

硕士论文开题报告

论文名称：JavaCard调试系统的设计与实现

姓 名：袁芳

学 号：GS132A554

专 业：软件质量管理与测试专业

所属院系：软件学院

学院指导老师：路新喜

企业指导教师：董成根

企业指导老师单位：金联汇通信息技术有限公司

论文起止时间：2016 年7月1日至2017年7月1日

**2016** 年 7月

目录

[1. 课题来源及意义 1](#_Toc451936647)

[1.1. 课题来源 1](#_Toc451936648)

[1.2. 课题意义 1](#_Toc451936649)

[2. 国内外现状分析 1](#_Toc451936650)

[2.1. 国外研究现状 1](#_Toc451936651)

[2.2. 国内研究现状 1](#_Toc451936652)

[2.3. 对比分析 2](#_Toc451936653)

[3. 论文研究目标和内容 2](#_Toc451936654)

[3.1. 研究目标 2](#_Toc451936655)

[3.2. 研究内容 2](#_Toc451936656)

[4. 需求分析 3](#_Toc451936657)

[4.1. 需求概述 3](#_Toc451936658)

[4.2. 业务需求梳理 3](#_Toc451936659)

[4.3. 功能需求分析 5](#_Toc451936660)

[4.3.1. XXX功能 5](#_Toc451936661)

[4.3.2. XXX功能 6](#_Toc451936662)

[4.3.3. XXX功能 7](#_Toc451936663)

[4.4. 系统功能结构图 7](#_Toc451936664)

[5. 关键问题和解决方案 8](#_Toc451936665)

[5.1. 系统解决方案 8](#_Toc451936666)

[5.1.1. 系统架构 8](#_Toc451936667)

[5.1.2. 技术选型 10](#_Toc451936668)

[5.1.3. 设计原则 11](#_Toc451936669)

[5.2. 关键问题及解决思路（可以没有） 12](#_Toc451936670)

[6. 预期的成果 13](#_Toc451936671)

[6.1. 研究成果 13](#_Toc451936672)

[6.2. 技术指标 14](#_Toc451936673)

[7. 论文工作计划 14](#_Toc451936674)

[8. 主要参考文献 15](#_Toc451936675)

# 课题来源及意义

## 课题来源

该项目来源于金联汇通信息技术有限公司JavaCard调试系统项目。研究目的在于设计一个JavaCard调试系统，解决Java源码级调试的问题，从而加速开发人员对Java卡操作系统和Applet的开发与调试。

JavaCard是在传统智能卡的基础上发展而来，它解决了传统智能卡的缺点和不足。近年来，Java Card发展迅猛，已经广泛的渗透到了人们生活的方方面面。Java Card广泛应用在银行卡、手机卡、公交卡、社保卡、ETC卡等各个领域，市场对Java Card要求越来越细致深入，开发也越来越复杂。市场竞争日趋激烈，致使对Java Card系统开发周期要求越来越短，因此，开发一套支撑Java Card系统高效开发的调试工具就显得至关重要。

Java开发与调试已经具备了比较成熟的平台调试体系规范——JPDA（Java Platform Debugger Architecture），而JavaCard开发与调试应用专业化，目前的Java卡平台规范中没有包含对JavaCard调试体系的描述，由于部分现有工具未开源等因素的制约，还没有能够完全满足Java卡操作系统和Applet的开发与调试工具。因此JavaCard调试系统的研究和实现具有一定的现实意义和难度

## 课题意义

本论文是基于已经成熟的Java平台调试体系规范JPDA（Java Platform Debugger Architecture），分析目前JavaCard调试的特殊性，结合自身的需求，设计实现一个JavaCard调试系统，从而完成对JavaCard操作系统和Applet的开发与调试，提高开发人员的工作效率，同时降低对开发人员的业务水平要求，这将会大大提高企业的市场竞争力。

# 国内外现状分析

基于Java调试已经有了（Oracle规范）详细的规范，但基于Java Card调试还未形成行业规范。由于Java Card竞争非常激烈，各大卡制造商或服务商都暗自研究设计自己的Java Card开发调试软件。

## 国外研究现状

Java调试器研究目前主要有Java语言的主要开发和维护公司-甲骨文（Oracle）公司所主导。目前甲骨文（Oracle）公司制定的Java平台调试体系规范—JPDA（Java Platform Debugger Architecture）已随着其JDK（Java Development Kit）的广泛应用而成为业界标准。

Sun公司的JDB（Java调试器）是基于文本和命令行的调试工具，虽然JDB可以免费获取并且是平台独立，但开发人员认为它很原始，难以使用。

针对JDB 的人机界面的改善方向，出现了一批独立的GUI Java 调试器，例如开源社区贡献的JDebugTool 工具和JSwat 工具。上述GUI 的开源Java 调试器都基于甲骨文(Oracle) 公司制订的JPDA 架构。国外的Java 厂商则主要致力于集成的Java 开发环境的设计和推广，如IBM 公司的Visual Age for Java, Symantec 公司的Visual Cafe 以及Borland 公司的JBuilder。

在跨平台开源集成开发环境(Java IDE)领域最成熟和使用最广泛的是开源社区的Eclipse。Eclipse 以插件的机制来实现其灵活的扩展性，开发人员可以编写插件来实现自己所需的额外功能 [3] 。Eclipse JDT 调试器可与任何支持JDPA 调试协议的Java 虚拟机连接。

IBM公司针对自己生产的JCOP（Java Card OpenPlatform）卡，是 IBM研究院为JavaCard平台开发的智能卡操作系统；JCOP卡中拥有能够运行Java应用程序的Java卡虚拟机，JCOP Shell是IBM公司专门针对JCOP卡而开发的调试工具,但只能对Applet调试。Applet调试可以加速应用的开发，能够快速定位问题，但是，对于Java Card操作系统的开发调试，还是无法实现。

## 国内研究现状

国内对于JavaCard调试器的研究主要集中在基于Java标准JPDA调试体系的特定应用的系统研究和实现。由于JavaCard竞争非常激烈，各大卡制造商或服务商都研究设计自己的JavaCard开发调试软件。主要有：

#### 通过字节码进行调试

通过自节码进行调试，此方法定位问题困难，对开发人员技术水平要求很高；

#### 基于Applet的调试

Java Card技术不断发展经过不断的发展，基于Applet的调试已经比较成熟，很多公司使用基于Applet的调试。Applet调试可以加速应用的开发，能够快速定位问题，比通过自节码进行调试的方法有了很大的提高，但是，对于Java Card操作系统的开发调试，还是无法实现。主要有《Java卡调试平台的研究与实现》，此种方法可以进行Applet的调试，方便Applet开发人员进行问题定位，提高Applet的开发效率，但是它不适合Java Card操作系统的开发调试[2]。

#### 基于Java Card操作系统的调试

系统开发人员可以在Java Card系统实现的方法内添加断点、查看系统对象的内容，还可以进行单步跟踪，全速运行。也可以在Applet的方法内添加断点、查看Applet对象内容。可以查看到JCVM（Java Card Virtual Machine）启动后完整的调用栈信息。帮助系统开发人员、Applet开发人员快速的定位问题。

## 对比分析

各种调试器的分析，参考《openfile》

与国外发达国家相比，我国XXXX信息化起步相对较晚，无论是XXXX存在着不少的差距。

表1 对比分析表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 字节码调试 | Applet调试 | JDB | JCOP Shell |  |
|  |  |  |  |  |  |

# 论文研究目标和内容

## 研究目标

1. 总体目标

本研究总体目标是设计并实现一个使用Eclipse做为调试器，JavaCard模拟器做为调试目标的JavaCard调试系统，解决JavaCard的Java源码级调试问题。

1. 具体目标

该系统的核心问题包括四点：

1. 实现与调试器的JDWP协议交互；
2. 实现JCRE的Java源码级调试；
3. 实现JavaCard应用的Java源码级调试；
4. 调试组件的动态管理。

## 研究内容

JavaCard调试系统的目的是为了解决Java源码级调试的问题，从而加速开发人员对JavaCard操作系统和Applet的开发与调试。为实现本课题，研究的内容主要包括：

首先，研究已成熟的Java平台调试框架——JPDA（Java Platform Debugger Architecture），JPDA定义了一套如何开发调试工具的接口和规范。JPDA由三个独立的模块组成：JVMTI（Java Virtual Machine Profiler Interface）、JDWP（Java Debugger Wire Protocal）、JDI（Java Debug Interface），调试者通过JDI发送接受调试命令，JDWP定义调试者和被调试者交流数据的格式，JVMTI可以控制当前虚拟机运行状态，~~通过分析发现，可以基于JVMTI开发特定的JavaCard调试工具，实现Java源码级调试。~~

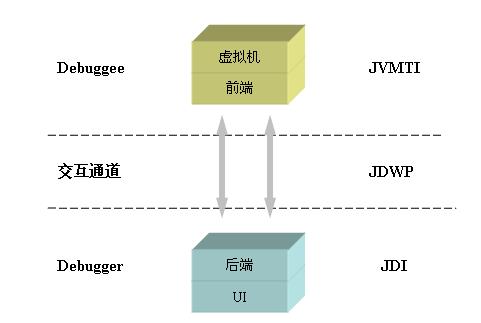


图1 JPDAP模块层次图

其次，Eclipse调试框架研究，包含Eclipse启动框架和Eclipse调试模型研究。Eclipse的JDT插件已经完整的实现了JDI接口。

再次，远程调试协议接口研究，包含JDWP通信协议。JDWP（Java Debug Wire Protocol）是一个通讯交互协议，它定义了调试器和被调试程序之间传递的信息的格式。 详细完整地定义了请求命令、响应数据和错误代码。

另外，分析Java编译文件class与JavaCard编译文件cap的的共同点与不同点，发现：JavaCard不支持多线程、不支持字符串，是信息查找是基于token的查找。Class文件是基于字符串的查找。由于它们的信息查找方式不同，如果要实现Java源码级调试，需要对JDWP协议进行剪裁。

# 需求分析

## 需求概述

本章对所要实现的JavaCard调试系统进行系统级的需求分析。包括业务需求梳理、功能需求分析、非功能需求分析、系统功能结构图四个方面。

## 业务需求梳理



#### ~~建立通信连接~~

~~应该删除，不是业务需求，只是实现手段~~

建立连接

启动Eclipse调试界面，建立Eclipse和调试器的连接。

#### 设置断点

断点的添加、删除和断点事件判断。

是否可以进行描述，最好添加流程图？

图1添加断点事件业务流程图

图2断点事件判断业务流程图

#### 变量查看及修改

通4.2.2

#### 单步运行

同4.2.2

#### 全速运行

同4.2.2

#### 调用栈查看

同4.2.2

#### 单步事件的添加、删除及步入、步出和步过事件判断

同4.2.2

## 功能需求分析

#### 断点管理功能

断点管理功能的使用者是开发人员，开发人员需要实现添加断点和删除断点，调试目标一般包括。。。。，当调试目标接收到添加断点和删除断点命令后，需要对断点事件进行判断。



图1 断点管理功能用例图

#### 变量管理功能

实现变量值显示、变量值修改、增加查看变量和删除查看变量。



图2 变量管理功能用例图

#### 单步处理功能



图3 单步处理功能用例图

#### 调试目标状态管理功能



图4 调试目标状态管理功能用例图

#### 方法调用栈显示功能



图5 方法调用栈显示功能用例图











## 系统功能结构图

本课题研究的内容主要包括调试代理和JDWP调试模块，具体内容详见下图12





图12 JavaCard调试系统功能结构图

## 调试器

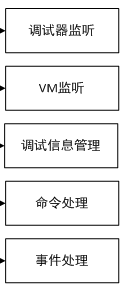
常用的调试器有Eclipse和NetBeans，本项目使用Eclipse作为调试器。调试器本身实现了JDI接口，通过JDWP协议与调试代理交换信息。本次设计不包含调试器设计。

## 调试代理

由于JavaCard运行的平台资源有限，将部分与调试器交互的命令处理由调试代理处理，代理将自己无法处理的JDWP命令转发给JavaCard，并将JavaCard发送的返回包转发给调试器。

调试代理与调试器Eclipse间的连接方式为Socket。调试代理与JavaCard间的连接方式则依赖于JavaCard的实现方式。

调试代理启动时，会创建一个调试器监听器，该监听器会建立一个Socket服务端，等待调试器链接，同时代理也会创建一个VM监听器，用于处理与JavaCard间的数据包。这些数据包包含代理发送给JavaCard的JCDWP命令和JavaCard发送给代理的命令返回包，还有JavaCard主动发送给代理的JCDWP事件命令和代理对事件命令的返回包。



调试器监听：负责。。。

VM监听：负责。。。

调试信息管理：负责。。。

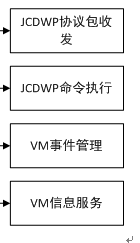
命令处理：负责。。。

事件处理：负责。。。

## JavaCard模拟器

JavaCard将JCDWP模块直接集成到JCVM中。JavaCard启动时，其内部的JCDWP调试模块会建立一个Socket服务端，等待代理连接。当调试代理和JavaCard都启动后，连接建立成功。





JCDWP协议包收发：负责接收由代理发送的JCDWP命令包和命令包的响应包，发送由JCVM触发的事件命令。

JCDWP命令执行：负责解析由调试代理发送的JCDWP命令，并返回命令对应的响应数据。

VM事件管理：记录由代理发送的事件请求，并检查事件是否触发，如果触发则发送事件命令给调试代理。

VM信息服务：负责记录JCVM的Frame栈信息，并为JCDWP命令执行模块和VM事件管理模块提供所需的JCVM信息。

## 主机

主机为控制JavaCard运行流程的PC端软件。主机通过APDU来控制JavaCard的执行流程。本次设计不包含主机设计。

# 关键问题和解决方案

## 系统解决方案

## 系统架构

根据JavaCard系统功能结构图，将本课题所研究的系统分为3个层次结构，具体如下图所示：



## 技术选型

1. C/S架构

本文论述JavaCard调试系统基于JPDA，从构架上说， JPDA 是一个 C/S 体系结构的应用，在这个构架下，我们可以方便地通过网络，在任意的地点调试另外一个虚拟机上的程序，这个就很好地解决了部署和测试的问题，尤其满足解决了很多网络时代中的开发应用的需求。前端和后端的分离，也方便用户开发适合于自己的调试工具。

1. Eclipse调试框架
2. Java平台调试体系——JPDA
3. Java调试线协议——JDWP
4. Socket连接技术
5. 远程调试协议接口

JDWP

## 设计原则

1. 信息真实可靠

所设计的调试器所反映的信息必须真实可靠。调试器是软件开发人员依赖的查错纠错的重要工具，因而它必须真实可靠的反映调试信息。

1. 遵循Herisenberg原则？什么原则查不到资料

调试器要尽量降低对被调试系统的影响程度。调试进程也要尽量降低对被调试进程的影响程度。举例说明其重要性：对于动态调试器来说，在调试执行过程中，通常需要对被调试的程序采取插入断点、修改变量信息等手段来控制被调试的程序，此时如果插入断点、修改变量信息等操作影响到被调试程序的正常执行，调试器就不能够真实的反映调试信息，这样的调试就失去意义，进而所得到的调试信息也就不能够当做程序员定位错误的依据了。对于分布式或者并行程序的调试来说，这一点更是至关重要。

1. 提供尽量多的程序上下文

调试器提供越多的程序上下文信息，就越能帮助程序人员定位错误，跟踪错误。通常功能越强大的调试器能够越多的提供出被调试程序上下文信息。一般来说，通常调试器必须具备设置断点、单步执行、查看制定寄存器值、查看内存值、全速执行、查看变量值等基本功能。

## 关键问题及解决思路（可以没有）

1. **难点分析**

* 难点1:梳理工作难度大,关系复杂

本课题需要根据

* 难点2：实现部、市、区三级数据全面共享，提高行政执法检查效率

本项目共涉及

1. **初步解决思路：**

通过对本课题建设需求的分析，

# 预期的成果

## 研究成果

实现以下开发目标：

1. 实现实现与调试器的JDWP协议交互；
2. 实现JCRE的Java源码级调试；
3. 实现JavaCard应用的Java源码级调试；
4. 调试组件的动态管理。
5. 应用一个程序实例验证JavaCard应用调试系统的有效性和稳定性。

## 技术指标

* 精度性要求：能够准确显示虚拟机的运行状态、变量值和方法栈结构。
* 时间特性要求：虚拟机以调试方式运行时没有明显的处理等待。
* 灵活度特性要求：当虚拟机结构修改时尽量减少对调试功能的影响。
* 可靠性要求：运行过程中不会因为调试信息处理而导致系统异常退出。
* 故障处理、灾难恢复性要求：当系统没有响应时使用进程管理器结束本系统启动的进程，之后再重新启动本系统

# 论文工作计划

| **序号** | **工作任务** | **计划开始时间** | **计划用时** |
| --- | --- | --- | --- |
| 1 | 确定论文题目，收集相关资料 | 2016年7月 | 1个月 |
| 2 | 认真整理、详细阅读和分析所收集的论文材料，结合自己的观点，起草论文的开题报告 | 2016年8月 | 2个月 |
| 5 | 对研究课题的功能性需求及非功能性需求进行深入研究和分析 | 2016年10月 | 2个月 |
| 6 | 系统总体设计，对关键问题找出解决方案 | 2016年12月 | 2个月 |
| 7 | 系统详细设计 | 2017年2月 | 2个月 |
| 8 | 撰写《论文中期检查报告》，准备论文中期答辩 | 2017年4月 | 1个月 |
| 9 | 完成论文初稿，交由指导老师审阅，根据指导老师的意见，不断对论文进行修改 | 2017年5月 | 1个月 |
| 10 | 论文定稿，准备论文答辩 | 2017年7月 | 1个月 |

# 主要参考文献

1. 胡建新, 吕丽民, 李彦超． Java卡调试平台的研究与实现[J]. 计算机与数字工程, 2006, 36(9): 32-34
2. Oracle. Java Card 3 Platform Runtime Environment Specification,Classic Edition Version 3.0.4[Z]. Oracle,2011
3. IBM. 深入 Java 调试体系: 第 1 部分，JPDA 体系概览[EB/OL].<http://www.ibm.com/developerworks/cn/java/j-lo-jpda1/index.html>
4. IBM. 深入 Java 调试体系，第 2 部分: JVMTI 和 Agent 实现[EB/OL]. <http://www.ibm.com/developerworks/cn/java/j-lo-jpda2/index.html>
5. IBM. 深入 Java 调试体系，第 3 部分: JDWP 协议及实现[EB/OL]. http://www.ibm.com/developerworks/cn/java/j-lo-jpda3/index.html
6. IBM. 深入 Java 调试体系，第 4 部分: Java 调试接口（JDI）[EB/OL]. <http://www.ibm.com/developerworks/cn/java/j-lo-jpda4/index.html>
7. oracle. JVM™ Tool Interface[EB/OL]. <http://docs.oracle.com/javase/7/docs/platform/jvmti/jvmti.html>.2009
8. oracle. VirtualMachine Command Set[EB/OL]. <http://docs.oracle.com/javase/7/docs/platform/jpda/jdwp/jdwp-protocol.html>
9. oracle. Java™ Debug Wire Protocol[EB/OL]. <http://docs.oracle.com/javase/7/docs/technotes/guides/jpda/jdwp-spec.html>
10. oracle. Java™ Platform Debugger Architecture[EB/OL]. <http://docs.oracle.com/javase/7/docs/technotes/guides/jpda/architecture.html>
11. Java Card 3 Platform Virtual Machine Specification, Classic Edition Version 3.0.4
12. 姚欣. Java Card调试系统中3J模块的设计与实现[D]. 北京:北京交通大学, 2011
13. 赵民栋.嵌入式软件集成开发环境中调试器的设计与实现[D]. 陕西:西北工业大学, 2004.3-7
14. 连丽红.嵌入式调试技术的研究与实现[D].福建.厦门大学, 2009.6-7
15. 李铠鑫. JAVA卡调试平台的设计与实现[D]. 北京:北京邮电大学, 2011. 1-50
16. 刘剑. Java卡测试平台的设计与实现[D]. 北京:北京邮电大学, 2010
17. 王爱英. 智能卡技术：IC卡与RFID标签[M]. 北京:清华大学出版社, 2009. 55-80
18. 张之津, 李胜广, 薛艺泽. 智能卡安全与设计[M]. 北京:清华大学出版社, 2010. 86-108
19. 李国俊, 董晶晶, 周瑾. 智能卡COS安全性测试研究[J]. 计算机技术与发展, 2014, 24(2): 164-167
20. 周志明. 深入理解Java虚拟机：JVM高级特性与最佳实践（第2版）[M]. 北京:机械工业出版社, 2013.
21. 葛一鸣. 实战Java虚拟机：JVM故障诊断与性能优化[M]. 北京:电子工业出版社, 2015.
22. Eric Clayberg, Dan Rube. Eclipse插件开发（原书第3版）（畅销插件开发指南新版）[M]. 北京:机械工业出版社, 2010
23. 张鹏, 姜昊, 许力. Eclipse插件开发[M]. 北京:电子工业出版社, 2008
24. Java Card 3 Platform Virtual Machine Specification, Classic Edition Version 3.0.4
25. Global Platform Card Specification Version 2.2.1 Public Release January 2011