# 实验一 黑盒测试用例设计实验

# 引言

## 标识

本文档适用于以下测试环境

系统：MacOS，version 14.3 (23D56)

CPU：Apple M1 pro

## 系统概述

本文档测试软件为“找零钱最佳组合程序”，具体功能如下：

1. 输入付款金额和商品价格，程序计算找零钱的最佳组合（找给顾客货币最少张数）
2. 商品价格应为一个整数且不大于100元。
3. 商品价格应为阿拉伯数字。
4. 找零货币只有50，20，10，5，2，1六种面值。

# 测试需求

## 输入验证

1）确保输入的是一个整数。

2）确保输入的不超过100元。

3）确保输入的是阿拉伯数字。

## 输出正确性

当正确输入后，程序应能正确计算出找零的最佳组合，即货币的最少张数。

# 测试执行结果

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，因果图方法来进行测试。

## 3.1 等价类划分方法

|  |  |
| --- | --- |
| 标识符 | test-charge |
| 测试项 | 用等价类划分方法来设计用例测试test的找零钱最佳组合功能 |
| 方法原理 | 把程序的输入域划分为若干个等价类（子集），然后从每个子集中选取少数有代表性的数据作为测试用例 |
| 测试环境要求 | macOS14.2，Clion 2024.2 |
| 预期输出 | 正确找零或者提示错误 |
| 测试人 | 赵建桥 |

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 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1-1 | 不输入数据，并进行找零 | - | 提示非法输入 | 提示非法输入 | Pass |
| 1-2 | 输入非法数据并执行找零 | 付款金额=abc | 提示非法输入 | 提示非法输入 | Pass |
| 1-3 | 输入超大数据 | 付款金额=999 | 提示非法输入 | 提示非法输入 | Pass |
| 1-4 | 输入负值 | 付款金额=-1 | 提示非法输入 | 提示非法输入 | Pass |
| 1-5 | 输入高精度数据 | 付款金额=1.1 | 提示非法输入 | 提示非法输入 | Pass |
| 1-6 | 输入数据前几位是0 | 付款金额=00015 | 提示非法输入 | 提示非法输入 | Pass |
| 1-7 | 付款金额＜商品价格 | 付款金额=5 商品价格=10 | 提示非法输入 | 提示非法输入 | Pass |
| 1-8 | 输入合法数据 | 付款金额=25  商品价格=12 | 找零：一张10元，1张2元，1张1元 | 找零：一张10元，1张2元，1张1元 | Pass |
| 1-9 | 输入合法数据 | 付款金额=50元  商品价格=40元 | 找零：一张10元 | 找零：一张10元 | Pass |
| 1-10 | 输入合法数据 | 付款金额=40元  商品价格=37元 | 找零：一张2元，一张1元 | 找零：一张2元，一张1元 | Pass |

表3.2 等价类划分测试用例

## 3.2边界值分析法

边界值分析是一种常用的黑盒测试方法，是对等价类划分方法的补充；所谓边界值，是指相对于输入等价类和 输出等价类而言，稍高于其最高值或稍低于最低值的一些特定情况。 边界值分析是通过选取指定数据域的“上点”“内点”“离点”来测试输入或输出的边界。

上点：就是边界上的点，无论域是开区间还是闭区间。若是开区间，上点在域外；若是闭区间，上点就在域内。

离点：是指离“上点”最近得点，这里跟待测数据域是闭区间还是开区间有关系。如果是开区间，那么离点就在域内；如果是闭区间，那么离点就在域外。

内点：域内的任意点都是内点。

### 3.2.1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 | 输入最大金额 | 付款金额=100  商品价格=100 | 找零：0元 | 找零：0元 | Pass |
| 2-2 | 输入最大金额+1 | 付款金额=101 | 提示非法输入 | 提示非法输入 | Pass |
| 2-3 | 输入最小金额 | 付款金额=0  商品价格=0 | 找零：0元 | 找零：0元 | Pass |
| 2-4 | 输入最小金额-1 | 付款金额=-1 | 提示非法输入 | 提示非法输入 | Pass |
| 2-5 | 找零0元 | 付款金额=55  商品价格=55 | 找零：0元 | 找零：0元 | Pass |
| 2-6 | 找零1元 | 付款金额=15  商品价格=14 | 找零： 1张1元 | 找零： 1张1元 | Pass |
| 2-7 | 找零2元 | 付款金额=25  商品价格=23 | 找零： 1张2元， | 找零： 1张2元， | Pass |
| 2-8 | 找零5元 | 付款金额=35  商品价格=30 | 找零： 2张2元，1张1元 | 找零： 2张2元，1张1元 | Pass |
| 2-9 | 找零9元 | 付款金额=25  商品价格=16 | 找零：1张5元， 2张2元， | 找零：1张5元， 2张2元， | Pass |
| 2-10 | 找零10元 | 付款金额=45  商品价格=35 | 找零：1张10元 | 找零：1张10元 | Pass |
| 2-11 | 找零20元 | 付款金额=45  商品价格=25 | 找零：1张20元 | 找零：1张20元 | Pass |
| 2-12 | 找零49元 | 付款金额=50  商品价格=1 | 找零：2张20元，1张5元，2张2元 | 找零：2张20元，1张5元，2张2元 | Pass |
| 2-13 | 找零50元 | 付款金额=70  商品价格=20 | 找零：1张50元 | 找零：1张50元 | Pass |

表3.3 边界值测试用例设计

## 3.3因果图测试

因果图方法最终生成的是判定表。它适合于检查程序输入条件的各种组合情况。

### 3.3.1测试样例设计

1. 输入先导0
2. 输入超过付款金额的商品价格数
3. 输入0到100之间的整数
4. 非法输入
5. 输入小数

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | | 3-1 | 3-2 | 3-3 | 3-4 | 3-5 | 3-6 | 3-7 |
| 原因 |  | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
|  | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|  | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 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:20.5  商品价格:10 | 付款金额:70  商品价格:80 | 付款金额:050.5  商品价格:30 | 付款金额:040.5  商品价格:60 | 付款金额:70.5  商品价格:80 |
| 预期结果 | | 找零：5元纸币1张 | 提示输入错误 | 提示输入错误 | 提示消费金额超出付款金额 | 提示输入错误 | 提示输入错误 | 提示输入错误 |
| 实际结果 | | 找零：5元纸币1张 | 提示输入错误 | 提示输入错误 | 您的消费金额超出付款金额 | 提示输入错误 | 提示输入错误 | 提示输入错误 |
| 测试状态 | | Pass | Pass | Pass | Pass | Pass | Pass | Pass |

表3.4 因果图测试用例设计

# 测试结果概述

## 4.1 对被测试软件的总体评估

软件应该能够正确处理用户的查询请求，对于非法的输入应该进行判断并进行错误提示。

## 4.2 测试环境的影响

与具体的系统环境无关

## 4.3 改进建议

测试用例覆盖程度有待改进

# 软件评价

## 5.1整体评价

程序能够满足用户需求，操作简单，健壮性良好，对于软件需求基本能够实现，能够处理大部分非法输入。

## 5.2缺点与不足

1. 使用不便，必须使用一些IDE（集成开发环境）中运行，具有一定的使用门槛。
2. 对于输入格式限制较为严格。

## 5.3总结

程序基本实现了计算找零钱最佳组合预设的需求，当用户时候能够返回正确结果，对于大部分非法输入能够正确判断并进行错误提示。 在软件测试的过程中能够完成大部分测试，基本满足需求。

# 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 | 2024.10.30 | 10:00 | 赵建桥 | 华硕无畏pro15 2022  win11，version 22H2 |