简易Java计算器

测试计划

版本 <1.0>

[注：以下提供的模板用于 Rational Unified Process。其中包括用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <12日/4月/2019年> | <1.0> | <编写了针对lab1的黑盒测试文档> | <汪喆昊> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 **错误!未定义书签。**

2. 测试需求 4

3. 测试策略 4

3.1 测试类型 4

3.1.1 数据和数据库完整性测试 4

3.1.2 功能测试 4

3.1.3 业务周期测试 4

3.1.4 用户界面测试 4

3.1.5 性能评价 4

3.1.6 负载测试 4

3.1.7 强度测试 **错误!未定义书签。**

3.1.8 容量测试 **错误!未定义书签。**

3.1.9 安全性和访问控制测试 **错误!未定义书签。**

3.1.10 故障转移和恢复测试 **错误!未定义书签。**

3.1.11 配置测试 **错误!未定义书签。**

3.1.12 安装测试 **错误!未定义书签。**

3.2 工具 4

4. 资源 4

4.1 角色 4

4.2 系统 4

5. 项目里程碑 4

6. 可交付工件 4

6.1 测试模型 4

6.2 测试日志 4

6.3 缺陷报告 4

7. 附录 A：项目任务 4

测试计划

# 简介

## 目的

简易Java计算器 的这一“测试计划”文档有助于实现以下目标：

• [确定现有项目的信息和应测试的软件构件。

• 列出推荐的测试需求（高层次）。

• 推荐可采用的测试策略，并对这些策略加以说明。

• 确定所需的资源，并对测试的工作量进行估计。

• 列出测试项目的可交付元素]

## 背景

[输入测试对象（组件、应用程序、系统等）及其目标的的简要说明。需要包括的信息有：主要的功能和特性、测试对象的构架以及项目的简史。本节应该只包含 3 至 5 个段落。]

## 范围

[描述测试的各个阶段，例如：单元测试、集成测试或系统测试，并说明本计划所针对的测试类型（如功能测试或性能测试）。简要地列出测试对象中将接受测试或将不接受测试的那些特性和功能。

如果在编写此文档的过程中作出的某些假设可能会影响测试设计、开发或实施，则列出所有这些假设。

列出可能会影响测试设计、开发或实施的所有风险或意外事件。

列出可能会影响测试设计、开发或实施的所有约束。]

## 

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：

[注：可以视情况删除或添加项目。]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |  |  |
| 功能性规约 | 🗹 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 用例报告 | 🗹 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 项目计划 | 🗹 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 设计规约 | 🗹 是 🞏 否 | 🞏 是 🞏 否 |  |  |
| 原型 | 🞏 是 🗹 否 | 🞏 是 🞏 否 |  |  |
| 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。

[在此处输入一个主要测试需求的高层次列表。]

# 测试策略

[测试策略提供了推荐用于测试对象的方法。上一节“测试需求”中说明了将要测试哪些对象，而本节则要说明如何对测试对象进行测试。

对于每种测试，都应提供测试说明，并解释其实施和执行的原因。

如果不实施和执行某种测试，则应该用一句话加以说明，并陈述这样做的理由。例如，“将不实施和执行该测试。。该测试不合适。”

制定测试策略时所考虑的主要事项有：将要使用的方法以及判断测试何时完成的标准。

下面列出了在进行每项测试时需考虑的事项，除此之外，测试还只应在安全的环境中使用已知的、受控的数据库来执行。 ]

## 测试类型

### 数据和数据库完整性测试

[数据库和数据库进程应作为<项目**名称**>中的子系统来进行测试。

在测试这些子系统时，不应将测试对象的用户界面用作数据的接口。对于数据库管理系统 (DBMS)，还需要进行深入的研究，以确定可以支持以下测试的工具和方法。]

|  |  |
| --- | --- |
| 测试目标： | [确保数据库访问方法和进程正常运行，数据不会遭到损坏。] |
| 方法： | • [调用各个数据库访问方法和进程，并在其中填充有效的和无效 的数据或对数据的请求。  • 检查数据库，确保数据已按预期的方式填充，并且所有 数据库事件都按正常方式出现；或者检查所返回的数据，确保为 正当的理由检索到了正确的数据] |
| 完成标准： | [所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。] |
| 需考虑的特殊事项： | • [测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接 输入或修改数据。  • 进程应该以手工方式调用。  • 应使用小型或最小的数据库（其中的记录数很有限）来 使所有无法接受的事件具有更大的可见性。] |

### 功能测试

[测试对象的功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面 (GUI) 与应用程序交互并分析输出结果来验证应用程序及其内部进程。以下列出的是每个应用程序推荐的测试方法概要：]

|  |  |
| --- | --- |
| 测试目标： | [确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。] |
| 方法： | [利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。] |
| 完成标准： | • [所计划的测试已全部执行。  • 所发现的缺陷已全部解决。] |
| 需考虑的特殊事项： | [确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）] |

### 业务周期测试

[业务周期测试应模拟在一段时间内对 <项目名称> 执行的活动。应先确定一段时间（例如一年），然后执行将在该时段内发生的事务和活动。这种测试包括所有的每日、每周和每月的周期，以及所有与日期相关的事件（如备忘录）。]

|  |  |
| --- | --- |
| 测试目标 | [确保测试对象及后台进程都按照所要求的业务模型和时间表正确运行。] |
| 方法： | [通过执行以下活动，测试将模拟若干个业务周期：  • 将修改或增强对测试对象进行的功能测试，以增加每项功能的执 行次数，从而在指定的时段内模拟若干个不同的用户。  • 将使用有效的和无效的日期或时段来执行所有与时间或日期相关 的功能。  • 将在适当的时候执行或启动所有周期性出现的功能。  • 在测试中还将使用有效的和无效的数据，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。 |
| 完成标准： | • [所计划的测试已全部执行。  • 所发现的缺陷已全部解决。} |
| 需考虑的特殊事项： | • [系统日期和事件可能需要特殊的支持活动  • 需要通过业务模型来确定相应的测试需求和测试过程。] |

### 用户界面测试

[通过用户界面 (UI) 测试来核实用户与软件的交互。UI 测试的目标在于确保用户界面向用户提供了适当的访问和浏览测试对象功能的操作。除此之外，UI 测试还要确保 UI 功能内部的对象符合预期要求，并遵循公司或行业的标准。]

|  |  |
| --- | --- |
| 测试目标： | [核实以下内容：  • 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括 窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法 （Tab 健、鼠标移动和快捷键）的使用  • 窗口的对象和特征（例如：菜单、大小、位置、状态和 中心）都符合标准。] |
| 方法： | [为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。] |
| 完成标准： | [证实各个窗口都与基准版本保持一致，或符合可接受标准] |
| 需考虑的特殊事项： | [并不是所有定制或第三方对象的特征都可访问。] |

### 性能评价

[性能评价是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。性能评价的目标是核实性能需求是否都已满足。实施和执行性能评价的目的是将测试对象的性能行为当作条件（例如工作量或硬件配置）的一种函数来进行评价和微调。

注：以下事务均指“逻辑业务事务”。这种事务被定义为将由系统的某个主角通过使用测试对象来执行的特定用例，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 测试目标： | [核实所指定的事务或业务功能在以下情况下的性能行为：  • 正常的预期工作量  • 预期的最繁重工作量] |
| 方法： | • [使用为功能或业务周期测试制定的测试过程。  • 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项 事务的迭代次数。  • 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为 基准），并在多台客户机（虚拟的或实际的客户机，请参见下面 的“需考虑的特殊事项”）上重复。] |
| 完成标准： | • [单个事务或单个用户：在每个事务所预期或要求的时间范围内 成功地完成测试脚本，没有发生任何故障。]  • [多个事务或多个用户：在可接受的时间范围内成功地完成测试 脚本，没有发生任何故障。] |
| 需考虑的特殊事项： | [综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：  • 直接将“事务强行分配到”服务器上，这通常以“结构化查询语 言”(SQL) 调用的形式来实现。  • 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客 户机。 此负载可通过“远程终端仿真”(Remote Terminal Emulation) 工具来实现。 此技术还可用于在网络中加载“流 量”。  • 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添 加负载。  性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。] |

### 负载测试

[负载测试是一种性能测试。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征，例如，响应时间、事务处理速率和其他与时间相关的方面。]

[注：以下事务均指“逻辑业务事务”。这些事务被定义为将由系统的最终用户通过使用应用程序来执行的具体功能，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 测试目标： | [核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。] |
| 方法： | • [使用为功能或业务周期测试制定的测试。  • 通过修改数据文件来增加事务数量，或通过修改测试来增加每项 事务发生的次数。] |
| 完成标准： | [多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。] |
| 需考虑的特殊事项： | • [负载测试应该在专用的计算机上或在专用的机时内执行，以便 实现完全的控制和精确的评测。  • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数 据库。] |

### 工具

此项目将使用以下工具：

[注：可以视情况删除或添加项目。]

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 测试管理 |  |  |  |
| 缺陷跟踪 |  |  |  |
| 用于功能性测试的 ASQ 工具 |  |  |  |
| 用于性能测试的 ASQ 工具 |  |  |  |
| 测试覆盖监测器或评价器 |  |  |  |
| 项目管理 |  |  |  |
| DBMS 工具 |  |  |  |

# 资源

## 角色

下表列出了在此项目的人员配备方面所作的各种假定。

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理，  测试项目经理 | 1 | 进行管理监督。  职责：   * 提供技术指导 * 获取适当的资源 * 提供管理报告 |
| 测试设计员 | 1 | 确定测试用例、确定测试用例的优先级并实施测试用例。  职责：   * 生成测试计划 * 生成测试模型 * 评估测试工作的有效性 |
| 测试员 | 1 | 执行测试。  职责：   * 执行测试 * 记录结果 * 从错误中恢复 * 记录变更请求 |
| 测试系统管理员 | 1 | 确保测试环境和资产得到管理和维护。  职责：   * 管理测试系统 * 授予和管理角色对测试系统的访问权 |
| 数据库管理员 | 0 | 确保测试数据（数据库）环境和资产得到管理和维护。  职责：   * 管理测试数据（数据库） |
| 设计员 | 1 | 确定并定义测试类的操作、属性和关联。  职责：   * 确定并定义测试类 * 确定并定义测试包 |
| 实施员 | 1 | 实施测试类和测试包，并对它们进行单元测试。  职责：   * 创建在测试模型中实施的测试类和测试包 |

## 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 |
| 数据库服务器 | / |
| —网络或子网 | / |
| —服务器名服务器名 | / |
| —数据库名 | / |
| 客户端测试 PC | / |
| —包括特殊的配置需求 | / |
| 测试存储库 | / |
| —网络或子网 | / |
| —服务器名服务器名 | / |
| 测试开发 PC | / |

# 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 | 1 | 2019/4/12 | 2019/4/12 |
| 设计测试 | 2 | 2019/4/12 | 2019/4/12 |
| 实施测试 | 5 | 2019/4/12 | 2019/4/12 |
| 执行测试 | 5 | 2019/4/12 | 2019/4/12 |
| 评估测试 | 5 | 2019/4/12 | 2019/4/12 |

# 可交付工件

[本节列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。]

## 测试模型

[本节确定将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用 ASQ 工具来创建或引用。]

## 测试日志

[说明用来记录和报告测试结果和测试状态的方法和工具。]

## 缺陷报告

[本节确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。]

# 附录 A：项目任务

以下是一些与测试有关的任务：

• 制定测试计划

* + 确定测试需求
  + 评估风险
  + 制定测试策略
  + 确定测试资源
  + 创建时间表
  + 生成测试计划

• 设计测试

- 准备工作量分析文档

- 确定并说明测试用例

- 确定并结构化测试过程

- 复审和评估测试覆盖

• 实施测试

* + 记录或通过编程创建测试脚本
  + 确定设计与实施模型中的测试专用功能
  + 建立外部数据集

• 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

• 评估测试

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准