



Stand:
2003

Linux
unter
Java

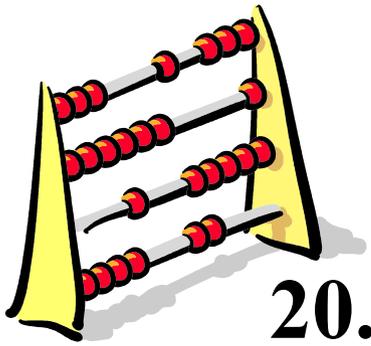
oliver.boehm@agentes.de

Java unter Linux

- Warum?
- Java Development Kit
- Entwicklung unter Linux
- Enterprise Java
- Testen
- Ausblick



Java unter Linux



20. Jahrhundert

- JDK 1.1
- Portierung von Blackdown
- geduldet
- IDE = Emacs
- Enthusiasmus gefordert

gestern
heute



- JDK 1.4
- von Sun offiziell unterstützt
- strategische Plattform
- IDE: freie Auswahl
- Normalität kehrt ein

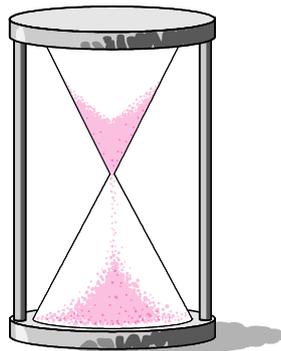
Warum Java?

- Eliminierung problematischer Features
 - keine Zeiger
 - keine Makros, Header-Dateien, ...
- Garbage Collection
- Runtime-Checks
- verbessertes Exception-Handlung

→ Java = das bessere C++

Warum nicht Java?

- Performance
- Realtime-Anwendungen
- Speicher- und I/O-intensive Programme
- Plattform-spezifische Lösung



Warum Linux?

„...Im Prinzip hast du ja auch recht. In der Schule hab' ich immer gerne unter NT programmiert. Jeder Absturz war eine Zigarettenpause :-). Aber wenn ich davon ausgehe, daß es auch Leute gibt, die nicht rauchen, dann hat Unix doch noch Existenzberechtigung.“

(Georg.Datterl@fhs-hagenberg.ac.at in de.com.advocay)

Warum nicht Linux?



- Fehlende HW-Unterstützung
- Fehlende SW / Altlasten
- Fehlende Kenntnisse (Schulung)
- Fehlender Mut
- Politische Gründe



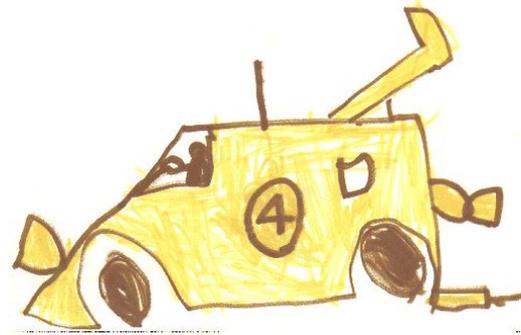
Java

Development



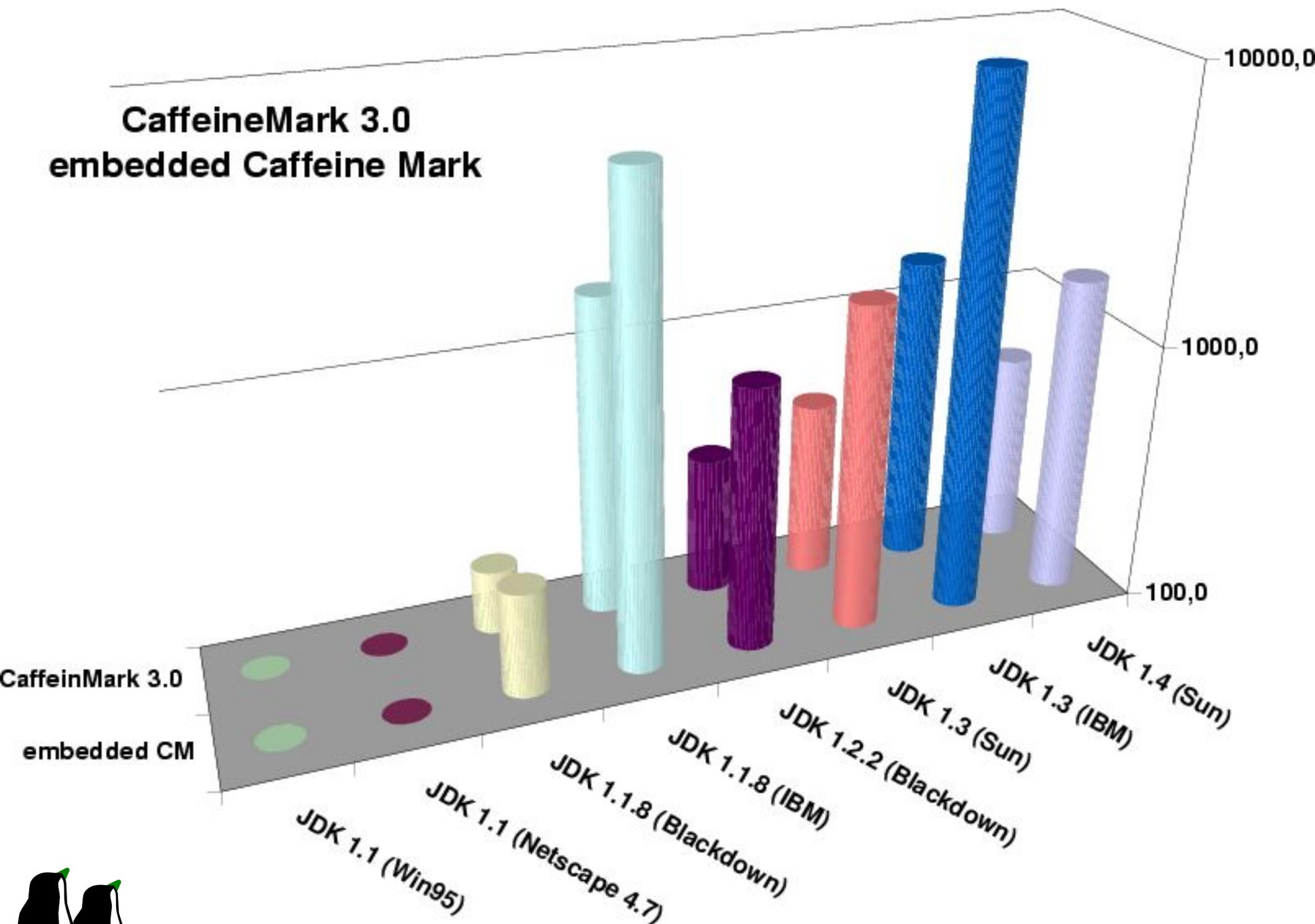
Kit

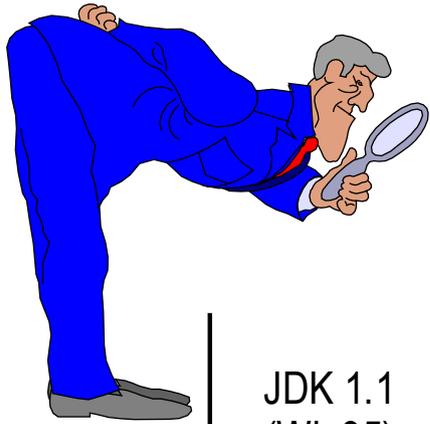
Welche JVM?



- JDK von Blackdown.org
 - erste Portierung für Linux
- JDK 1.4.1 von Sun
 - baut auf Blackdown-Portierung auf
 - Linux inzwischen offiziell unterstützt
 - die „Referenz“
- JDK 1.4 von IBM
 - schnell

CaffeineMark 3.0 embedded Caffeine Mark





Benchmark im Detail

	JDK 1.1 (Win95)	JDK 1.1 (Netscape 4.7)	JDK 1.1.8 (Blackdown)	JDK 1.1.8 (IBM)	JDK 1.2.2 (Blackdown)	JDK 1.3 (Sun)	JDK 1.3 (IBM)	JDK 1.4 (Sun)
Sieve	100	82	239	1366	832	628	1578	622
Loop	100	83	299	9199	1315	2418	10608	2624
Logic	100	61	284	196316	2239	1711	204019	1872
String	100	44	300	2601	479	3714	5658	2068
Float	100	82	191	2875	718	1932	7355	1815
Method	100	71	240	3794	1474	1628	6266	1741
Graphics	100	186	200	398	121	92	194	94
Image	100	126	134	195	22	30	15	204
Dialog	100	26	23	25	22	18	19	13
CaffeineMark 3.0	100,0	73,8	177,2	1725,8	346,9	485,8	1541,8	556,6
embedded CM	100,0	68,8	255,6	6419,5	1036,8	1766,2	9808,6	1647,4

In the computer industry, there are 3 kind of lies:
 lies, damn lies, and benchmarks.
 (aus: *The Jargon File*)



Weitere VM's

- Kaffe OpenVM
 - eigenständige PersonalJava-Implementierung
 - kompakt und konfigurierbar
 - gut geeignet für Embedded
- Japhar 
 - OpenSource-Implementierung ohne Sun-Sourcen
- ORP (Open Runtime Platform)
 - baut auf GNU Classpath auf
- GNU Classpath
 - keine VM
 - Bereitstellung eines JDKs

Compiler



- GNU Java Compiler (gcj)
 - Java Compiler (Bytecode)
 - kann aber auch native Code erzeugen (aus .java und .class-Dateien)
- Jikes
 - schnellster Java-Compiler
 - inkrementeller Build-Mechanismus
 - sehr genau (findet Fehler, die andere nicht finden)

Font-Problematik

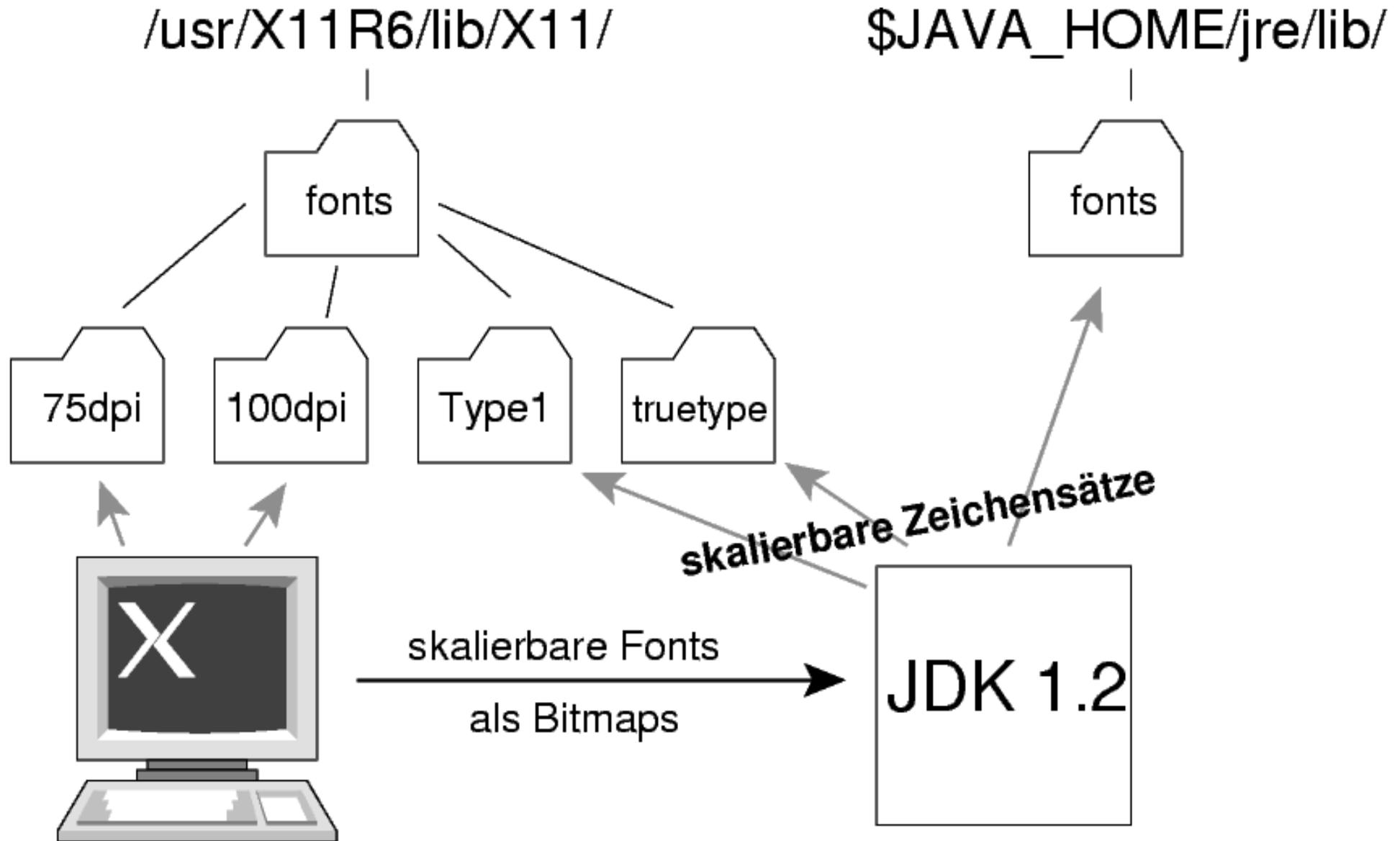


- Fehlermeldung einer Swing-Anwendung:

```
Font specified in font.properties not found [--symbol-  
medium-r-normal--*-%d-*-*p-*-adobe-fontspecific]  
Font specified in font.properties not found [--symbol-  
medium-r-normal--*-%d-*-*p-*-adobe-fontspecific]  
Font specified in font.properties not found [--symbol-  
medium-r-normal--*-%d-*-*p-*-adobe-fontspecific]
```

- Abhilfe:
 - ignorieren *oder*
 - symbol-Zeichensatz in font.properties anpassen

JAVA_FONTS

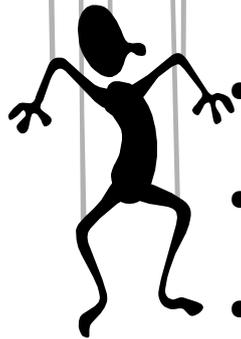


Threads



Green Threads

- Default-Implementierung
- `java -green`
- `THREADS_FLAG=green`



Native Threads

- Abbildung auf OS-Threads
- `java -native ...`
- `THREADS_FLAG=native`



Nebenwirkungen

- viele Dinge im Thread-Modell nicht spezifiziert
- dadurch unterschiedl. Verhalten möglich

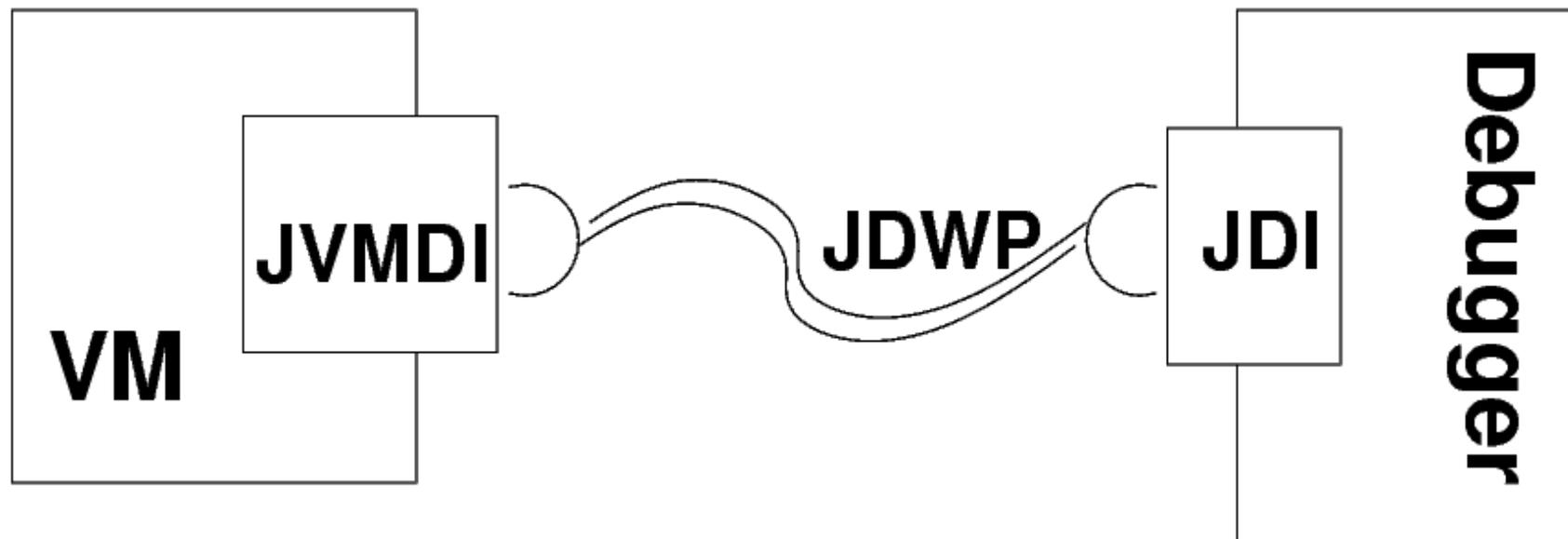
Umgebungsvariablen

- CLASSPATH: Klassen-Suchpfad
- DEBUG_PROG: Native-Debugger
- DO_NOT_CHECK_MEM (f. green threads)
- DO_NOT_FREE (f. green threads)
- JAVA_COMPILER: JIT-Compiler
- JAVA_FONTS: Suchpfad f. Zeichensatz
- JAVA_HOME: Java-Installationsverzeichnis
- JDK_NO_KERNEL_FIX (f. green threads)
- THREADS_FLAG: green / native

JDPA



- Java Platform Debugging Architecture (JDPA)
 - Java VM Debugging Interface (JVMDI)
 - Java Debug Wire Protocol (JDWP)
 - Java Debug Interface (JDI)



Entwicklung

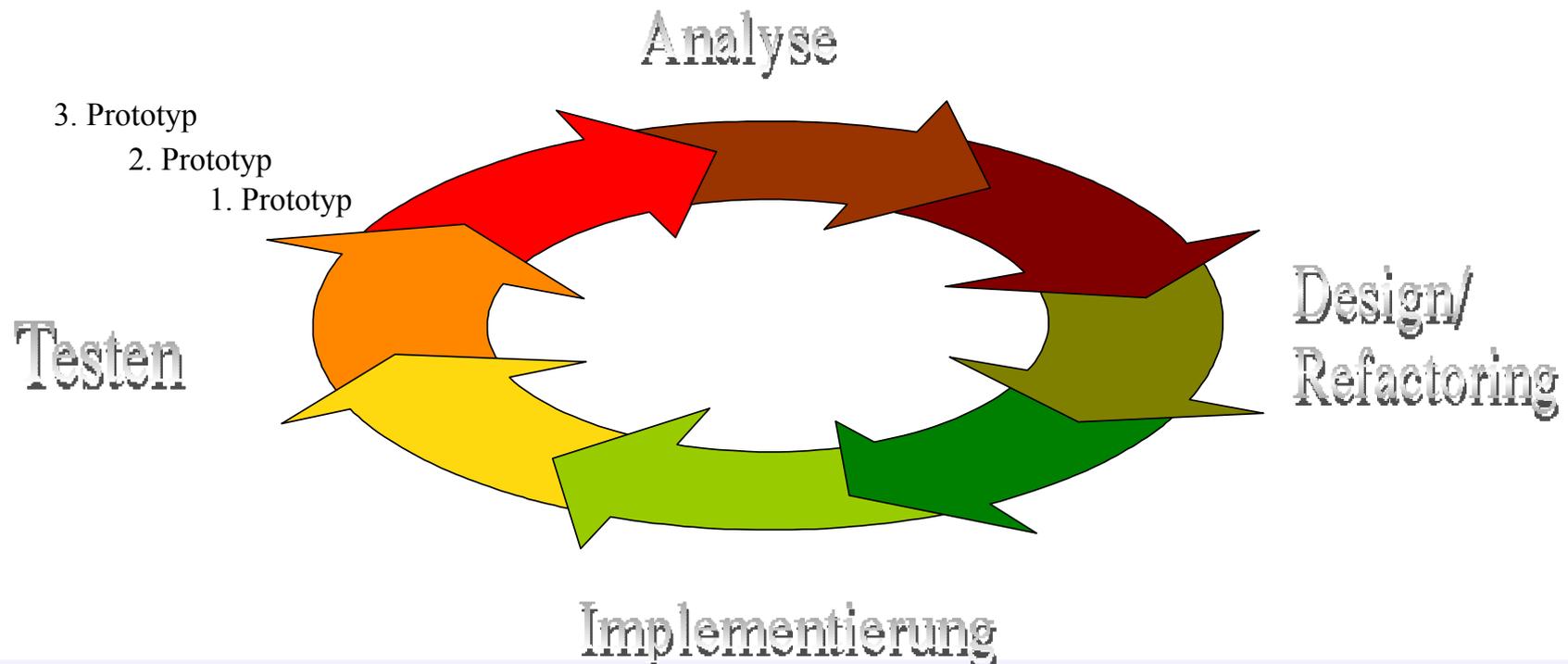
Phasen der SW-Entwicklung

Spiral-Modell

Erfassung der Requirements



Grady Booch: design a little, code a little, test a little...



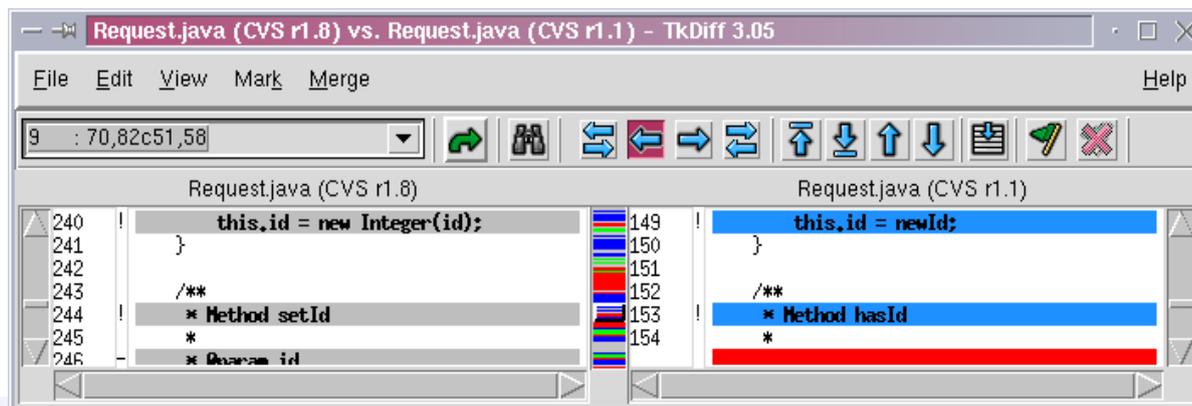
Versionierung

OpenSource

- CVS
 - basiert auf RCS, ausgereift
 - Multi-Site-fähig
 - Basis vieler OSS-Projekte
 - kommando-orientiert
 - graphische Frontends verfügbar

Kommerziell

- ClearCase
 - mächtig + komplex
- PVCS
 - GUI + Kommandozeile
- Perforce
 - performant



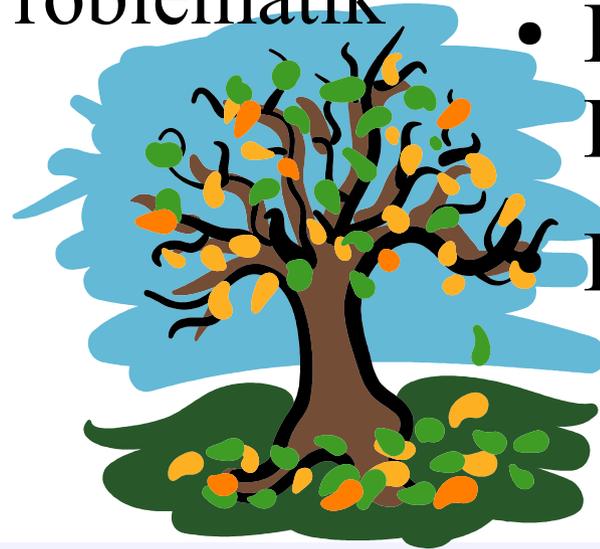
CVS-Alternative

CVS

- keine Versionierung von Verzeichnissen
- Binärdateien = Sonderfall
- Firewall-Problematik

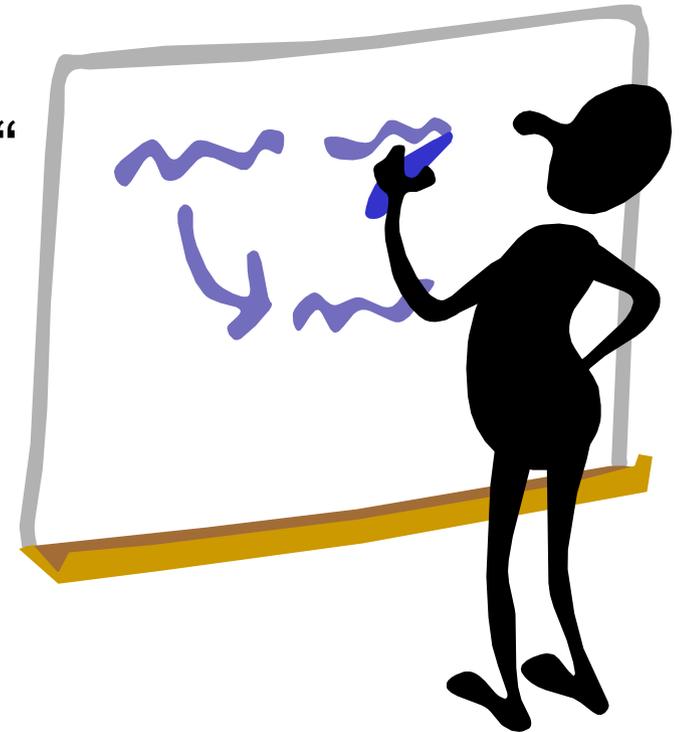
Subversion

- noch Alpha
- Versionierung von Verzeichnissen
- eigene Attribut mögl.
- HTTP zur Kommunikation
- Kommandos CVS-like



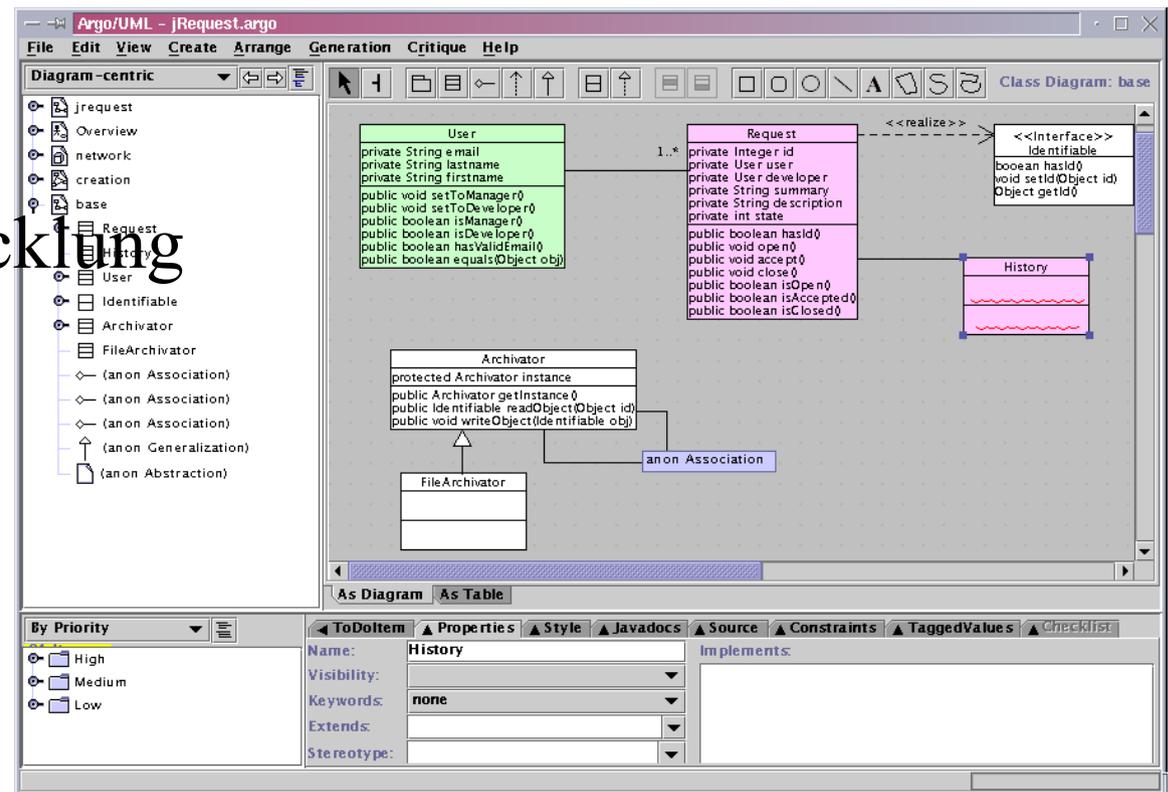
Analyse / Requirement-Erfassung

- DOORS
(<http://www2.telelogic.com/doors/index.cfm>)
 - Erfassung von Requirements
 - für „most popular UNIX platforms“
- Requirement-Tool No. 1: Word
 - Alternative: OpenOffice
 - Ablage wahlweise .doc oder als XML-Datei



Analyse / Design mit UML

- Argo/UML 0.12
 - frei verfügbar
 - ToDo-Liste
 - noch unter Entwicklung

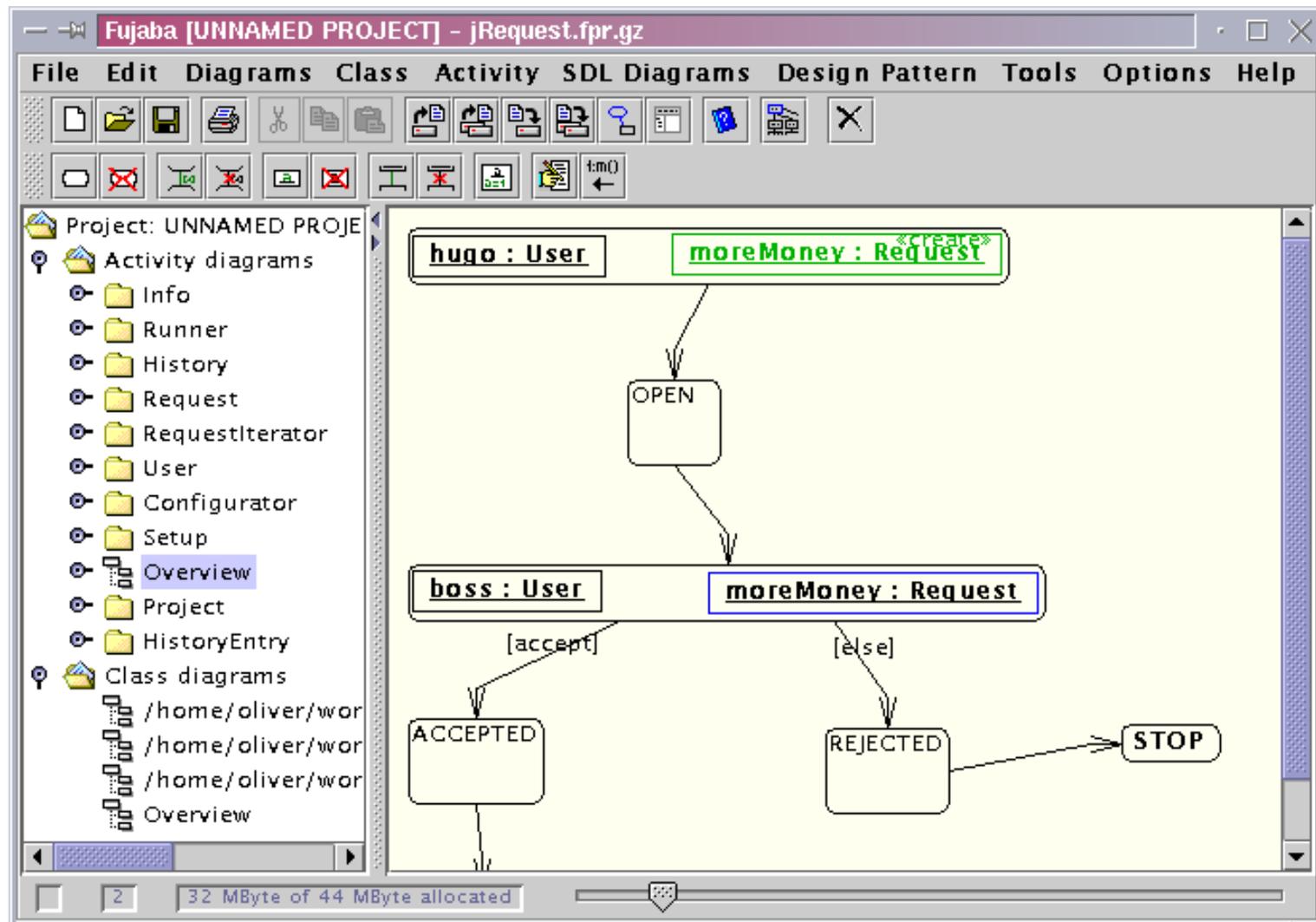


UML-Tools



- FUJABA - From UML to Java And Back Again
 - aus Diplomarbeit an Uni Paderborn entstanden
 - Unterstützung von SDM (Story Driven Modeling)
 - komplett graphische Modellierung
 - sehr gutes Reverse Engineering
 - Schwächen bei der Code-Generierung

FC JABA



kUML

The screenshot shows the kUML 0.5.0 application window. The title bar reads "kUML 0.5.0". The menu bar includes "Project", "Edit", "Diagram", "Setup", and "Help". The toolbar contains icons for file operations, editing, and help. The main workspace displays a UML class diagram with the following elements:

- ClassDiagramViewFactory**: A class with methods `+ClassDiagramViewFactory(Diagram * d)`, `+createView(AssociationRelationship * asso, bool lvi) : DrawingAssociation *`, `+createView(ClassDiagram * cd, bool lvi) : DrawingClassDiagram *`, `+createView(GeneralizationRelationship * gen, bool lvi) : DrawingGeneralization *`, and `+~ClassDiagramViewFactory()`.
- DiagramViewFactory**: A class that inherits from `ClassDiagramViewFactory` (indicated by a solid line with an open triangle arrowhead). It has a protected attribute `#diagram : Diagram *` and methods `+DiagramViewFactory(Diagram * d)`, `+createPrimitiveRectangle(const QString & name, bool lvi) : DrawingRectangle *`, `+createTextnote(const QString & name, bool lvi) : DrawingText *`, `+createTextnoteLink(const QString & name, bool lvi) : DrawingTextnoteLink *`, `+registerView(DrawingItem * view) : void`, and `+~DiagramViewFactory()`.
- UseCaseDiagramViewFactory**: A class that inherits from `DiagramViewFactory`. It has methods `+UseCaseDiagramViewFactory(Diagram * d)`, `+createView(Actor * actor, bool lvi) : DrawingActor *`, `+createView(UseCase * uc, bool lvi) : DrawingUseCase *`, `+createView(UseCaseAssociation * asso, bool lvi) : DrawingUseCaseAssociation *`, `+createView(UseCaseExtendsIncludes * ext, bool lvi) : DrawingUseCaseExtendsIncludes *`, `+createView(UseCaseGeneralization * gen, bool lvi) : DrawingUseCaseGeneralization *`, and `+~UseCaseDiagramViewFactory()`.

On the right side of the diagram, there are two boxes: **Diagram** and **DrawingItem**. An arrow points from **DrawingItem** to **Diagram**. A text box at the top of the diagram area contains the instruction: "Use these factories to create diagram elements instead of creating them yourself. The diagram element will be stored in the diagram and a list view item is created."

The left sidebar shows a project tree with the following structure:

- lines
- factories
 - Textnotes
 - New Textnote 1
 - Associations
 - New Composition 1
 - Diagram -> DiagramViewFactory
 - Generalizations
 - UseCaseDiagramViewFactory
 - ClassDiagramViewFactory ->
 - Classes
 - DrawingItem
 - UseCaseDiagramViewFactory
 - ClassDiagramViewFactory

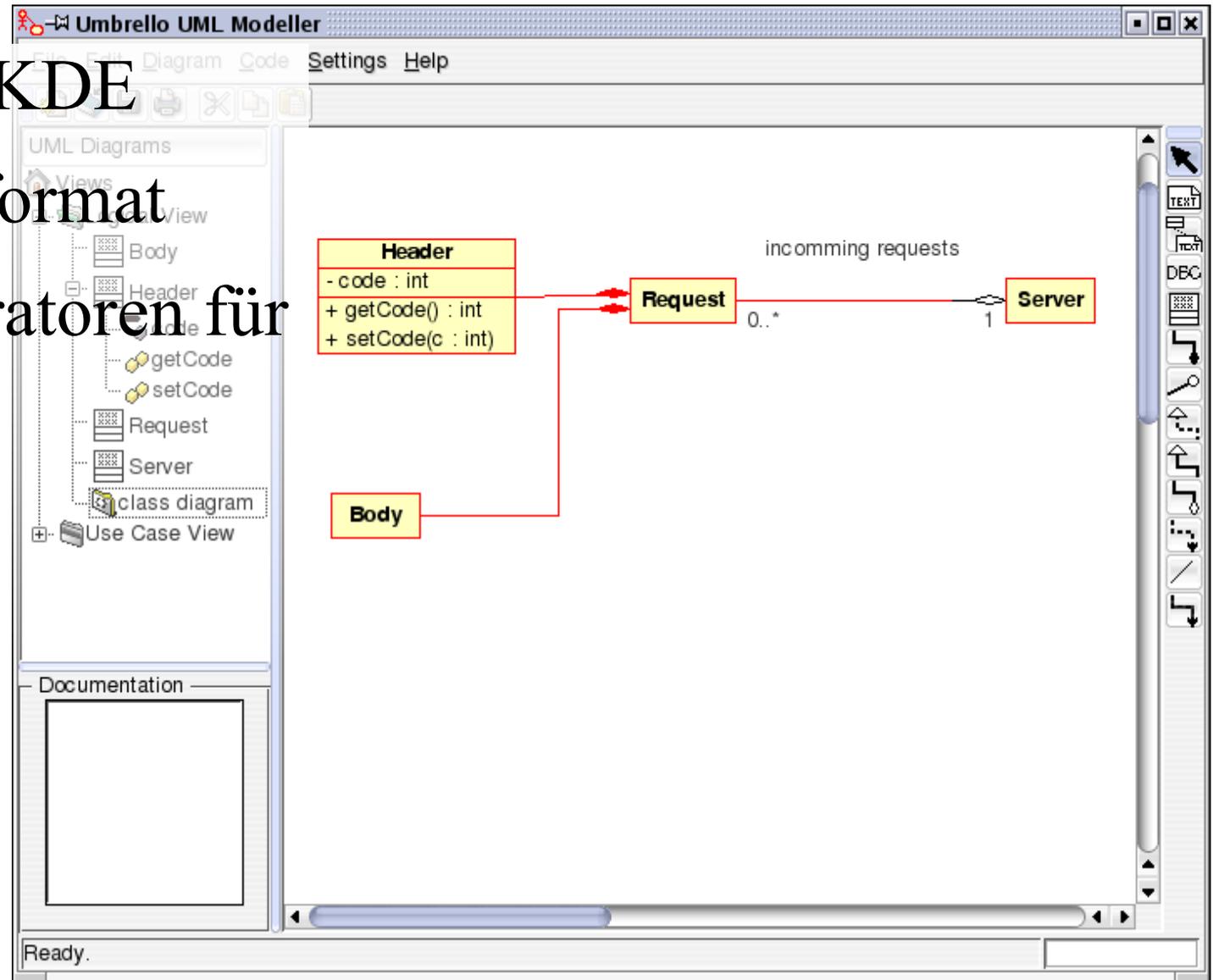
The code editor on the left contains the following text:

```
/** This factory class should be used to create  
 * textnotes or primitive rectangles. These are  
 * kinds of diagrams. Diagram specific views are  
 * by special factories.  
 *  
 * The factory will add the created view to the  
 * list view item for the project browser. The  
 * updated automatically. You have to explicitly  
 * if you want to see the view immediately.  
 *  
 * <pre>  
 * QString viewName;  
 * int counter = 1;
```

**UML-Tool von/für KDE
z.Zt. im Redesign steckengeblieben**

Umbrello UML Modeller

- kommt mit KDE
- XMI-Dateiformat
- Code-Generatoren für
 - Java
 - C++
 - PHP



Kommerz. UML-Tools

- Together/J (<http://www.togethersoft.com/>)
 - intuitiv bedienbar
 - Single-Source-Prinzip
(„der Code ist das Modell“)
 - beliebig konfigurierbar
 - mehrfach ausgezeichnet
 - 100% pure Java



Together

The screenshot displays the Together 4 IDE interface for a project named "jRequest". The left sidebar shows a project tree with folders like "core", "db", "io", "log", "plan", "q", "test", "ui", "util", "jrequest", "network", "public_html", and "UML". The main workspace is divided into two panes. The top pane shows a UML class diagram with the following classes and relationships:

- Project**: Attributes: -name:String, -description:String.
- HistoryEntry**: Attributes: -date:Date, -user:String, -action:int.
- User** (Serializable): Attributes: -name:String, -e mail:String, -type:int. Methods: + User(name:String), + User(name:String, e mail:String).
- Identifiable**: Interface with attribute: -id:int.
- Request**: Implements Identifiable. Attributes: -project:String, -type:String, -idea:String, -state:int=0, -priority:short=0, -difficulty:short=0, -effort:int=-1.
- History** (Serializable): Attributes: -creationTime:Date, -user:User. Method: History(user:User).
- Init**: A small class.

Relationships in the diagram:

- Project (1) is associated with Request (0..*).
- HistoryEntry (1..*) is associated with Request (1).
- Request (1) is associated with History (1).
- HistoryEntry (1..*) inherits from User (1).
- History (1) inherits from User (1).

The bottom pane shows the Java code for `Request.java`:

```
/*
 * @author $Author$
 * @version $Revision$ $Date$
 */
public class Request implements Identifiable {

    private int    id = -1;
    private String project;
    private String type;
    private String idea;
    private int    state;
    // ...
}
```

IDE

- JBuilder
- Eclipse
- NetBeans / Forte
- (X)Emacs
- KDevelop
- Nedit
- vi



JBuilder 8

- Pure Java
- gute Unterstützung bei GUI-Erstellung
- UML-Browsing
- Hot-Swap-Debugging
- Integration von Test-Werkzeugen

JBuilder - /home/oliver/work/jRequest/src/jrequest/base/Request.java

File Edit Search View Project Run Team Wizards Tools Window Help

BaseTest build Request User

jrequest.jpx

- jrequest
 - admin
 - base
 - Attachment.java
 - History.java
 - HistoryEntry.java
 - ObjectIterator.java
 - Project.java

jrequest.base.Request

- Extended Classes
- Implemented Interfaces
- Associations
 - java.lang
 - Object
 - jrequest.base
 - Attachment
 - History
 - Project
 - User
 - org.apache.log4j
 - Dependencies

UML Diagram:

- java.lang: Object
- jrequest.db: Identifiable
- jrequest.base: Request
 - ACCEPTED : int
 - ASSIGNED : int
 - cat : Category
 - CLOSED : int
 - OPEN : int
 - QUERIED : int
 - REJECTED : int
 - attachment : Attachment
 - developer : User
- jrequest.base: User
- jrequest.sample: JavaRulesForm
- jrequest.test: BaseTest, DB

Source Design Bean UML Doc History

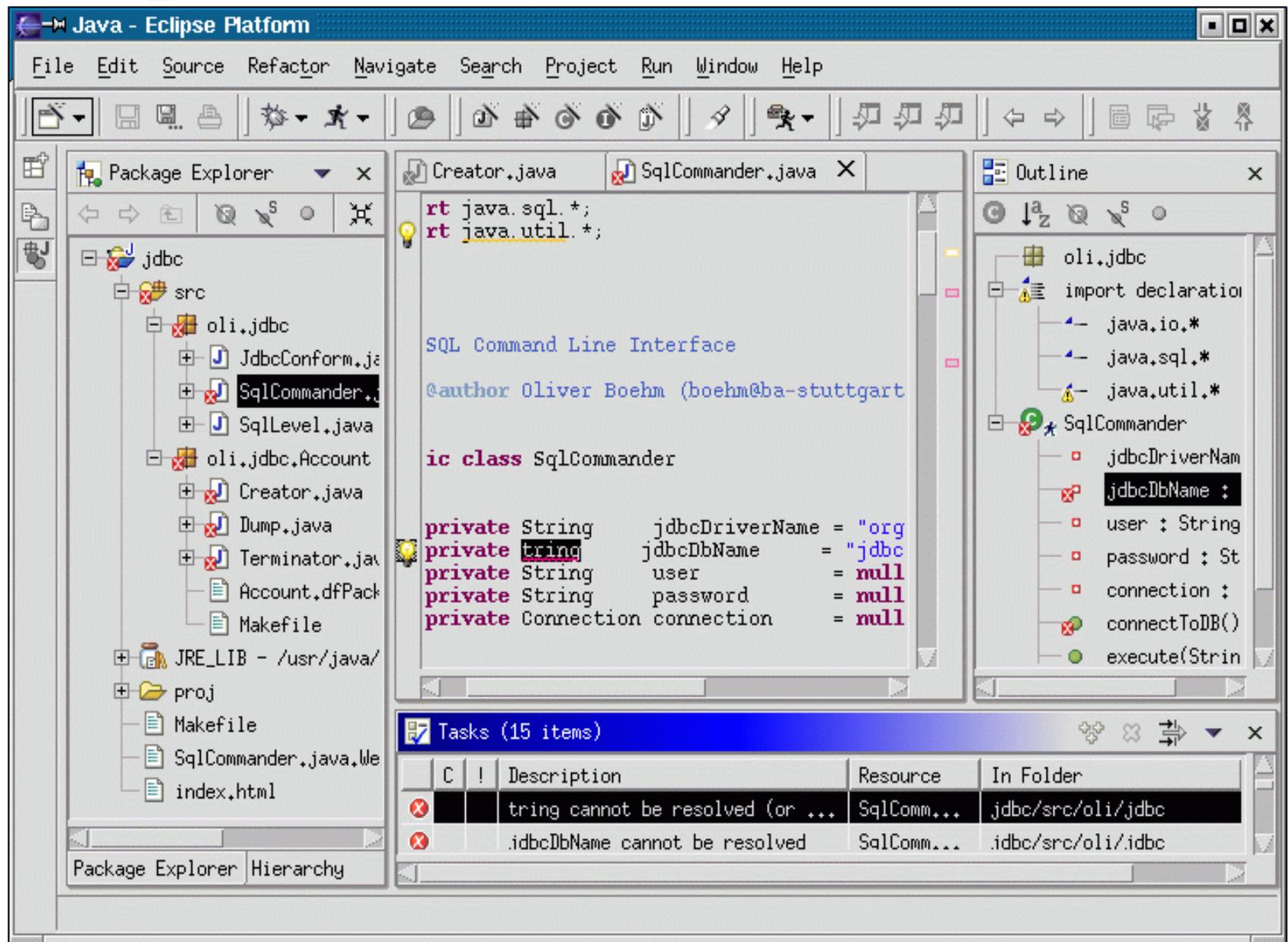
Build completed with 2 warnings. Build took 8 seconds.



Eclipse

- Nachfolger von VisualAge
- Native Look&Feel (SWT)
- inkrementeller Compiler
- Basisplattform für viele Plugins
- gute Refactoring-Unterstützung
- auch für andere Sprachen
 - C++, AspektJ, Cobol, ...

Eclipse



Forte / NetBeans

- **Pure Java**
 - langsam
- **offen / Einbindung anderer Tools**
 - integrierter CVS-Support
 - Ant-Build-Support
 - visuelle GUI-Design-Tools
 - integr. Web-Server für Applets/JSP-Entwicklung
 - Erstellung von Jar-Dateien
- **Forte (Sun)**
 - Schwerpunkt: Stabilität
 - Support von Sun
 - zukünftige Basis für alle Progr.-Sprachen
- **NetBeans**
 - Open Source
 - akt. Entwicklung findet hier statt

(X)Emacs

- sehr mächtig
- hat eine lange Geschichte hinter sich
- hat alles, was eine IDE braucht (u.v.m.)
 - Syntax-Highlighting
 - Build-Support
 - Debugger
 - Navigationsfenster (Meta-X speedbar)
- wird entweder geliebt oder gehasst

emacs: /home/oliver/doc/Kurs/Java4Linux/jRequest/jrequest/core/Request.java

File Edit Mule Apps Options Buffers Tools DelBu Rect Java JDE Help

```

/**
 * Method getId
 *
 * @pre      (hasId() == true)
 * @post     (this.id > 0)
 * @bug      should throw an exception if (hasId() != true)
 */
public int getId() {
    return this.idx;
}

/**
 * Method setId
 *
 * @pre      (hasId() == false)
 * @post     (this.id > 0)
 * @bug      what happens if id is already set?
 */
public void setId (int newId) {
    this.id = newId;
}

public boolean hasId() {
    return (this.id > 0);
}

/**

```

IS08-----XEmacs: Request.java (JDE PenDel Font)-----L104--C15--61%-----

Request.java:104:16:104:21: Error: No field named "idx" was found in type "jrequest/core/Request". However, there is an accessible field "id" whose name closely matches the name "idx".

make: *** [jrequest/core/Request.class] Error 1

Compilation exited abnormally with code 2 at Sun Aug 20 11:15:11

IS08-----XEmacs: *compilation* (Compilation PenDel Font:exit [exit-status 2])

Parsing error messages...done

Speedbar

```

~/work/jRequest/...
 /jrequest/
<+> admin
<-> base
[+] Attachment.java
[+] History.java
[+] HistoryEntry.java
[+] Makefile
[+] Project.java
[-] Request.java !
  {+} Classes
  {+} Constructors
  {+} Variables
  > finish
  > accept
  > assignTo
  > equals
  > equals
  > inState
  > confirm
  > getUser
  > create
  > hasId
  > setId
  > setId
  > setId
  > setId
  > getId
  > setDescription
  > getSummary
  > setSummary
  > setUser
[+] RequestFolder.java
[+] RequestIterato

```

Build-Prozess (Daily Build)

- make
 - Erkennung von Abhängigkeiten
 - weit verbreitet
 - fehleranfällige Syntax
- jmk
 - java-basiertes „make“
 - verbesserte Syntax
 - GUI
- ant
- viele unterstützende Tools
 - sed
 - awk
 - touch
 - diff
 - ... (u.v.m.)



ant

- **OS-unabhängig**
 - läuft unter Linux, Sun, NT, ...
- **Java-Unterstützung**
 - Compiler-Aufruf
 - Javadoc-Generierung
 - Jar
- **Test-Unterstützung**
 - Junit
- **SCM -Unterstützung**
 - CVS
 - ClearCase (ab 1.3)
 - SCM-API
- **Build-Events**
 - eigene Listener möglich, z.B.
 - Mail-Benachrichtigung
 - XML-Log
- **EJB-Unterstützung**
 - War
 - WebLogic
 - TopLink
- **Installations-Unterstützung**
 - copy, ftp

Entorpris Java

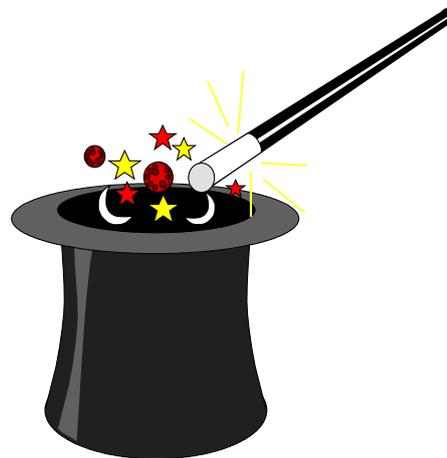
Datenbanken / JDBC

frei verfügbar

- MySQL
- PostgreSQL
- SAP-DB
- Interbase / Firebird
- Hypersonic

kommerziell verfügbar

- IBM DB2
- Oracle
- Sybase



MySQL / PostgreSQL

- Gemeinsamkeiten:
 - in vielen Distributionen enthalten
 - Benutzerberechtigungen mit GRANT
 - Unterstützung von BLOB
 - JDBC-4-Treiber verfügbar

MySQL / PostgreSQL

	MySQL 3.23	PostgreSQL 7.2
Dokumentation	gut	mäßig
Hilfsprogramme	phpAdmin u.v.a.	pgAdmin
SQL-Konform	mit Einschränkungen	weitgehend SQL2
Transaktionen	geplant	read committed, serializable
Foreign Keys, Subselects, Trigger	nein	ja
Geschw. bei wenigen Benutzern	sehr schnell	mittel
Geschw. bei vielen Benutzern	mittel	gut

EJB Application Server

kommerzielle AppServer

- WebSphere (IBM)
 - gute Integration für WSAD
- WebLogic (BEA)
 - performant

freie AppServer

- Jonas
 - OpenSource-Projekt
- jBoss
 - OpenSource-Projekt
 - ehemals EJBoss
 - einfaches Deployment
 - mäßig dokumentiert



jBoss

- sehr einfaches (Hot-)Deployment:
 - EJB ins Deploy-Verzeichnis kopieren
 - fertig
 - auch im laufenden Betrieb möglich
- zusätzliche Bibliotheken sind einfach installiert
 - einfach unter lib/ext kopieren
- schnelles Auffinden von Objekten

Testen

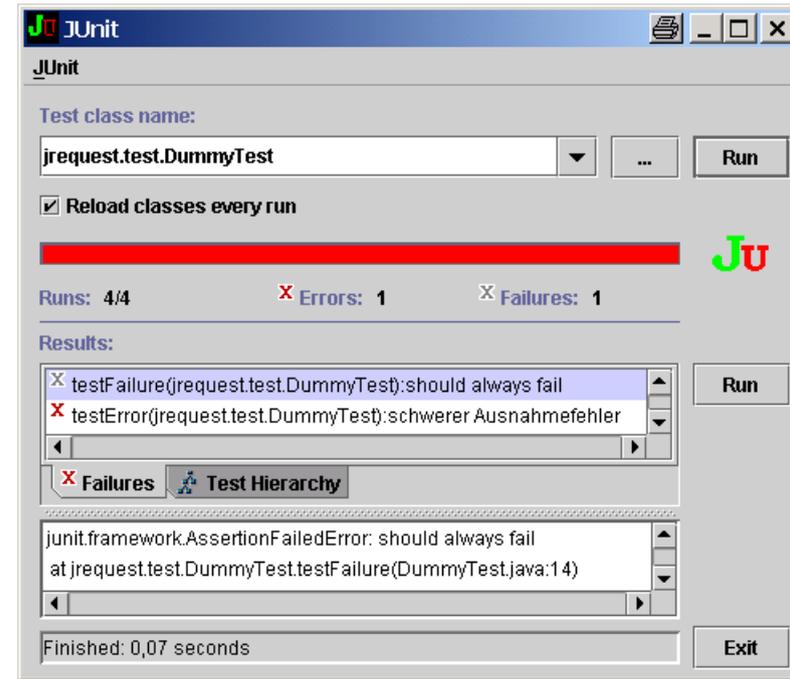


Fehlersuche

- Debugger
 - nicht immer hilfreich
- Logging
 - Log4J (<http://jakarta.apache.org/log4j/>)
- Vorbeugung
 - DbC (Design by Contract)
 - assert (ab JDK 1.4)

Testframeworks

- JUnit
 - reines Java-Testframework
(v. Erich Gamma u. Kent Beck)
- expect
 - Skriptsprache, um Tests zu steuern
- Dejagnu
 - Framework, basierend auf expect



Warum ist Java so langsam?

- Interpretation des Byte-Codes
- Garbage Collection
- Runtime-Checks



Rules of Optimization (M. A. Jackson)

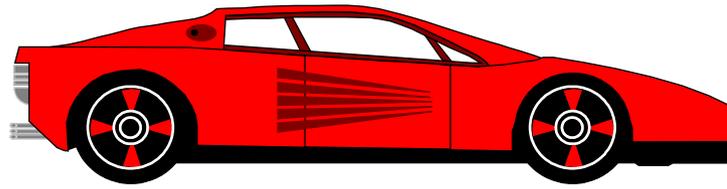
Rule 1:

Don't do it!

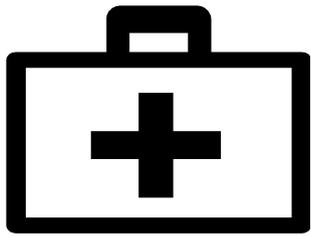
Rule 2: (for experts only)

Don't do it **yet!**

Java-Tuning



- schnellere HW, mehr Speicher
- schnellere JVM
- Heap Size vergrößern (-Xms64m)
- Native Compiler (in Ausnahmefällen)
- C/S: Prozessverlagerung
- Caching
- Objekt-Pools
- ... (Buch-Tipp: Java2 Performance and Idiom Guide v. Craig Larman, Rhett Guthrie)



Installations-Hilfen

- Java Installations-Kit von IBM
 - frei verfügbar und erweiterbar
- InstallShield
 - kommerziell mit großen Funktionsumfang
 - Dokumentation verbesserungswürdig
- RPM
 - unter Linux weit verbreitet
- Java WebStart
 - Programmstart über Web-Browser

Ausblick



Ein Blick in die Zukunft



- Java
 - Generics (JDK 1.5)
 - gemeinsame Schnittstelle für gemeinsame Lösungsansätze
 - verstärkter Einzug in Embedded
 - Handy, PDAs, ...
- Konkurrenz durch .NET
- Linux
 - Server etabliert
 - weitere Unterstützung
 - IBM, HP, Sun, ...
 - 2003 = Jahr des Linux-Desktops?
 - verstärkter Einzug in Embedded

Erfahrungswerte

- ◆ kein Moorhuhn
- ◆ geringeres kommerz. SW-Angebot
- ◆ kein Lotus-Notes
- ◆ mehr Handarbeit und Hintergrundwissen gefragt
- ◆ stabil
 - ◆ auch unter Last
- ◆ effektivere Oberfläche
 - ◆ CLI
 - ◆ besseres GUI
- ◆ „Pipes & Filters“
 - ◆ viele kombinierbare Tools
- ◆ transparenter

Wichtige Termine

- JFS 2003 (Java Forum Stuttgart)
3. Juli 2003

<http://www.jfs2003.de>

JUGS
java user group stuttgart

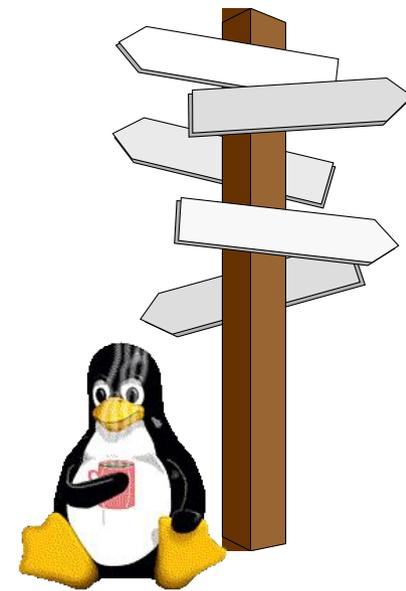


- **JFS 2010**

1. Juli 2010

<http://www.java-forum-stuttgart.de/>

Links



- Java unter Linux

<http://www.2xp.de/J4L/>

- Java Programming on Linux (v. Nathan Meyers)

<http://www.javainux.net/>

- Blackdown.org

<http://www.the-love-shack.net/2007/08/30/blackdown-java-retires/>

<http://www.blackdown.org>

- DeveloperWorks

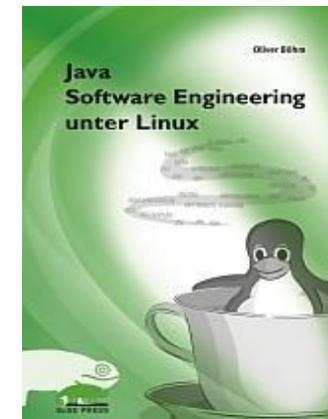
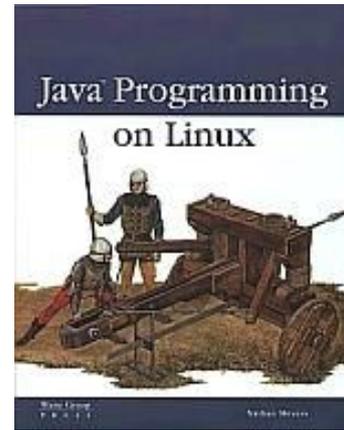
<http://oss.software.ibm.com/developerworks/opensource/>

- Alphaworks

<http://www.alphaworks.ibm.com/>

Bücher

- Nathan Meyers:
Java Programming on Linux
Waite Group Press 1999
- Oliver Böhm:
Java SW-Engineering unter Linux
SuSE-PRESS 2002





oliver.boehm@agentes.de
<http://oli.blogger.de>