

aicas news

Neuigkeiten für Softwareentwickler kritischer Anwendungen

Herbst 2005

↓ Editorial

Lieber Leser

Ich freue mich, Ihnen die aktuellsten Neuigkeiten und technischen Informationen von aicas präsentieren zu können. Schon wieder liegt ein arbeitsreicher Sommer hinter uns. Mehrere Pilotkunden verwenden bereits mit Erfolg unsere neue Swing-Implementierung. Diese wird mit der Release 2.8 von JamaicaVM allgemein verfügbar. Außerdem bereiten wir ein neues Produkt für Sie vor: sicherheitskritisches Java. Dies wäre jedoch keine vollständige aicas news, ohne einen technischen Artikel über Echtzeit-Java-Technologie.

Dieser Hauptartikel beleuchtet verteilte Echtzeit-Java-Applikationen mit CORBA von Object Interface Systems. OIS erklärt die Echtzeit CORBA-Funktion und deren Einsatz. Selbstverständlich unterstützt ORBexpress RT die JamaicaVM.

Ich hoffe, diese Ausgabe entspricht Ihren hohen Erwartungen. Wir freuen uns auf Ihre Rückmeldung. Natürlich sind wir ganz besonders an Ihren technischen Anforderungen interessiert.

Mit freundlichen Grüßen,

Dr. James Hunt,
Geschäftsführer aicas GmbH

JamaicaVM Swingt!

JamaicaVM 2.8 unterstützt AWT und Swing Grafik

Zur Erntezeit hat das aicas Team die neueste Release der JamaicaVM fertiggestellt. Die Version 2.8 erweitert das Anwendungsspektrum der JamaicaVM erheblich. Die Highlights sind Grafik-Support, verbesserte Performance und eine höhere Benutzerfreundlichkeit der VM.

Ein wichtiger Fortschritt in dieser Release ist die Unterstützung einer Untermenge von AWT und Swing. Diese ermöglicht es Ihnen, anspruchsvolle grafische Anwendungen auf eingebetteten Systemen ablaufen zu lassen. Grafikunterstützung gibt es für VxWorks, OS-9 und Unix. Grafik für Windows CE ist aktuell in Arbeit und wird Ende 2005 verfügbar. Dank ihrer echtzeitfähigen, automatischen Speicherverwaltung ermöglicht JamaicaVM 2.8 das Entwickeln moderner Anwendungen für eingebettete Systeme. Diese fordern nicht nur eine schöne, vielfältige Benutzeroberfläche, sondern vor allem zuverlässiges und vorhersagbares Verhalten.

Laut unseres Partners NeoMore erfüllt JamaicaVM alle Anforderungen, die momentan

an Java in kritischen Echtzeitanwendungen gestellt werden. Die hohe Qualität und die innovative Natur der JamaicaVM bildeten die Grundlage für ihre Entscheidung, JamaicaVM Vertriebspartner zu werden.

JamaicaVM ist für Evaluierungszwecke kostenlos verfügbar. Die Release 2.8 kann von der aicas Webseite www.aicas.com herunter geladen werden.



↓ Neuigkeiten

Neue Vertriebspartner

Das Vertriebs- und Support-Netzwerk von aicas expandiert mit neuen Partnern in der Schweiz und in Frankreich.

NetModule hat kürzlich eine Vertriebsvereinbarung mit aicas für die JamaicaVM in der **Schweiz** unterzeichnet. NetModule ist eine Technologiefirma, die eingebettete Java-Lösungen anbietet. Ihre Dienstleistungen reichen von Hardware-Entwicklung bis hin zur Gesamtlösung

inklusive OS, Java VM und Applikation. Ihre Produkte umfassen Hardware-Plattformen wie z.B. für Java optimierte Bedien- & Steuer-Terminals, das Software-Framework JPC (Java for Process Control) und die JOPC-Bridge (eine Java-Anbindung an OPC von Microsoft).

Die 2003 von Programmierexperten gegründete Firma **NeoMore**, bietet den Kunden in **Frankreich** Lösungen für Embedded Systems an. Dies beinhaltet u.a. Softwareentwicklungs-

werkzeuge, Echtzeitbetriebssysteme, Kommunikationsschnittstellen, Middleware und Hardware. Das NeoMore-Team hat 15 Jahre Erfahrung in eingebetteten Systemen.

NeoMore und NetModule sind bedeutende neue Partner für unser weltweites Vertriebs- und Dienstleistungsnetz. Mehr Informationen über NeoMore, NetModule und unsere anderen Partner finden Sie auf unserer Webseite:

www.aicas.com/resellers.html

Echtzeit CORBA für Echtzeit-Java

Ein Beitrag von Victor Giddings und Joseph Jacob von Objective Interface Systems, Inc.

Einführung

Die Real-Time Specification for Java (RTSJ) bietet, insbesondere mit deterministischer Garbage Collection, eine Standardplattform zur Echtzeitprogrammierung in Java und bringt damit den "Write Once, Run Everywhere" Ansatz von Java zu den Echtzeitemwicklern. Eine RTSJ Implementierung kann die Anforderungen eines Echtzeitsystems erfüllen, etwa die Fähigkeiten, Tasks zu planen, die Ausführung dieser Tasks vorherzusagen und insbesondere vorherzusagen, wie das System bei Überlastung reagieren wird. Echtzeitsysteme sind oft aus miteinander vernetzten Prozessoren aufgebaut, die auch Echtzeitgarantien für die Kommunikation bieten müssen. Echtzeit-CORBA ist die erste Standard-Middleware für die Echtzeitkommunikation zwischen Prozessorknoten in einem Echtzeitsystem.

RT-CORBA ermöglicht den Entwicklern eine Vorhersage des Verhaltens eines gesamten Systems zu erstellen und bietet gleichzeitig Ortstransparenz bei der Entwicklung, beim Test und im Einsatz. Im Gegensatz zur Common Object Request Broker Architecture (CORBA) und der Remote Method Invocation (RMI) bietet RT-CORBA Mechanismen zur Verwaltung der Weitergabe von Prioritäten zwischen den Knoten. Diese Weitergabe ist notwendig, um eine konsistente Ausführungsreihenfolge an allen Prozessorknoten zu garantieren. Die RT-CORBA Spezifikation erweitert CORBA um diese und andere Fähigkeiten wie prioritätsgebundene Verbindungen, Thread Pools und Mutexe.

Bei der Weitergabe von Prioritäten wird die Priorität des aufrufenden Threads benötigt, damit der Server den Aufruf mit dessen Priorität ausführen kann. Dadurch kann die Scheduling-Analyse garantieren, dass eine Anwendung, welche die Deadlines auf einem Prozessor einhält, genauso auch auf einem verteilten System ausgeführt werden kann. Herkömmliche Echtzeit-Scheduling-Analyse sieht den Prozessor als die wichtigste geteilte Ressource an, deren Verwendung genau geplant sein muss um Deadlines einzuhalten. In einem verteilten System sind die Netzverbindungen auch geteilte Ressourcen. Damit diese Ressource zwischen Prozessen mit unterschiedlichen Prioritäten auf-

geteilt werden kann, spezifiziert das Feature der prioritätsgebundenen Verbindungen den Aufbau von multiplexierten Verbindungen und deren Zuweisung zu unterschiedlichen Prioritäten.

Threadpools bieten genauere Kontrolle über die Anzahl und Prioritäten der Threads, die zur Verarbeitung von Anfragen auf einem Server verwendet werden. Keines dieser Features ist effektiv, ohne einen Object Request Broker (ORB), der die Prioritäten berücksichtigt. Um für andere Implementierungen offen zu sein, wird keine bestimmte Scheduling-Strategie vorausgesetzt. Das ORB muss jedoch Mutexe bereitstellen, damit die Anwendungen mit einer konsistenten Strategie erstellt werden können.

Mit ORBexpress RT können Entwickler RT-CORBA verwenden, um die deterministische Ausführung von Echtzeit-Java-Programmen in verteilten Umgebungen zu gewährleisten. Es ist eine Implementierung der RT-CORBA Spezifikation der Object Management Group. Die Funktionen, die ORBexpress zur Verfügung stellt, ermöglichen eine vorhersagbare Kombination von ORB und Anwendung. Die Anwendung verwaltet Ressourcen über die Schnittstellen von ORBexpress, während ORBexpress die Aktivitäten der Anwendung koordiniert.

ORBexpress RT ist für C++, Java und Ada verfügbar. Obwohl jeder ORB auf der gleichen Architektur basiert, ist jeder ORB vollständig in seiner nativen Sprache geschrieben. Das gibt Object Interface Systems (OIS) die Möglichkeit, ORBexpress RT für Java zu testen und mit ORBexpress RT für andere Sprachen zu vergleichen. Da jeder ORB die gleiche Architektur hat, wäre es interessant die Leistung und die Vorhersagbarkeit zwischen den Sprachen zu vergleichen.

Echtzeitverhalten erreichen

Ein ORB kann nur so gut sein wie das System, auf dem er läuft. Wenn ein ORB auf einem Betriebssystem läuft, das keine Prioritäten berücksichtigt, kann der ORB kein Verhalten zeigen, das Prioritäten berücksichtigt. Ein wichtiger Test für jede Echtzeit-JVM ist die Vorhersagbarkeit ihres Verhaltens.

In den letzten Jahren hat Objective Interface Systems einen Test entwi-

ckelt, der mit ORBexpress ausgeliefert wird. Der Test, *rttaskdemo* genannt, untersucht das Verhalten des Betriebssystems, der JVM, des ORB, des Stacks usw. im Hinblick auf die Berücksichtigung der Prioritäten. Dieser Test erstellt, zusammen mit anderen internen Tests, einen Benchmark für die Geschwindigkeit von Echtzeit-Java. Diese Tests zeigen, dass RT-CORBA (z.B. ORBexpress) zusammen mit einer Echtzeit-JVM (z.B. JamaicaVM) Echtzeitemwickler unterstützt mit einer deterministischen Plattform für

- hochportable Echtzeitsysteme,
- vorhersagbare Echtzeitkommunikation,
- eine höhere Abstraktionsebene über dem Betriebssystem und
- Interoperabilität mit anderen RTOS- und Sprachumgebungen.

Ein wichtiger Punkt, mit dem Echtzeitemwickler konfrontiert werden, ist die Priority Inversion. Wenn Threads Ressourcen teilen, können überraschende Effekte auftreten. Eine Echtzeit-JVM verwendet prioritätsbasiertes, präemptives Scheduling, um Threads zu verwalten. Die JVM weist jedem Thread eine eindeutige Priorität zu. Der Scheduler garantiert, dass von den Threads, die zur Ausführung bereit sind, derjenige mit der höchsten Priorität immer zuerst ausgeführt wird. Um dieses Ziel zu erreichen kann der Scheduler einen Thread mit niedriger Priorität unterbrechen, selbst wenn er mitten in der Ausführung ist.

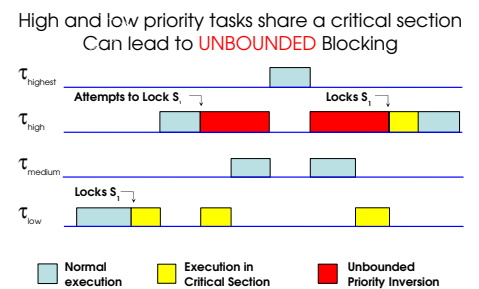


Bild 1: Priority Inversion

Weil Threads Ressourcen teilen, können Ereignisse außerhalb der Kontrolle des Schedulers den bereitstehenden Thread mit der höchsten Priorität an der Ausführung hindern. Wenn das passiert, könnte eine kritische Deadline verpasst werden, was das Versagen des Systems verursachen würde (s. Bild 1). Priority Inversion ist der Begriff für ein Szenario, in dem der Thread mit der höchsten Pri-

orität nicht zum geplanten Zeitpunkt ausgeführt werden kann. Kurz gesagt, beansprucht ein Thread mit niedriger Priorität eine Ressource, die von einem Thread mit hoher Priorität benötigt wird, blockiert den Thread mit hoher Priorität bis der Thread mit niedriger Priorität diese Ressource freigibt. Dies führt zu einer "Umkehr" der relativen Prioritäten dieser zwei Threads.

Der Schlüssel zu einem vorhersagbaren System ist die Begrenzung der Priority Inversion. Diese kann in der Scheduling-Analyse betrachtet werden. Unbegrenzte Priority Inversion macht den Taskablauf unberechenbar.

Das Beispiel in Bild 2 illustriert, dass ein Echtzeit-ORB ohne einen echtzeitfähigen Unterbau kein zuverlässiges vorhersagbares Verhalten bieten kann. Dieser Test mit *rtaskdemo* wurde mit ORBexpress RT für C++ auf Windows XP ausgeführt. Jeder Punkt im Graphen (die dicken "Linien" sind Abfolgen von Punkten) repräsentiert die Zeit der Beendung eines Threads, der einige ferne CORBA Aufrufe ausführt. Die Threads werden "gleichzeitig" gestartet. Die Zeiten der Beendungen sind in Bezug auf die Priorität dargestellt. Ein perfektes Ergebnis wäre die Beendigung aller Threads mit der höchsten Priorität, bevor die nächst höchste Prioritätsstufe irgendeine Ressource beansprucht. Die niedrigste Prioritätsstufe (Priorität 0) sollte erst ausgeführt werden, wenn alle anderen Prozesse beendet sind.

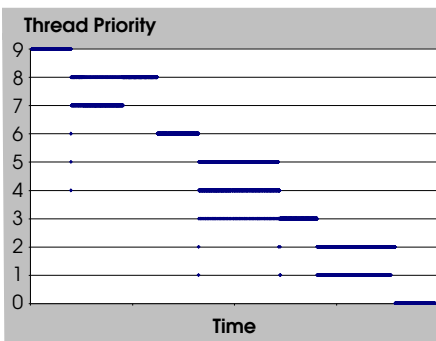


Bild 2: RT-CORBA für C++ auf Windows XP

Eine Implementierung von RT-CORBA auf einem RTOS reicht allein auch nicht aus, um Echtzeitgarantien einzuhalten. Bild 3 verdeutlicht die Unfähigkeit von Standard-Java, die Ausführung in der richtigen Reihenfolge zu garantieren. Das Beispiel zeigt ORBexpress RT auf einer Standard-JVM. Es gibt verschiedene Störungen in denen Tasks außerhalb der Reihenfolge ausgeführt wurden. Das Fehlen eines echtzeitfähigen Garbage Collectors (GC) führt zu einem unvorhersagbaren Systemverhalten. Das ist nicht un-

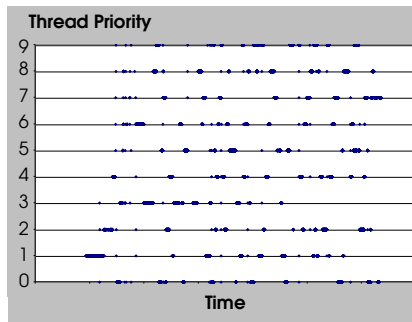


Bild 3: RT-CORBA für eine Standard-JVM und einem RTOS

bedingt ein Problem für Systeme ohne Echtzeitanforderungen, aber für einen Echtzeitentwickler macht es das System unbrauchbar.

ORBexpress RT und die JamaicaVM

Bild 4 zeigt das Prioritätsverhalten von RT-CORBA auf einer Echtzeit-JVM und einem Echtzeitbetriebssystem. In diesem Beispiel werden alle Tasks in der Reihenfolge der Prioritäten ausgeführt. Dieses Verhalten kann sowohl mit *NoHeapRealtimeThreads* als auch mit normalen *RealtimeThreads* zusammen mit einem echtzeitfähigen GC erreicht werden.

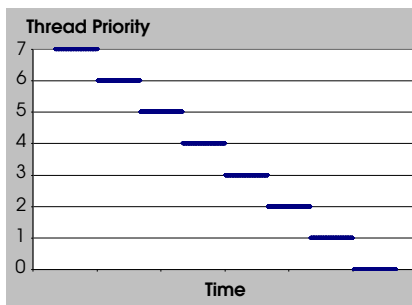


Bild 4: RT-CORBA für eine RTSJ JVM und einem RTOS

Der Nachteil von RTSJ mit einem normalen GC ist, dass Entwickler den Speicher explizit verwalten müssen. Nur *NoHeapRealtimeThreads* können in diesem Fall verwendet werden. Das beschränkt die Verwendung von Speicher in Echtzeit-Tasks auf die neuen Speicherbereiche, die in der RTSJ definiert sind: *ScopedMemory* und *ImmortalMemory*. Wenn ein Thread in einem dieser Speicherbereiche ist, allozieren alle "new" Operationen Speicher in diesem Bereich statt auf dem Heap. Nachdem der letzte Thread den Bereich verlässt, wird der Speicher in einem *ScopedMemory* freigegeben; Objekte im *ImmortalMemory* belegen Speicher bis zum Ende der Anwendung.

Die RTSJ stellt auch ein bedeutendes Hindernis bei der Implementierung eines ORBs dar. Die neuen Speicherbereiche sind an Threads gebunden,

↓ **Veranstaltungen**

Wir laden Sie ein, uns auf folgenden Messen zu besuchen:

Embedded Technology 2005, Yokohama
16. - 18. November 2005

SPS 2005, Nürnberg
22. - 24. November 2005

Embedded World 2006, Nürnberg
14. - 16. Februar 2006

RTS Embedded Systems 2006, Paris
04. - 06. April 2006

Hannover Messe Industrie 2006, Hannover
24. - 28. April 2006

nicht an Objekte. Der ORB wird als Bibliothek implementiert, die während des Systemsablaufs immer wieder betreten und verlassen wird. Das macht es schwierig, einen konsistenten internen Zustand des ORBs zu bewahren. Sobald die Anwendung *ScopedMemory* betreten hat, kann der ORB keine Informationen eines Aufrufs ohne Kopieren in einen vom ORB kontrollierten Bereich speichern; ansonsten könnte er eine Verletzung des Gültigkeitsbereichs verursachen.

Entwickler können diese Anforderungen an die Speicherverwaltung durch einen deterministischen GC vermeiden. Das macht die JamaicaVM mit ihrem echtzeitfähigen GC optimal geeignet für Entwickler, die alle Vorteile des RT-CORBA nutzen wollen. Dazu ist der Java-ORB schnell: er übertrifft die Leistung des ADA-Produkts und vieler C++-ORBs.

Zusammenfassung

Echtzeit-JVMs können hartes, echtzeitfähiges, vorhersagbares Verhalten bieten. Diese Vorhersagbarkeit kann mit RT-CORBA auf verteilte Systeme ausgeweitet werden. Die Verwendung eines echtzeitfähigen GCs ist entscheidend, um die Restriktionen der RTSJ bezüglich der Speicherverwaltung zu vermeiden. Auch die Leistung ist konkurrenzfähig. Diese Vorteile können schon jetzt durch die Verwendung von ORBexpress RT mit der JamaicaVM genutzt werden.

Ausblick

Die nächste Ausgabe wird die Analyse des Ressourcenbedarfs eines Java-Programms vorstellen.

Das GNU Classpath Projekt

aicas arbeitet zusammen mit anderen Anbietern von Java Technologie am GNU Classpath Projekt

Einführung

Keine Java VM wäre vollständig ohne die Standard Klassen. Die JamaicaVM hat ihre eigenen Kernklassen, verwendet aber auch Klassen des Open Source Projekts GNU Classpath. GNU Classpath ist Teil des GNU Projekts, welches sich zum Ziel gesetzt hat, eine komplette Open Source Entwicklungsumgebung zu entwickeln. Classpath möchte eine völlig kompatible Implementierung aller Java Standard Klassenbibliotheken erreichen. Besondere Stärken von GNU Classpath bestehen in Bereichen, in denen man auf die Funktionen anderer freier Bibliotheken zurückgreifen kann. So werden die Daten für die Internationalisierung aus dem CLDR-Projekt extrahiert und dadurch die Unterstützung für 274 Sprachen er-

möglicht. SUNs Java unterstützt dagegen nur 134 Sprachen.

Geschichtliches

1998 wurde das Projekt GNU Classpath gegründet, um eine freie Java-Klassenbibliothek für die virtuelle Maschine Japhar zu schreiben. Zunächst dachte noch niemand daran, andere virtuelle Maschinen zu unterstützen. Das ursprüngliche Team bestand gerade mal aus fünf Entwicklern.

Im Laufe der Zeit erweiterte man das Ziel dahingehend, dass man mehrere VMs unterstützen wollte. Im Jahr 2000 wurde die Klassenbibliothek von GCJ einverleibt und es kamen weitere Entwickler hinzu. Drei Jahre später schloss sich das Projekt Kaffe Classpath an und begann damit, Patches mit den Classpath Entwicklern auszutauschen. Zu diese Zeit wurde Classpath auch in die JamaicaVM integriert.

Vorher bestand die Klassenbibliothek von Jamaica aus einer eigenen Implementierung der Kern-Packages. Obwohl einige hoch spezialisierte Klassen der VM sowie der Echtzeit Support immer noch im Haus entwickelt werden, stammt nun ein Großteil der Klassen von GNU Classpath.

Das aicas Team ist nun ein bedeutender Mitwirkender im Classpath Projekt. Es besteht ein reger Austausch von Patches zwischen aicas und GNU Classpath. Im Rahmen von Classpath konzentriert sich die aicas Entwicklung vor allem auf Codegröße, Plattformunabhängigkeit und die Grafikpakete AWT und Swing.

Mitwirkende

Betreut wird das GNU Classpath Projekt von der Free Software Foundation. Zur Zeit gibt es ca. 20 aktive Entwickler. Etwa die Hälfte der Entwickler stammen von den kommerziellen Trägern Red Hat und aicas, die andere Hälfte von freien Software-Projekten.

Rechtliches

GNU Classpath wird unter einer Open Source Lizenz der Free Software Foundation vertrieben. Diese Lizenz ist eine erweiterte Form der GNU Public License (GPL), welche die Integration der Classpath Bibliotheken in kommerzielle Produkte erlaubt. Andere Programme dürfen GNU Classpath verwenden, ohne dass sie selbst Ihren Quellcode offen legen müssen.

Aktueller Stand

Alle Kernklassen der J2SE 1.2 sind komplett implementiert. Volle Kompatibilität zu J2SE 1.4 ist in Arbeit. Große Teile der Klassen von J2SE 1.4 sind bereits implementiert und konform mit der Spezifikation. Der Kern von AWT ist abgeschlossen, die Widgets sind jedoch nicht komplett. Die Swing Implementierung basiert auf AWT und viele der einfachen Widgets funktionieren bereits. Swing wird in vielen Projekten bereits erfolgreich eingesetzt und geht schnell der Vollendung entgegen.

Zukünftige Entwicklungen

Die größte Entwicklungsaktivität findet momentan bei den graphischen Toolkits AWT und Swing statt. Speziell Red Hat und aicas steuern sehr viel Code zu diesen Bereichen bei. Ebenso wird die Arbeit an der 1.4 Kompatibilität fortgesetzt. Danach werden die Java-5-Erweiterungen implementiert.

GNU Classpath und aicas

Seit 2002 werden Klassen von Classpath in der JamaicaVM verwendet. Damit hat die JamaicaVM eine vollständige 1.2 Implementierung und ist zu großen Teilen kompatibel zu JDK 1.4. Aufgrund der speziellen Anforderungen von Echtzeit Systemen verwendet Jamaica eine komplett eigene Implementierung des Kern-Packages *java.lang* sowie eine Reihe von Modifikationen in den anderen Kern-Paketen *java.io*, *java.net* und *java.util*. aicas verwendet aber nicht nur das Classpath Projekt, sondern trägt auch zu dessen Entwicklung bei. Bevor aicas Classpath Code zusammen mit der JamaicaVM heraus gibt, wird er intensiv im Haus getestet, um die Integrität von aicas JVM Produkten zu gewährleisten. Während sich Classpath auf Desktop Systeme fokussiert, konzentriert sich aicas stark auf die Bedürfnisse von eingebetteten Systemen.

↓ Neues Produkt

Sicherheitskritisches Java

Die erste Umsetzung des im HI-JA Projekt definierten, sicherheitskritischen Java Profils wird von aicas kommen. Diese Variante von JamaicaVM wird zertifizierbar nach dem Sicherheitsstandard DO-178B, Level A, der höchsten Zuverlässigkeitsstufe.

Sicherheitskritisches Java wird eine Untermenge der RTSJ und Java Standard Klassen enthalten. Schwer zertifizierbare Features, wie der Garbage Collector, werden nicht enthalten sein.

Es werden verschiedene Werkzeuge für die statische Analyse sicherheitskritischer Java-Anwendungen bereit gestellt. Diese Werkzeuge unterstützen bei der Analyse des Ressourcenbedarfs (maximaler Heap und Stack Bedarf), Überprüfung korrekter Verwendung von Scoped Memory und der Bestimmung der Worst-Case Execution Time eines Programms. Ein weiteres Werkzeug überprüft, ob Java Laufzeitfehler, wie *NullPointerException* oder *ClassCastException*, in einer Applikation auftreten können.

Eine Vorversion wird Ende 2005 verfügbar sein und die erste Release erscheint mit JamaicaVM 3.0 im Frühjahr 2006.

↓ Kontakt

aicas GmbH

Haid-und-Neu-Straße 18
D-76131 Karlsruhe, Germany

tel +49 721 663 968-0
fax +49 721 663 968-99
email info@aicas.com
web www.aicas.com