Traumhochzeit:

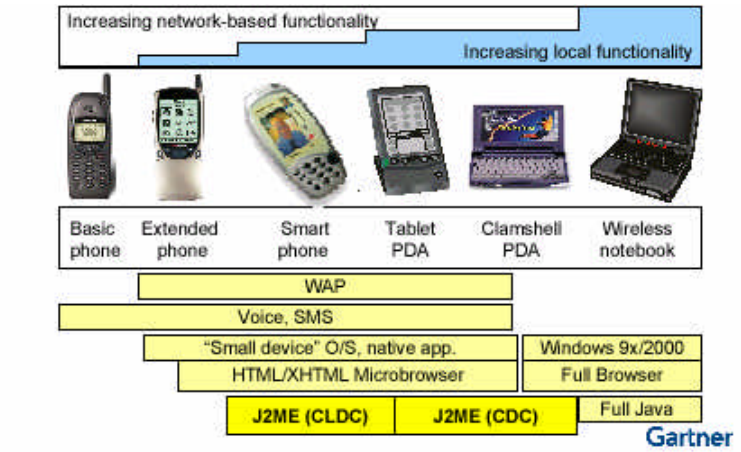
Java und Palm

Dr. Gernot Starke
gernot@starke-team.de
Tel. +49-177 - 728 2570

The Future Is...

Mobile

Rolle von Java in Wireless & Mobile Devices



Source: Gartner Research

- Wireless, Mobiles (and Embedded) devices are the the major future opportunities for Java!

Inhalt

- Palm & PDA-Konsorten
  - Geräte, Betriebssysteme
  - Programmierung
- Java 2 Micro Edition
- Entwicklung
  - Prozesse
  - Tools
- Virtuelle Maschinen

(siehe: www.claudiaschiffer.com)

Palm und Konsorten

- PDA = Personal Digital Assistants
- Palm (auch Palm Pilot, Clones von Handspring (Visor), TRG, bald auch Sony)
  - Betriebssystem Palm-OS
  - Motorola 68k CPU, 10-30MHz, 2-8MB RAM
  - 160x160 Pixel LCD Display (meist Grayscale)
- Psion
  - Betriebssystem EPOC
  - Tastatur, 640x320 LCD Farbdisplay
  - CPU bis 200Mhz!!
- Windows-CE (Casio, HP und andere)

Programmierung von PDA's

3 kritische Besonderheiten:

- Speicher
- Performance
- „Sofort"-Prinzip
  (für Bedienung & Persistenz)

Ausserdem:
- Batteriebetrieb
- Kleine Bildschirme
- K(l)eine Tastatur
- Nur temporäre Connectivity

Lösung:

Keep **i**t **S**mall & **S**imple

Wichtigste Quelle: Palm Programmers Companion & Reference Manual:
Als PDF frei erhältlich bei
www.palmcom/devzone

Besonderheiten: Wenig Speicher

2-8 MB Ram für Anwendungen und Daten!!

Beispiele:

- Palm-Tabellenkalkulation: 152kByte (zzgl. 31k für Chart-AddOn)
- 300 Adresseinträge: 34kByte
- Routenplaner Deutschland: 1MByte

Für ProgrammiererInnen:

- max. 64k Heap/Stack

Besonderheiten: Geringe Performance

10-20 MHz Taktrate...

Beispiel für Benchmark: Entschlüsselung mit RC5 Krypto-Verfahren (Greg Hewgill: RC5Java, www.distributed.net)

- H70 IBM RS6000 (AIX 4.3): 280000 Keys/Sek.
- Pentium II, 300MHz: 180000 Keys/Sek.
- Palm-V (Jbed VM) 58 Keys/Sek.
- Palm-V (KVM): 7 Keys/Sek.

Rechenintensive Anwendungen:

- Auf Server auslagern,
- Datentransfer per Synchronisation

Besonderheiten: „Sofort“-Prinzip

(Computer-) Standardverhalten	PDA
Einschalten: Bootvorgang (mehrere Minuten)	Einschalten, loslegen
Anwendung anklicken – warten	Anwendung selektieren, (1-2 Sek.), loslegen
Anwendung beenden: Daten explizt speichern	Kein Pendant... (Daten sind immer gespeichert)
Anwendung erneut starten – Daten lesen/selektieren	Kein Pendant... (an gleicher Stelle weiterarbeiten)

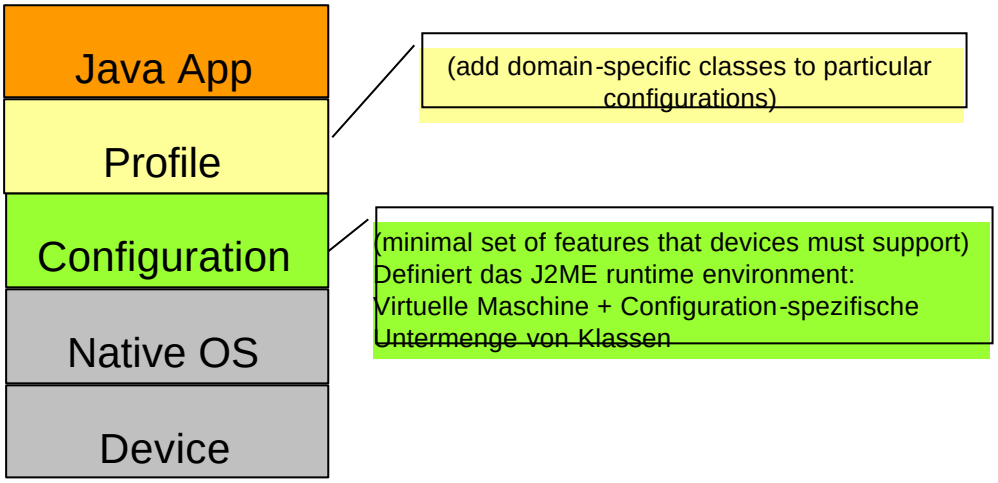
Lösungsansätze zum „Sofort“-Prinzip

- Hochoptimiertes Benutzerverhalten
  - Minimiere Anzahl „Penstrokes/Mausklicks“ („Every Klick Counts!“)
  - Minimiere Navigation zwischen Fenstern/Dialogen
- Implizite & unmittelbare Speicherung des Anwendungszustands
  - Oft kein „Dateisystem“ verfügbar (z.B. unter Palm-OS)
  - Speicherung in „Datenbanken“ (entsprechen Flat-Files)
- Probleme:
  - (konventionelle) Java-VM‘s benötigen VIEL Zeit zum Start: Classloader, Verification, JIT-Compiler, Hotspot-Compiler...

Java 2 Micro Edition

- Problem: Völlig unterschiedliche Laufzeitumgebungen
  - Mit / Ohne Tastatur
  - Mit / Ohne Display
  - Mit / Ohne Netzverbindung (LAN, WAN, Internet)
  - Speicher von wenige-10kByte bis 16MByte
- „One-Size-Fits-All“ funktioniert nicht
  - Daher: J2ME, J2SE, J2EE
- Lösungsansatz für J2ME: Configuration & Profile
  - Mindestmenge der Java-Klassen für eine Geräteklasse

J2ME Configuration & Profile



Connected Limited Device Configuration

The goal of this work is to define a standard, minimum-footprint Java platform for small, resource-constrained, connected devices characterized as follows:

- 160 kB to 512 kB of total memory budget available for the Java platform
- 16-bit or 32-bit processor

- Eingeschränkte Basisklassen (identische Paketnamen)
  - Fehlt gegenüber J2SE (Auszug): Float, JNI, Reflection, daemon Threads, user-defined ClassLoader, beans, rmi, net
- KVM (ohne Float) ist Referenzimplementierung

J2ME Profile

- Profile: Definiert API für Geräte mit „ähnlichem“ Einsatzbereich
- Beispiele:
  - Mobiltelefone: MIDP (Mobile Information Device Profile)
  - Palm Profile (in Arbeit...)

Die schlimmsten J2ME Missverständnisse

- (Alle) Java Programme laufen auf J2ME
  - Seien Sie realistisch: Nur ein (kleiner) Teil...
- KVM == CLDC = J2ME
  - J2ME ist eine Spezifikation, KVM eine VM und CLDC „nutzt“ KVM (CDC basiert auf „klassischer“ VM)
- Mit J2ME können Sie Java lernen!
  - Unkluger Ansatz – auf jeden Fall mit J2SE beginnen!!
- J2ME Programme sind portabel
  - Höchstens non-GUI Teile...

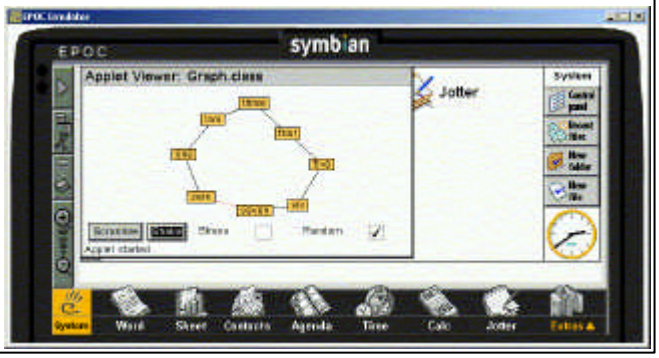
Idee: www.ericgiguere.com

Entwicklungsprozess

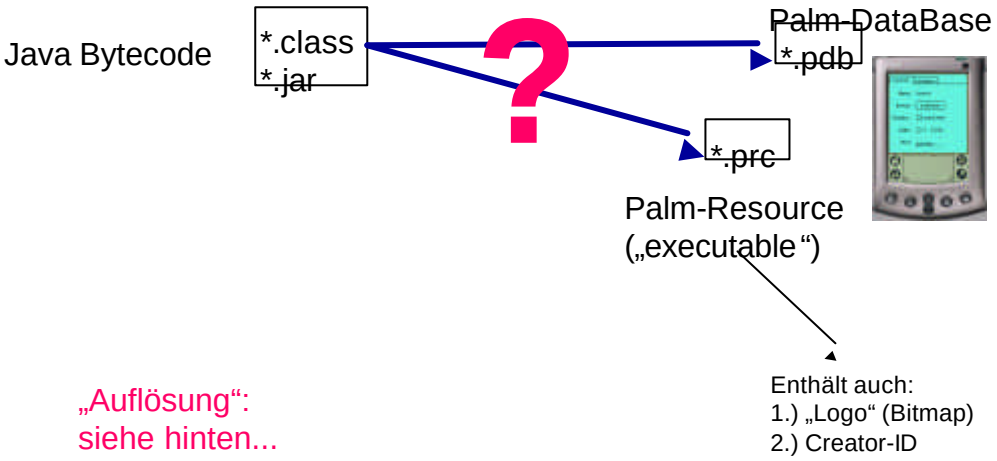
- Entwicklung unter Windows/Linux
- PDA‘s sind für viele Entwicklungsaktivitäten ungeeignet:
  - Edit/Compile
  - Debug
  - Test
- Daher: Entwicklung mit Emulatoren
  - späte Testphasen in Originalumgebung (Hard-/Software)
  - spezieller Debug-Support notwendig

Entwicklung mit Emulatoren

- POSE: Palm-OS Emulator
  - Stabil (unter Win32 ab V. 3.0)
  - Benötigt ROM-Inhalt eines Original-Palm
  - Wie das „Original“!!
  - Fazit: Unverzichtbar (weil „unkaputtbar“)
- EPOC Emulator: „Original“ JDK 1.1.4



Vorgehen bei der Palm/Java Entwicklung



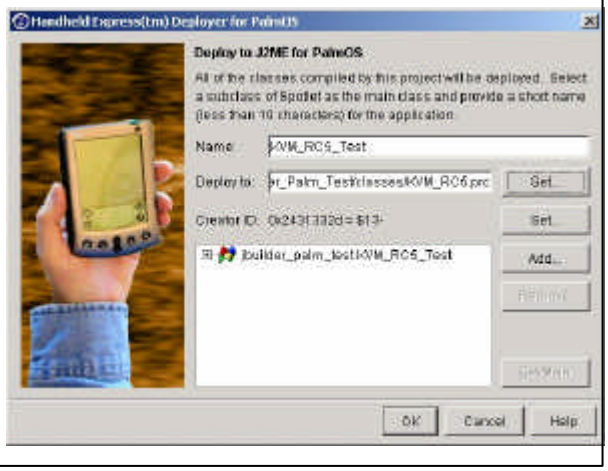
„Auflösung“:
siehe hinten...

Entwicklungsumgebungen

- Die Tools:
  - Hand-Am-Arm: Kommandozeile, emacs/vi, javac...
  - JBuilder 4 (www.inprise.com)
  - VisualAge/MicroEdition (www.embedded.oti.com)
  - JBed (www.esmertec.com)
- Tools mit spezifischer VM „verheiratet“

JBuilder 4.0

- „kleine“ Erweiterung (Handheld-Express)
- Sammlung von Wizards/Tools
- Direktes Deployment für Palm-OS
- (Voraussetzung: KVM)



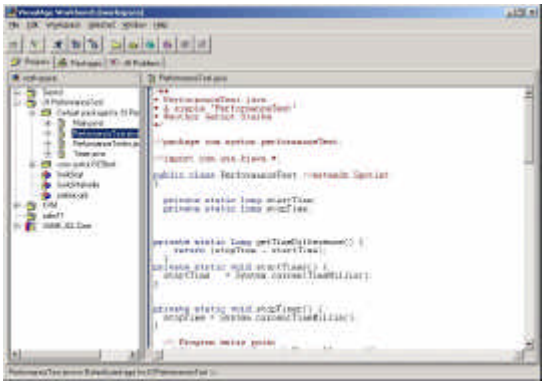
Fazit JBuilder

- Umfangreiches und bekanntes Werkzeug
- Einfache Bedienung, sinnvoller Palm-Assistent
  - Einfaches Deployment auf Emulator/Palm
- Basisversion kostenfrei

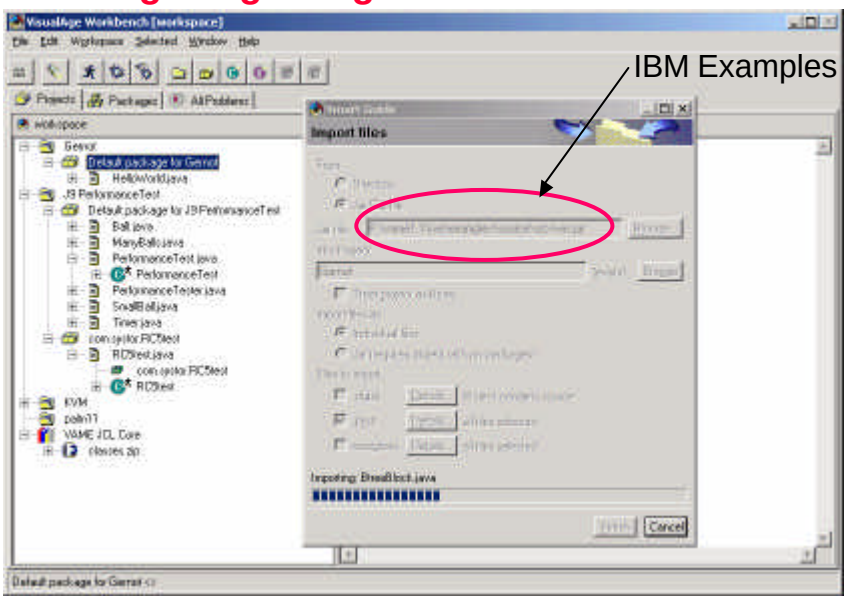
- Empfehlenswert

VisualAge/MicroEdition (Version 1.3)

- Bedienung sehr ähnlich zu „VisualAge/Enterprise“
  - Geeignet auch für GROSSE Projekte
  - Viele Klassenbibliotheken (runtimes) verfügbar
    - u.a. Palm
- Eigenes Werkzeug, deutlich kompakter
  - Installationsumfang 30MB
  - Enterprise Edition 3.5: 700MB
- Basisversion kostenfrei (ohne Team-Support)

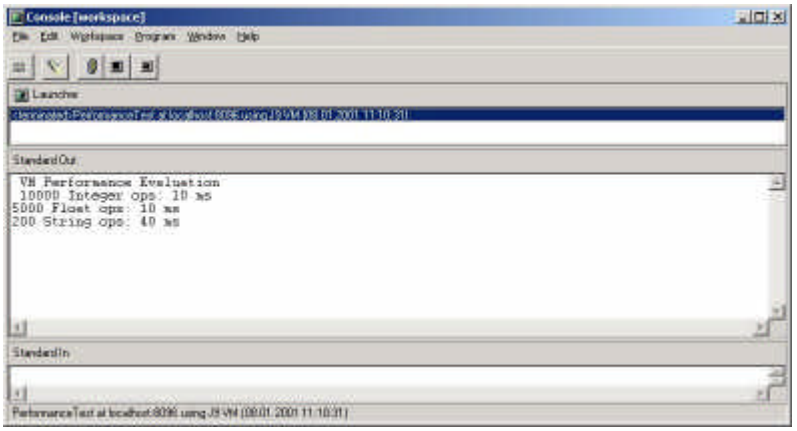


Entwicklungsumgebung VA/MicroEdition



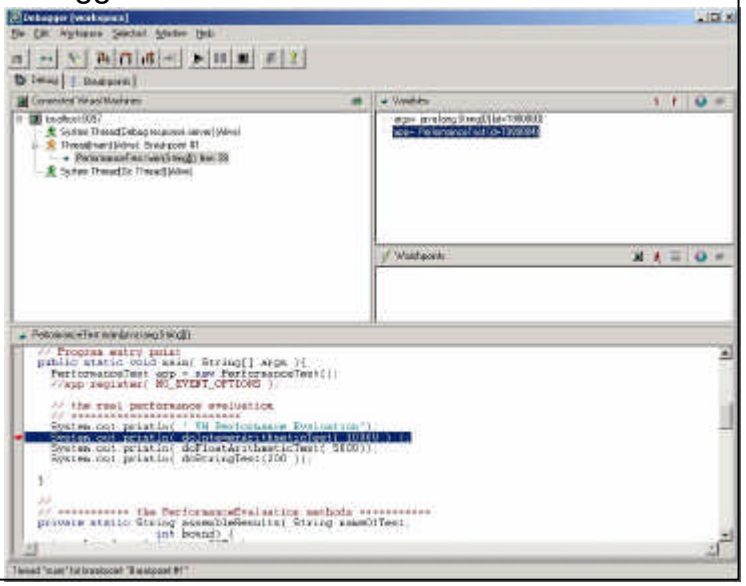
VisualAge/MicroEdition 1.2 (3)

- Ausführung von Programmen in „Konsole“



VisualAge/MicroEdition 1.2 (4)

Leistungsfähiger Debugger



VisualAge/MicroEdition & KVM

- Problematisch, wegen intransparentem „Smartlinker“
- Deployment über Makrosprache
  (ähnlich MSDos-Batch)
- Nicht-Standard Zwischenformat: JXE (Java-Executable)
  - wird aus class/jar Files erzeugt
- IBM eigene Wrapper Klassen für Palm-OS
- Nicht 100% kompatibel zu KVM
  - Beispiel: warp.prc (siehe www.microjava.com) führt auf J9 zu „Bus Error“

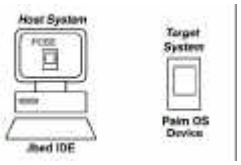
Fazit VisualAge/MicroEdition

- Für grosse Projekte geeignet
- Komfortable Entwicklung
- Basis: IBM-eigene J9-VM
- Keine Ausführung von Spotlets in VA
  - Adapter Klassen für Windows/Linux fehlen (noch?!)
- Zum Einstieg nur bedingt geeignet

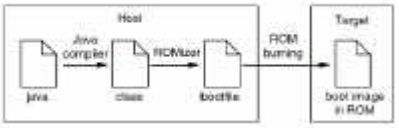
Entwicklungsumgebung „Jbed“

- Hersteller Esmertec AG
  - Spezialisiert auf Echtzeitsysteme
- Stabile Umgebung
- Ursprünglich für „Embedded & RealTime Systems“
- Testversion kostenfrei

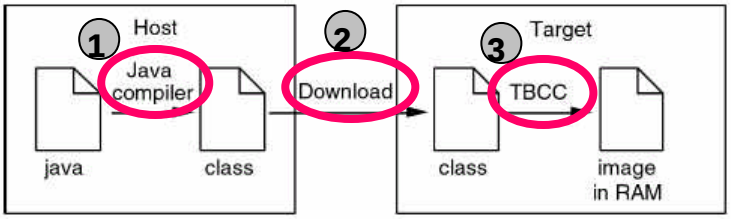
Jbed: Host- und Target Systeme



Jbed Host und Target Systeme



Jbed Closed System (Embedded...)



Jbed Open System (Embedded...)

Quellen: Esmertec

Entwicklungsprozeß (Jbed)

1.) Übersetzen der Java-Klassen mit javac (aus JDK)
1b.) Linken (Nicht-Standard)
2.) Transfer auf Palm-Emulator oder Palm-Device
3.) Übersetzung mit TBCC (Native-Compiler)
4.) Remote-Debugging (mit Jbed IDE)

Geschieht automatisch

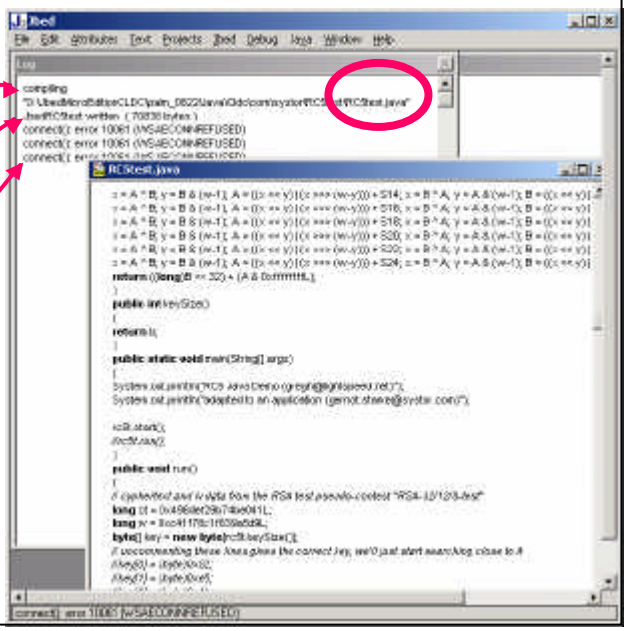
Sämtliche Schritte innerhalb Jbed IDE!

Beispiel: Erzeugen eines Palm-Programms...

javac

integrierter Linker

remote Debugger
(hier: offline!)



Jbed

Fazit Jbed IDE

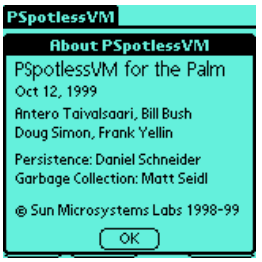
- ~~Hervorragend!~~!
  - Performant, Kompakt, Stabil
  - Technisch führende Echtzeitumgebung!!
- IDE: große Unterschiede zu VisualAge & Jbuilder
  - Standard-JDK Compiler
  - Kein Repository, keine Integration von Versionsverwaltung
- Entwicklung mit KVM und (Esmertec-eigener) Jbed-VM möglich

Jbed

Virtuelle Maschinen für PDAs

- Sun: KVM (Kilobyte Virtual Machine)
  - Sun: Spotless VM (Vorgänger von KVM)
- IBM: J9
- Esmertec: Jbed VM
- Wabasoft: Waba-VM

- EPOC/Psion: Standard JDK 1.1.4



KVM: Mutter aller PDA-VM's

- Erste offizielle Palm-Java VM in Version 0.1 (Juni 1999)
- (relativ) stabile (immer noch unvollständige) Version seit Mitte 2000
- Standard-/Referenzimplementierung der J2ME

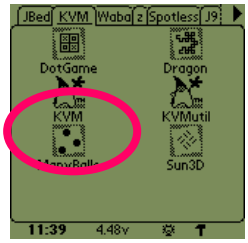
- Verfügbar unter Win32, Solaris, (etc.)
- Nutzt J2SE Compiler javac

Entwicklungsprozeß mit KVM / CLDC

- 1. compile der Java Sourcen (Standard javac 1.1.x oder 1.2.x).
- 2. preverify der Java classfiles mit dem Preverifier der *J2ME CLDC*
- 3. Packen aller Klassen in JAR Datei
- 4. Erzeugen der PRC-Datei mit MakePalmApp tool

- Kleine Probleme: Tools (MakePalmApp etc.) werden nur als Java Source ausgeliefert, passende Makefiles laufen nicht unter Win32/bash...
  (Preisrätsel: Warum liefert Sun gerade DIESE Tools nur als Source, nicht als übersetzte Klassen aus?)

Ausführung eines KVM Java Programms



Fazit KVM:

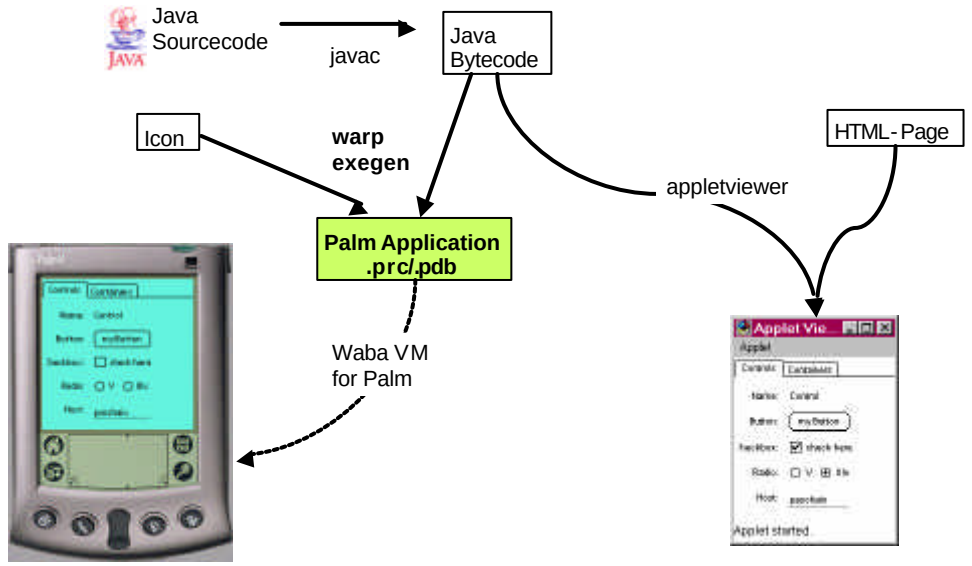
- Referenzimplementierung der J2ME
- „Gangbarer Weg“ mit Standard-Tools
- Mäßige Performance
- KVM ist hervorragend portierbar
  - u.a. auf „Geldautomaten“ und andere MicroDevices
- Auf Palm-OS: Überlange Ladezeit

- Als Einstieg in Java&Palm gut brauchbar
  - Läuft auch auf 2MB Palms
  - Kommerzielle Anwendungen: unbedingt pilotieren!!

Wabasoft Waba VM

- Waba ist NICHT Java
- Andere Basisklassen!
- Kein main(), keine Exceptions, keine Threads...
- Aber: kleine VM (74kByte), recht stabil, OpenSource

Entwicklungsprozeß mit Waba



Waba Pakete und -klassen (Auszug)

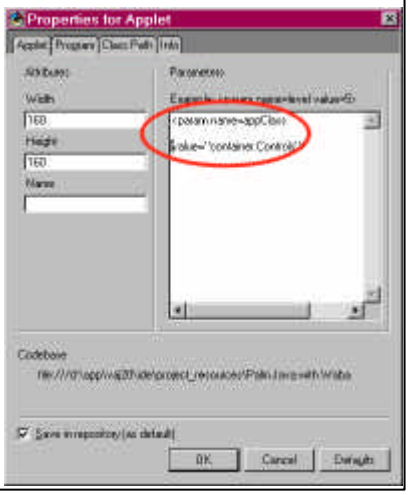
waba.fx	2D graphics, image and sound classes
waba.io	Filesystem, networking and communication classes
waba.lang	Basic language classes including Object and String
waba.sys	System classes including Convert, Time and Vm
waba.ui	User-interface classes
waba.util	General purpose classes including Vector

Button	Button is a push button control.
Check	Check is a check control.
Container	Container is a control that contains child controls.
Control	Control is the base class for user-interface objects.
ControlEvent	ControlEvent is an event posted by a control.
Edit	Edit is a text entry control.
Event	Event is the base class for all events.
KeyEvent	KeyEvent is a key press event.
Label	Label is a text label control.
MainWindow	MainWindow is the main window of a UI based application.
PenEvent	PenEvent is a pen down, up, move or drag event.
Radio	Radio is a radio control.
Tab	Tab controls appear as tabs in a TabBar control.
TabBar	TabBar is a bar of tabs.
Timer	Timer represents a control's timer.
Welcome	Welcome is the welcome application.
Window	Window is a "floating" top-level window.

Tabelle 2: Klassen von waba.ui

Fazit Waba-VM:

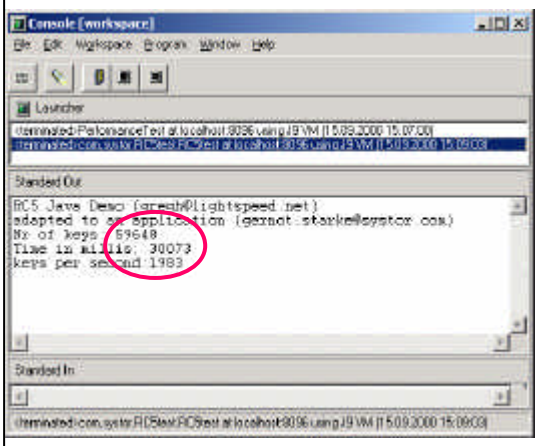
- Nicht KVM / J2ME kompatibel
  - Völlig anderes Konzept von GUI & Eventhandling
- Guter Einstieg in Java-Entwicklung
  - Mapper-Klassen für AWT -> Waba-Programme unter Standard-JDK lauffähig
  - Beispiel: Entwicklung mit VisualAge!!
- Unter Windows-CE und Palm-OS verfügbar!
  - Klassen ohne Änderung ablauffähig



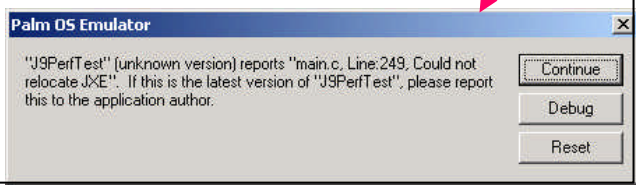
IBM J9

- Integriert mit VisualAge/MicroEdition
- Verfügbar für Windows32, PalmOS (und andere!)
- Professionelle Entwicklungsumgebung
  - Repository-Unterstützung (Versionsverwaltung!)
  - VM (theoretisch) austauschbar
  - Aber: OHNE KVM Support!!
- Palm-Unterstützung durch externe Tools
  - Kommandozeile!
  - JAR -> JXE -> PRC
- IBM-eigene Beispiele
  - Teilweise mit proprietärer GUI-Bibliothek (DEGAS)
    - ? NICHT PDA-geeignet...

IBM J9 Performance Results



- Int: 0 ms,
  float: 10 ms,
  string:10 ms
- PerformanceTest Application unter POSE nicht testbar…



Esmertec Jbed VM

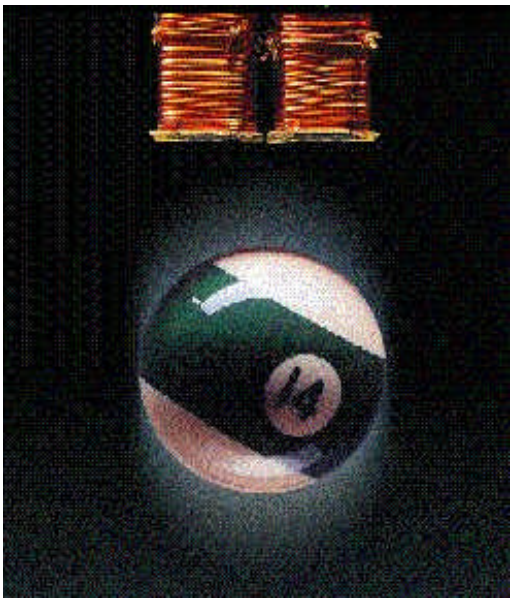
- Esmertec: Spezialisiert auf Echtzeit-Betriebssystem in Java
  - Unglaublich aber wahr! (www.esmertec.com)
  - Devicedriver, IP-Stack, Interrupt-Handler, Bootloader: Alles in Java
- Jbed IDE für Java, VM's für unterschiedliche Hardware/OS:
  - u.a. PowerPC, Palm
- Native Compiler (Target Byte Code Compiler, TBCC)
- Clean-Room Implementierung (fast) aller Standard-JDK Klassen (ausser RMI)

Jbed esmertec

Esmertec - Java in Echtzeit

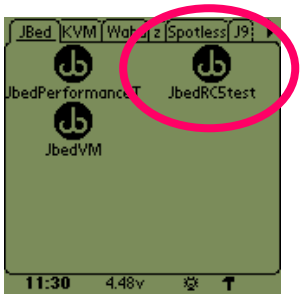
Steuerung der
„fliegenden Kugel“
in Java realisiert
(Als Echtzeit-Task)

Dynamisches Klassenladen
für Echtzeit-Anwendungen!
(z.B. Austausch des
Regelalgorithmus)



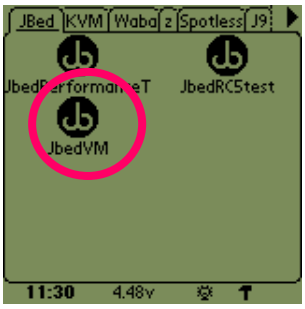
Quelle: Esmertec

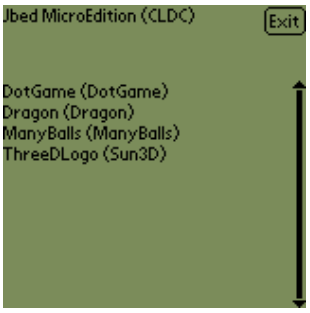
Fortsetzung Jbed-Beispiel: Ausführung...

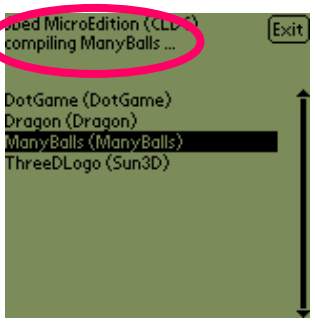


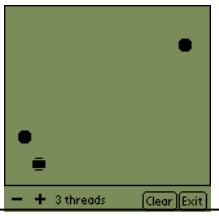
Ausführung „fremder“ Java-Klassen mit JBed

Jbed VM „sucht“ vorhandene Klassen, compiliert auf Palm!









Jbed Implementierung von JDK Klassen

Geschwindigkeit versus Speicherbedarf

- Standard-JDK Klassen garantieren „nichts“

```
/** esmertec's clean room implemenattion of java.io.StringWriter
  * @author pi
  * @version 1.0.0
  * @since 18. June 1999
  * @see java.io.StringWriter
  */
public void write( String str, int off, int len ) {
// 1.) slower but without dynamic memory allocation
      synchronized( buff_ ) {
            int end=off+len;
            for( int i=off; i<end; ++i ) {
                  buff_.append(str.charAt(i));
            }
      }
// 2.) faster, but allocates memory.
      //    char[] cb=new char[len];
      //    str.getChars(off, off+len, cb, 0);
      //    buff_.append(cb);
```

Fazit Jbed VM

- Hervorragend!!
  - Performant, Kompakt, Stabil
- Klassenbibliotheken: Meist neu implementiert (Cleanroom)
  - Fokus auf geringem Speicherbedarf und minimaler dynamischer Allokation. Gründe:
    - Kleiner Speicher von Embedded Systems,
    - Echtzeit-Tasks & Garbage-Collection!
  - Keine Lizenzgebühren (an Sun Microsystems)

Jbed

Performance Kennzahlen von Palm-VM's

	10000 int	5000 float	200 String	30 Sek. RC5 Decrypt	Details
VA/Java (JDK 1.1.7)	16	6	20	5800	VA 2.0, 330Mhz Pentium II
VA/MicroEdition mit J9	0	10	10	2260	
POSE mit Waba	3180	3670	7350		POSE 3.06a
POSE mit Jbed	330	15080	2850	58	Jbed 0.9
POSE mit J9	"------ abgestürzt ------"				NoCh VAME 1.2
POSE mit KVM	4570	X	6610	4	
Palm V mit KVM	2550	X	3630		
Palm V mit Waba	1290	1760	3880		Palm V 2MB, Waba 0.9
IBM JDK 1.1.8, Win32	0	0	20	180000	300Mhz Pentium II
IBM JDK 1.1.8, AIX	1	0	2	280000	PowerPC, H70
KVM Win32 native	20	X	20	1091	300Mhz Pentium II
X = verify error (trotz erfolgreichem preverify-Lauf!)					

Fazit

Java & PDA‘s:
- **Herausforderung, aber: „Es geht!“**
- **technische Unwägbarkeiten**
  (PalmOS versus Psion-EPOC versus Windows-CE)