

软件测试实训项目测试计划

计划人：赵洋

最后更新时间：2018/08/23

目录

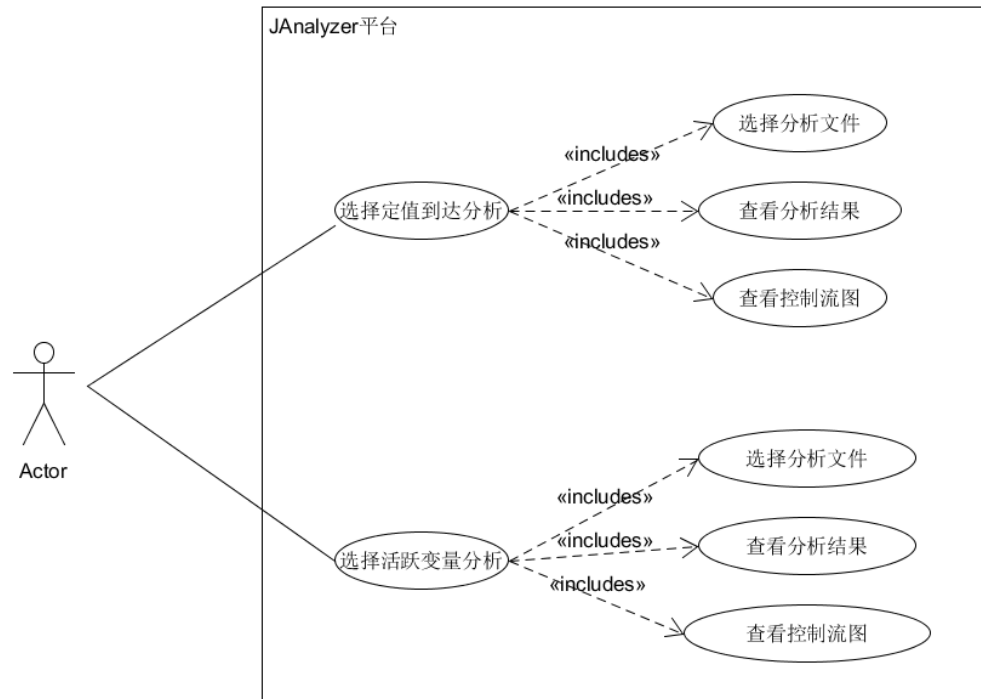
引言.....	2
目的	2
背景	2
缩略语.....	3
参考文献	3
测试策略.....	3
测试计划.....	5
功能	5
性能	7
时间点统计与反馈总结	8

一、引言

a) 目的

对我们设计的结果进行测试分析

1.设计多种测试用例来对我们以下几个结果进行分析



2.同时对生成的 GUI 界面进行分析测试

对界面美观与按键测试分析

b) 背景

JAnalyzer 有完成抽象语法树生成、名字表生成和软件结构提取这三个基础构件。这三个功能是对源代码做静态分析的基础，任何对软件源代码的理解和分析都需要在生成抽象语法树和名字表的基础上，基于软件的结构完成。

所以需要对我们所做的软件，在 AST, Variable Table 以及 Software Structure 所提取结果检测。对定值和活跃变量结果对比分析，同时对可

视化界面的结果进行测试性能进行分析以及外观用户使用进行测试。

c) 缩略语

简单来说，通过设计多个 JAVA 测试代码用例进行分析同时对 GUI 的界面进行分析。

d) 参考文献

[1]陆卫东,金成植.基于活跃变量分析的流图语言的部分求值器[J].软件学报,1997(01):30-36.

[2] JAnalyzer 基础构建开发文档

[3]网上相关博客和论坛

[4]维基百科

二、测试策略

a) 功能

测试技术	是否采用	说明
里程碑技术	是	在不同阶段的里程碑采用不同的技术方法以及验收方法进行测试并在完成后制定。
编写测试用例	是	在产品编码前进行测试用例编写
单元测试技术	是	通过 JUNIT 对于我们写的每一个代码单元进行测试

主要对我们生成的到达定制分析和动态变量结果进行分析

一个循环测试

主要采用的策略是：TDD（测试驱动开发(Test-Driven Development)）

TDD 的基本思路就是通过测试来推动整个开发的进行，但测试驱动开发并不只是单纯的测试工作，而是把需求分析，设计，质量控制量化的过程。

主要在我们此次代码中，我们使用了

通过例子编写出测试代码再通过测试代码完成算法代码，再反过来相互测试。

达到了以下几个标准

独立测试：不同代码的测试应该相互独立，一个类对应一个测试类（对于 C 代码或 C++全局函数，则一个文件对应一个测试文件），一个函数对应一个测试函数。用例也应各自独立，每个用例不能使用其他用例的结果数据，结果也不能依赖于用例执行顺序。一个角色：开发过程包含多种工作，如：编写测试代码、编写产品代码、代码重构等。做不同的工作时，应专注于当前的角色，不要过多考虑其他方面的细节。

测试列表：代码的功能点可能很多，并且需求可能是陆续出现的，任何阶段想添加功能时，应把相关功能点加到测试列表中，然后才能继续手头工作，避免疏漏。

测试驱动：即利用测试来驱动开发，是 TDD 的核心。要实现某个功能，要编写某个类或某个函数，应首先编写测试代码，明确这个类、这个函数如何使用，如何测试，然后在对其进行设计、编码。

先写断言：编写测试代码时，应该首先编写判断代码功能的断言语

句，然后编写必要的辅助语句。

可测试性：产品代码设计、开发时的应尽可能提高可测试性。每个代码单元的功能应该比较单纯，“各家自扫门前雪”，每个类、每个函数应该只做它该做的事，不要弄成大杂烩。尤其是增加新功能时，不要为了图一时之便，随便在原有代码中添加功能，对于 C++ 编程，应多考虑使用子类、继承、重载等 OO 方法。

及时重构：对结构不合理，重复等“味道”不好的代码，在测试通过后，应及时进行重构。

小步前进：软件开发是复杂性非常高的工作，小步前进是降低复杂性的好办法。

b) 性能

在性能方面我们在时间上进行了规划和计划

阶段	标准说明
开始测试	进入算法系统
中断测试	无法正确输出或者无法输出
完成测试	完成规划并达到质量停止时间

上述可以测试时间并对我们性能分析

三、测试计划

我们主要采用了 Junit 和 GUIDancer 两个测试工具进行测试

JUnit 是一个 Java 语言的单元测试框架。它由 Kent Beck 和 Erich Gamma 建立，逐渐成为源于 Kent Beck 的 sUnit 的 xUnit 家族中最为成功的一个。

JUnit 有它自己的 JUnit 扩展生态圈。多数 Java 的开发环境都已经集成了

JUnit 作为单元测试的工具。

JUnit 是由 Erich Gamma 和 Kent Beck 编写的一个回归测试框架 (regression testing framework)。JUnit 测试是程序员测试，即所谓白盒测试，因为程序员知道被测试的软件如何 (How) 完成功能和完成什么样 (What) 的功能。JUnit 是一套框架，继承 TestCase 类，就可以用 JUnit 进行自动测试了。

而 GUIDANCERr 可以作为 Eclipse 插件运行或者独立运行。测试用例现在自动包含在项目中，允许测试人员跨项目重用关键字并使用拖放快速创建它们。现在可以更容易地从可执行文件开始测试中的应用程序。其他可用性的改进包括“增加欢迎页面、备忘单和新的示例项目，”

下面是我们基本的一个测试人员分化与表格

人员	测试负责项目
赵洋	制定测试计划与 TDD 测试策略, 衡量对比分析测试工具 Junit 和 GUIDancer。和组员编码探讨测试用例并分析测试结果，给予开发反馈。
张毅超	编写测试用例，分析基本 Junit 单元，对到达定值和动态变量的结果进行分析
张书博	编写 TDD 初始测试计划, 和赵洋探讨基本的到达定值和动态变量策

	略。
张睿	循环反馈 TDD 策略通过测试结果，进行开发算法测试。
杨茂林	对 TDD 的测试编写算法，同时对 GUI 反馈结果进行调整
邓清方	协助各个部分完成代码工作。循环验收代码重构部分。

四、时间点统计与反馈总结

2018 年 7 月 23 日 基本完成前驱测试用例

2018 年 7 月 29 日 使用测试用例驱动代码重构

2018 年 8 月 10 日 完成循环 15 分测试代码用例 完成后驱测试

2018 年 8 月 12 日 完成 GUIDancer 的 GUI 测试

2018 年 8 月 15 日 完成 Junit 单元测试

2018 年 8 月 20 日 完成对测试结果反馈以及最后一次调整

2018 年 8 月 22 日 撰写测试报告并对测试计划的结果进行分析总结