自动售货机（VendingMachine）程序

测试计划

版本 <1.3>

**曹金坤、李晗东、程浩、吴正雨**

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 01/05/2018 | 1.1 | 确定初步的测试计划 | 吴正雨、李晗东、程浩、曹金坤 |
| 02/05/2018 | 1.2 | 撰写测试计划 | 程浩 |
| 03/05/2018 | 1.3 | 错误修改 | 曹金坤、李晗东 |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

1.4 项目标识 4

2. 测试需求 5

3. 测试策略 5

3.1 测试类型 5

3.1.1 功能测试 5

3.1.2 安全性和访问控制测试（包含开关机功能） 6

3.1.3 安装测试 6

3.2 工具 7

4. 资源 7

4.1 系统 7

5. 项目里程碑 7

6. 可交付工件 8

6.1 测试模型 8

6.2 测试日志 8

6.3 测试报告 8

6.4 测试评估报告 8

测试计划

# 简介

## 目的

编写测试用例报告，对各个功能进行用例设计、测试代码撰写和分类测试的执行，测试软件功能的实现是否符合设计要求、软件的安全性和可用性是否满足设计目标的需要、软件的健壮性是否可以支持在异常情况下的故障处理、软件的易用性是否支持功能程序的快速部署和执行。在这些测试的过程中，发现软件开发中存在的各种问题，形成有数据可以参考和依赖的分析结果，以便在之后对于程序的功能性和非功能性的缺陷进行改正乃至于进一步的程序优化。

## 背景

本测试用例报告主要针对“自动售货机”（VendingMachine）项目第二次迭代完成以后，对该系统的开关机，投币售卖商品以及退币等功能的测试用例设计。在之前使用的程序基础上，添加了大量的功能函数实践和基于Junit的单元测试程序。因为之前的测试是完全基于程序输入和输出进行的“黑盒测试”，所以不涉及测试代码的撰写，但此次的“白盒测试”则把相关的测试用例和逻辑体现在和测试代码中，相关的测试需求和测试代码的对应关系会在测试报告中进行说明。

不同于之前测试的自助饮料售卖机程序，此次的售货机程序功能细节发生了一些变化，其主要的功能如下：

* 自动售货机接受10元，20元，50元与100元硬币，也可根据交易的需要找回上述面值的硬币；
* 测试中售货机贩售三种商品，分别是售价为60的A，100的B，170的C；
* 程序提供开关机的功能实现；
* 投币购买商品后，在必要时的找零，退币的功能。

根据上述的各个程序的功能点，小组成员进行了测试用例设计、单元测试代码编写、测试计划撰写和测试用例报告撰写等工作。

## 范围

本文档适用于“自动售货机”项目第二次迭代开发成果的测试工作。

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 | 程序雏形作者的博客 |  |
| 功能性规约 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 | 小组对于程序的分析 |  |
| 用例报告 | ☑ 是 🞏 否 | 🞏 是 ☑ 否 | 小组撰写 |  |
| 项目计划 | ☑ 是 🞏 否 | 🞏 是 ☑ 否 | 小组撰写 |  |
| 设计规约 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |
| 原型 | ☑ 是 🞏 否 | 🞏 是 ☑ 否 | 小组撰写 | 由小组撰写的条件表进行细化写出，该条件表格部分内容可见附件部分。 |
| 用户手册 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |
| 业务模型或业务流程 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |
| 数据模型或数据流 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |
| 业务功能和业务规则 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |
| 项目或业务风险评估 | 🞏 是 ☑ 否 | 🞏 是 ☑ 否 |  |  |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

2.1 **功能性需求：**

* 客户操作自动售货程序进行投币、购买和找零退款的功能；
* 客户对自动售货机的开启与关机功能；

2.2 **非功能性需求**：

* 当库存和余额数目被修改为非有效数字（如负数）时进行提示和操作禁止的数据安全性需求；
* 多次连续购买，后台库存和余额数据的稳定性需求；
* 错误操作后的程序功能恢复和相关警告和报错功能。

# 测试策略

## 测试类型

### 功能测试

|  |  |
| --- | --- |
| 测试目标： | 在程序的操作逻辑中，用户可以选择投入10元，20元，50元与100元硬币，之后可以选择一款有库存的商品，和我们的日常经验相符合，只有当投入的硬币足够支付该商品并且售货机中有足够的硬币来进行找零，用户才可以成功的购买对应的商品。售货机会识别投入硬币的种类。 |
| 方法： | 设置一定情景，对单个功能函数进行测试，预期获得以下效果：  1. 系统允许的功能性操作得到预期的操作结果；  2. 不满足系统要求的功能性操作被禁止，并返回对应的错误信息； 3. 在错误操作后，系统当下的正确状态不发生修改。 |
| 完成标准： | 1. 所有计划的测试用例均已完成测试；  2. 所有有效用例均获得对应的预期结果；  3. 无效用例返回了相应的错误信息；  4. 暴露的缺陷已经被修改。 |
| 需考虑的特殊事项： | 1. 错误的操作在没有被禁止之前，可能已经修改了程序的相关状态，从而导致错误；  2.正确的操作，在进行极端条件的尝试后，可能会导致常规用例无法测试到的错误类型。 |

### 安全性和访问控制测试（包含开关机功能）

|  |  |
| --- | --- |
| 测试目标： | * 确保关机状态下，所有对售货机的操作（开启售货机除外）都应该不生效。具体表现为：不响应关机状态下所执行的操作；关机状态下投入的硬币应该被退出来；在关闭售货机时，其中的商品和钱应该被退出来；在关机状态下执行开机操作时可以正确执行。 |
| 方法： | * 1. 在机器关闭的状态下，对售货机的各个功能进行单独测试，检查程序的响应情况。 * 2.在机器关闭的状态下，执行开机操作，检查程序的响应情况。 |
| 完成标准： | 确认关机状态下不可执行除了开机以外的其他操作。关机状态下的非开机操作被禁止并且返回对应的提示信息、开机的操作运行正确并且进行反馈。 |
| 需考虑的特殊事项： | 无 |

### 安装测试

|  |  |
| --- | --- |
| 测试目标： | 核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中：  1. 首次安装，以前没有安装过Java的新计算机；  2.更新，以前安装过编写和测试程序所用相同版本Java(Java 1.8.0\_151)的计算机；  3.更新，以前安装过较早版本Java的计算机。 |
| 方法： | 1. 对于没有安装过Java的计算机，从oralce下载合适版本的Java进行安装，之后部署该应用程序，测试运行；  2. 对于已经安装了某版本Java的计算机，直接部署该程序，尝试运行和相关功能操作。 |
| 完成标准： | 程序在计算机上正确运行，并且正确处理所有用户逻辑和事务、可以成功运行基于Junit的测试程序组件。 |
| 需考虑的特殊事项： | 原先程序由maven进行管理，并依赖maven集成了junit组件，在测试计算机上如果不进行maven的部署，测试是否会引起程序错误和缺陷。 |

## 工具

此项目将使用以下工具：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 测试管理 | Maven + Junit | The Apache Software Foundation、Kent Beck & Erich Gamma | Maven 2.0 & Junit 4 |
| 缺陷跟踪 | IDEA + Junit | Jetbrain、Kent Beck & Erich Gamma | IDEA 2017.3.4 & Junit 4 |
| 测试覆盖监测器或评价器 | Junit | Kent Beck & Erich Gamma | Junit 4 |
| 项目管理 | Maven + IDEA | The Apache Software Foundation、Jetbrain | Maven 2.0 & IDEA 2017.3.4 |

# 资源

## 系统

下表列出了测试项目所需的系统资源。

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 客户端测试 PC | MacOS 10.12.6、Windows 10 |
| 测试开发 PC | MacOS 10.12.6、Windows 10 |

# 项目里程碑

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 12人·时 | 01/05/2018 | 01/05/2018 |
| 设计测试 | 10人·时 | 01/05/2018 | 01/05/2018 |
| 实施测试 | 6人·时 | 02/05/2018 | 02/05/2018 |
| 执行测试 | 10人·时 | 02/05/2018 | 02/05/2018 |
| 评估测试 | 10人·时 | 03/05/2018 | 03/05/2018 |

# 

# 可交付工件

## 测试模型

目标程序由maven组织的java文件写成，对应的测试自动化程序由Junit模块组成，该测试计划采用了“白盒测试”，可以利用Junit对测试覆盖率进行查看。此外，更直观的测试需求则由基于用户界面的互动测试完成。

因为Junit的特性，可以对于功能程序中的语句和不同功能函数和分支模块的测试效果进行评估，方便之后的分析报告和对于测试的优化。

## 测试日志

1. Junit内置的工具给出了测试代码的运行情况报告和对应的日志信息，内容包括了测试代码对于功能代码的覆盖率和正确性等必要的信息。
2. IDEA的运行日志记录了功能函数运行过程中的相关内部变量和可观察的程序反馈情况，作为了测试报告的补充，为评估测试的细节提供帮助。

## 测试报告

基于前述的测试计划要求，对应的满足基于需求覆盖的测试用例被规范地设计，相关的测试代码被撰写并且在Junit下组织成为不同的测试部件，独立的对于不同的程序功能和模块进行测试，相关的细节也在测试报告中被体现。此外，对于测试用例的解释也被提供，作为对整个测试工作的最基础的说明文档和细节说明。

## 测试评估报告

在测试完成之后，对测试过程和结果的评价会被分析进而生成测试的评估报告。

测试报告主要分为以下部分进行组织：

1. 测试结果的概述，包括测试的覆盖率和测试的正确性等；
2. 不同功能部分的测试结果比对，细致分析不同功能逻辑和测试函数的运行情况；
3. 对测试结果的总结和对进一步优化的分析。