一张五元纸币、一	找零 1 张 5 元纸币、1	Pass

			张一元纸币	张 1 元纸币	
2-11	找零 9 元	付款金额 =10 商品价格=1	找零一张五元纸币、两张两元纸币	找零 1 张 5 元纸币、2 张 2 元纸币	Pass
2-12	找零 10 元	付款金额 =11 商品价格=1	找零一张十元纸币	找零 1 张 10 元纸币	Pass
2-13	找零 11 元	付款金额 =20 商品价格=9	找零一张十元纸币、一张一元纸币	找零 1 张 10 元纸币 1 张 1 元纸币	Pass
2-14	找零 19 元	付款金额 =20 商品价格=1	找零一张十元纸币、一张五元纸币、两张两元纸币	找零 1 张 10 元纸币、1 张 5 元纸币、2 张 2 元纸币	Pass
2-15	找零 20 元	付款金额 =30 商品价格=10	找零一张二十元纸币	找零 1 张 20 元纸币	Pass
2-16	找零 21 元	付款金额 =40 商品价格=19	找零一张二十元纸币、一张一元纸币	找零 1 张 20 元纸币、1 张 1 元纸币	Pass
2-17	找零 49 元	付款金额 =50 商品价格=1	找零两张二十元纸币、一张五元纸币、两张二元纸币	找零 2 张 20 元纸币、1 张 5 元纸币、2 张 2 元纸币	Pass
2-18	找零 50 元	付款金额 =100 商品价格=50	找零一张五十元纸币	找零 1 张 50 元纸币	Pass
2-19	找零 51 元	付款金额 =100 商品价格=49	找零一张五十元纸币、一张一元纸币	找零 1 张 50 元纸币、1 张 1 元纸币	Pass
2-20	找零 99 元	付款金额 =100 商品价格=1	找零一张五十元纸币、两张二十元纸币、一张五元纸币、两张二元纸币	找零 1 张 50 元纸币、2 张 20 元纸币、1 张 5 元纸币、2 张 2 元纸币	Pass

表 3.3 边界值测试

3.3 因果图测试

3.3.1 测试用例设计

编号		3-1	3-2	3-3	3-4	3-5	3-6	3-7
原因	①	0	1	0	0	1	1	0
	②	0	0	1	1	1	1	1
	③	1	1	0	1	1	1	0
	④	0	0	0	0	0	0	0
	⑤	0	0	1	0	0	1	1
测试用例		付款金额=10 商品价格=5	付款金额=010 商品价格=5	付款金额=10.5 商品价格=10	付款金额=10 商品价格=20	付款金额=012.3 商品价格=10	付款金额=032.1 商品价格=40	付款金额=12.3 商品价格=20
预期结果		找零一张五元纸币	提示非法输入	提示非法输入	提示付款金额不足	提示非法输入	提示非法输入	提示非法输入
实际结果		找零 1 张 5 元纸币	提示非法输入	提示非法输入	提示付款金额不足	提示非法输入	提示非法输入	提示非法输入
测试状态		Pass	Pass	Pass	Pass	Pass	Pass	Pass

表 3.4 因果图测试

4. 测试结果分析

4.1 对被测试程序的总体评估

该程序能够正确处理用户的请求，对于非法的输入能够进行判断并进行错误提示

4.2 测试环境的影响

与系统环境无关

4.3 测试改进建议

测试用例覆盖范围应该更广

5. 软件评价

5.1 整体评价

程序基本满足用户需求，操作简单，提示完备，健壮性好，能够处理非法输入。

5.2 缺点与不足

- i. 使用不便，需要在 IDE 中才能运行
- ii. 付款金额的输入只设置了 0-100 的限制，没有考虑实际情况下不合理的输入，如：商品价格 98 元，实际情况下付款金额只能是 98 元或是 100 元，不可能是 99 元。
- iii. 对于输入范围的限制过于严格，上限 100 元过低。
- iv. 非法操作提示不够详细，应提示更加具体的非法原因

5.3 总结

程序基本完成了计算最佳找零组合的需求，对于大部分非法输入有错误提示，对于合法输入也有正确的返回结果，基本满足需求。

6. 测试记录

测试项目	测试日期	测试时间	测试人员	测试环境
功能测试	2024/11/4	11:08	赵乐君	Windows 11 AMD Ryzen 9 7940HX