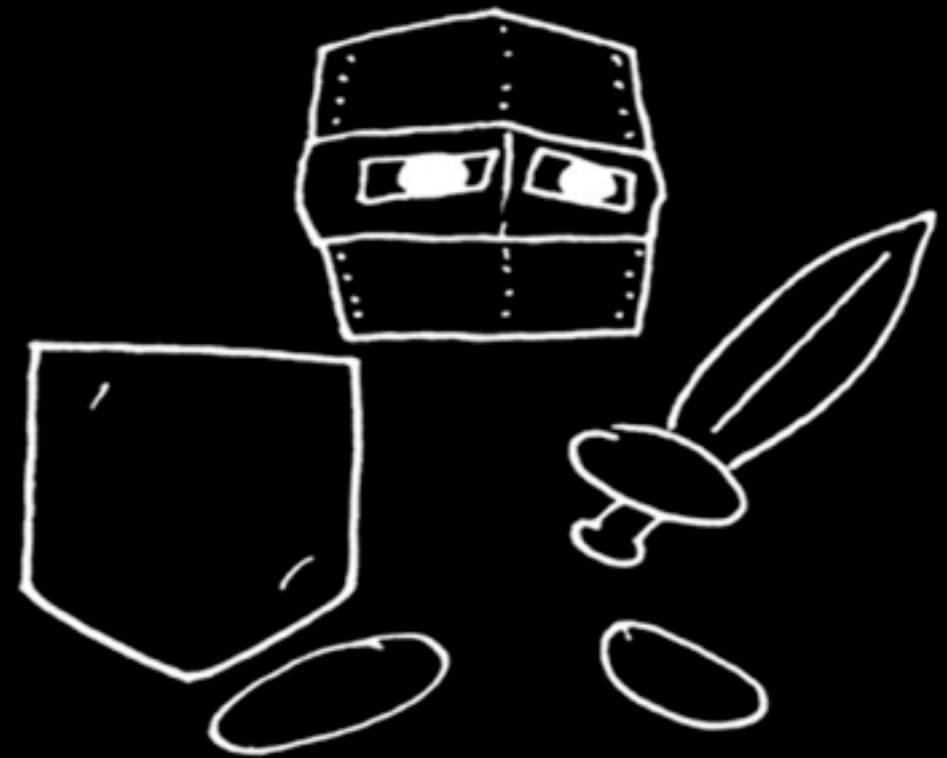
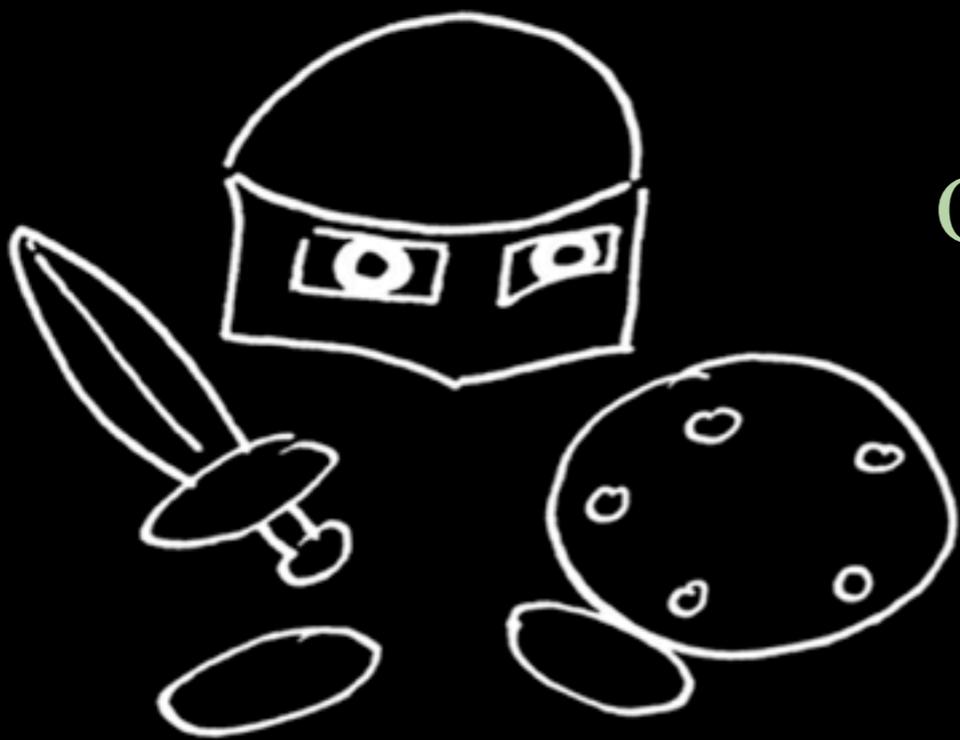


# Java 7

Oliver Szymanski



**SOURCE-KNIGHTS.COM**



- Ende 2006: Java 6

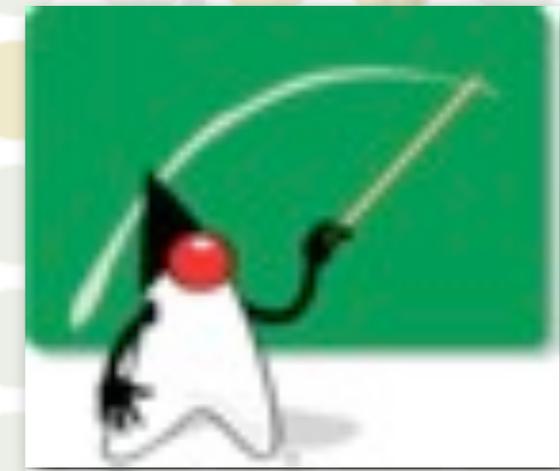
- 2010: Java 7 Dolphin  
Entwicklung im Rahmen des OpenJDK

- Bessere Softwareprodukte?



- OpenJDK-Community  
Entwickler auch außerhalb von Sun  
Quelloffene JDK-Implementierung
- GPLv2 mit Classpath-Ausnahmen  
Sichert, dass Verbesserungen zurückfließen
- Strikte Einreichung  
Mindestens 2 Sun-Mitarbeitern prüfen
- Bessere Softwareprodukte?

# OpenJDK





Copyright © 2010 Source-Knights.com

- Java ist nicht nur eine Sprache
- Java als Technologie-Plattform  
JVM, Sprache, APIs, Programmiermodell



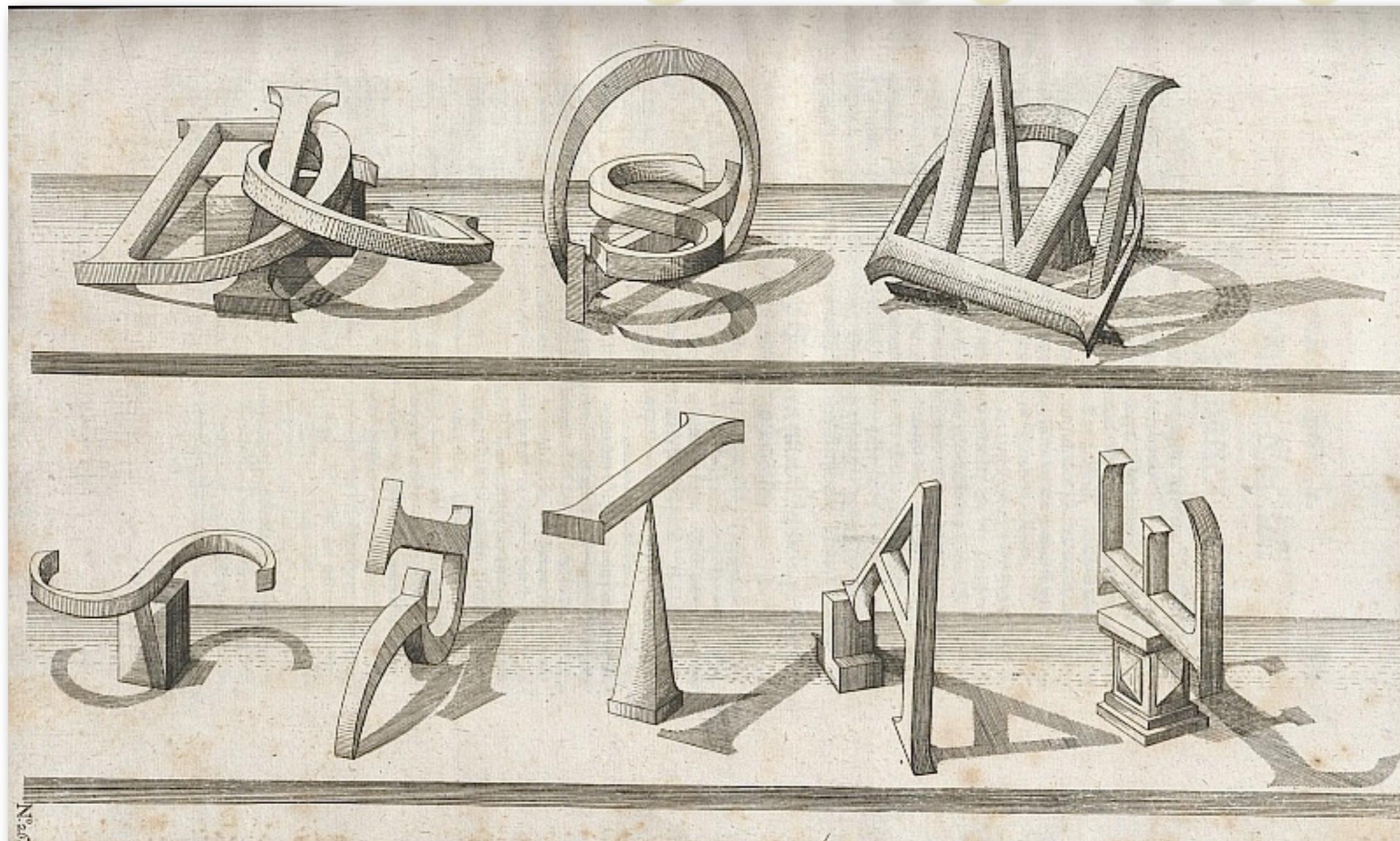


- Wissen zu Java 7

- Nutzen

- Orakeln





Quelle: Deutsche Fotothek

Quelle: Deutsche Fotothek

- Spracherweiterungen
  - JSR 294: Super-Packages

Pakete und Klassen, *public* Inhalte

```
src/com/foo/super-package.java:  
superpackage com.foo {  
    member package  
        com.foo.api,  
        com.foo.model,  
        com.foo.util;  
    export com.foo.api; // public  
}
```

```
export com.foo.api; // public  
com.foo.util;
```

- **Spracherweiterungen**

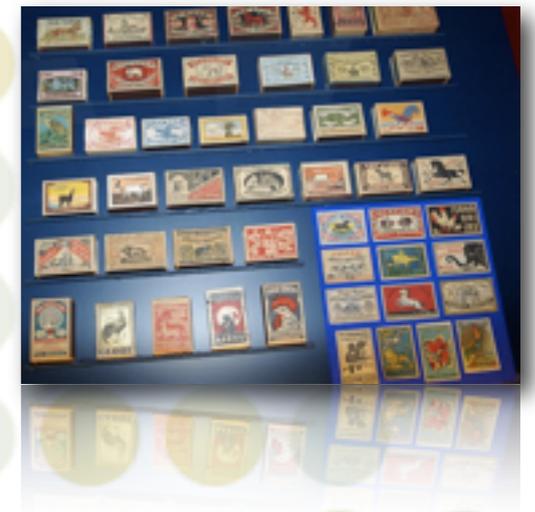
- JSR 277: Module

fehlende Versionierung von JARs

Abhängigkeiten

Inhalt-Exportlisten

=> Distributions und Deployment, Java-Plattform in Modulen



- **Spracherweiterungen**

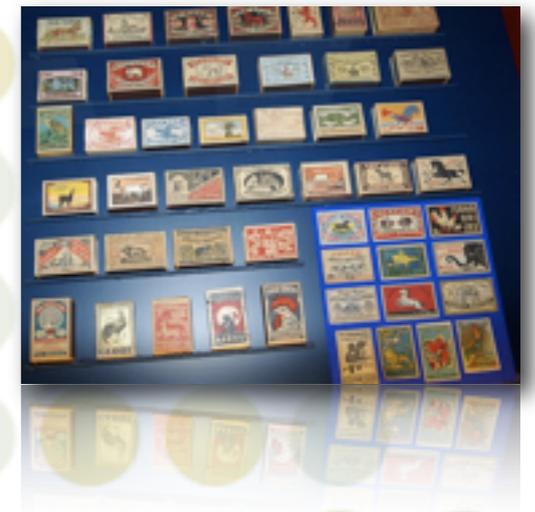
- JSR 277: Module

JAM-Modul-System zu komplex  
Nicht integriert in Java 7

=> Projekt Jigsaw

Nicht offizieller Bestandteil, aber Unterstützung durch Sun

evtl. OSGI-Framework in Zukunft



- Spracherweiterungen
  - JSR 308: Annotation-Erweiterung

Annotationen als Metadaten auf **Typen**

Semantik Kompilier- / Lade- / Laufzeit nicht geändert

```
@MyUnmodifiableDataContainer  
@MyReturnsNonNullNever  
class UnmodifiableList implements @ReadOnly  
List<@ReadOnly T>  
    throws @Critical SomethingBadHappenedException { ... }
```

```
throws @Critical SomethingBadHappenedException { ... }  
List<@ReadOnly T>
```

- **Spracherweiterungen**

- Project Lambda: Lambda-Ausdrücke (Closures) für Java

### **Funktions-Typen, First-Class-Functions**

Erleichtern Angabe anonymer Methoden, Vermeidung Callback-Interfaces

```
#(int x, int y) ( x*y);
```

```
#(int x, int y) {  
  if (x < y ) return x;  
  else return y;  
}
```

```
int(int, int) multiplier = #(int x, int y) ( x * y);  
int result = multiplier(7, 2);
```

```
int result = multiplier(7, 2);  
int(int, int) multiplier = #(int x, int y) ( x * y);
```



- **Spracherweiterungen**

- **Projekt Coin**

OpenJDK-Community Projekt für leichtgewichtige Sprachanpassungen

Sammlung von Vorschlägen und Entscheidung

- Spracherweiterungen
  - Projekt Coin: String-Switches

lesbarer Code

Gefahr auf performantere Alternativen zu verzichten



- **Spracherweiterungen**

- **Projekt Coin: Ressourcen-Management**

Erweiterung des *try*-Statements zur Ressourcen-Deklaration

```
static String readFirstLineFromFile(String path) throws IOException {  
  
    try (BufferedReader br = new BufferedReader(  
        new FileReader(path)) {  
  
        return br.readLine();  
  
    }  
}
```

- **Spracherweiterungen**

- Projekt Coin: Generische Instanziierung

Verkürzte Schreibweise bei generischen Typen

```
Map<String, List<String>> department2Names =  
    new HashMap<String, List<String>>();
```

// verkürzt geschrieben:

```
Map<String, List<String>> department2Names = new HashMap<>();
```

```
Map<String, List<String>> department2Names = new HashMap<>();
```

- Spracherweiterungen
  - Projekt Coin: Unsafe-Warnung

Bei Methoden-Deklaration statt Aufruf

```
static <T> List<T> asList(T... elements) { ... }

static List<Callable<String>> stringFactories() {
    Callable<String> a, b, c;
    ...
    /* Warning: uses unchecked or unsafe operations */
    return asList(a, b, c);
}
```

- **Spracherweiterungen**

- Projekt Coin: Unsafe-Warnung

Bei Methoden-Deklaration statt Aufruf

```
/* Warning: enables unsafe generic array creation */  
static <T> List<T> asList(T... elements) { ... }
```

```
static List<Callable<String>> stringFactories() {  
    Callable<String> a, b, c;  
    ...  
    return asList(a, b, c);  
}
```

```
return asList(a, b, c);  
...
```

- **Spracherweiterungen**

- Project Coin: **mehrere** Exception-Typen in catch

**final** catch-Parameter:

rethrow => nur Typen aus Try-Block, Subtype aus catch und nicht in vorherigen catches verarbeitet

```
try {  
    ...  
}  
catch (final IOException | DBException ex) {  
    logger.log(ex);  
}  
}
```

```
logger.log(ex);
```

- Weitere Spracherweiterungen Coin

- Unterstriche bei Literalen erlaubt

Binärmasken bei Literalen

Mehr Unterstützung bei Collections

```
long cardNo = 1234_5678_9012_3456L;  
int bin = 0b101;
```

```
final List<Integer> digits = [3, 1, 5, 9];  
Map<Integer, String> m = ...  
m[0] = "hello world";
```

```
m[0] = "hello world";  
Map<Integer, String> m = ...
```



Copyright © 2010 Source-Knights.com

- API-Erweiterungen

- JSR 203: Vervollständigung von NIO

Service Provider Interface für beliebige Dateisysteme

Asynchroner Zugriff Sockets, Dateien, Attribute, File-System-spezifische Daten

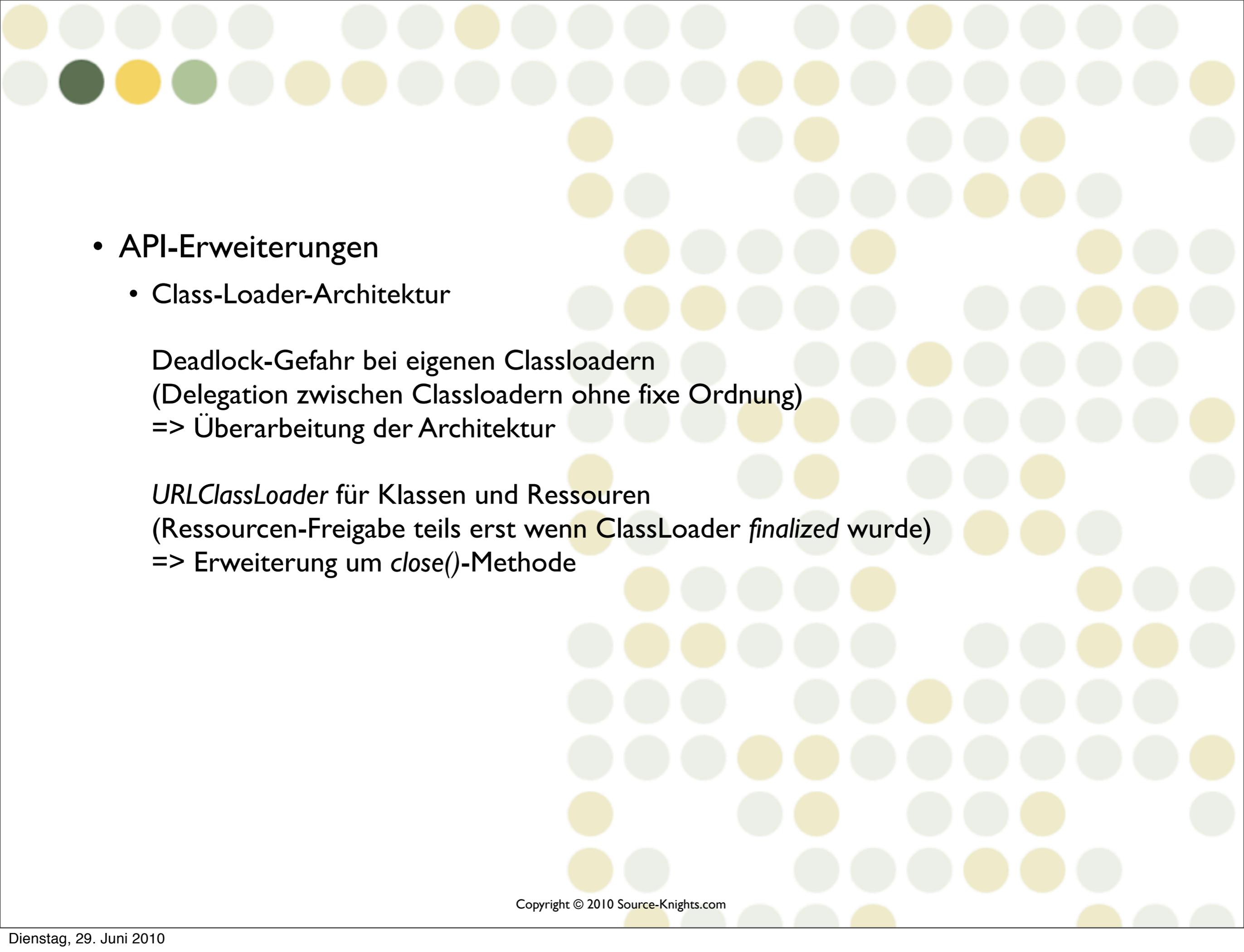
- JSR 51: Mehr NIO

Multicast-Datagramme und Binding

- JSR 166y: ForkJoin-Task-Framework

Leichtgewichtiges Concurrency Framework (Doug Lea)





- API-Erweiterungen

- Class-Loader-Architektur

Deadlock-Gefahr bei eigenen Classloadern  
(Delegation zwischen Classloadern ohne fixe Ordnung)  
=> Überarbeitung der Architektur

*URLClassLoader* für Klassen und Ressourcen  
(Ressourcen-Freigabe teils erst wenn *ClassLoader finalized* wurde)  
=> Erweiterung um *close()*-Methode

- API-Erweiterungen

- JSR 296: Swing Application Framework

Infrastruktur für Swing-Anwendungen  
(Lebenszyklus, Tasks, Sessions, u.a.)

Nicht Teil von Java 7, inaktiver JSR





Copyright © 2010 Source-Knights.com

- Anpassungen VM

- JSR 292: Supporting Dynamically Typed Languages

Bessere Unterstützung für Skriptsprachen  
Erweiterung Byte-Code um *invoke-dynamic*  
(dynamisch typisierte Sprachen Ruby, Java-Script, ...)

Struktur von Klassen zur Laufzeit modifizieren

```
Object x = Dynamic.getMeSomething();
```

```
Dynamic y = ...;  
y.doAnything("with any parameters");
```

- Anpassungen VM
  - Beliebige Ausdrücke als Identifier

```
int #"answer to life, the universe and everything" = 42;  
  
System.out.println(  
    #"answer to life, the universe and everything");  
  
// prints 42
```

```
\\ string \\
```

- Anpassungen VM

- Komprimierungstechnik für 64-Bit Pointer

Passen dann in 32 Bit

Senkt Speicherplatzbedarf und schont Bandbreite



- Anpassungen VM

- Garbage First (G1)

Seit Java 6 Update 14 bereits verfügbar  
(experimentell)

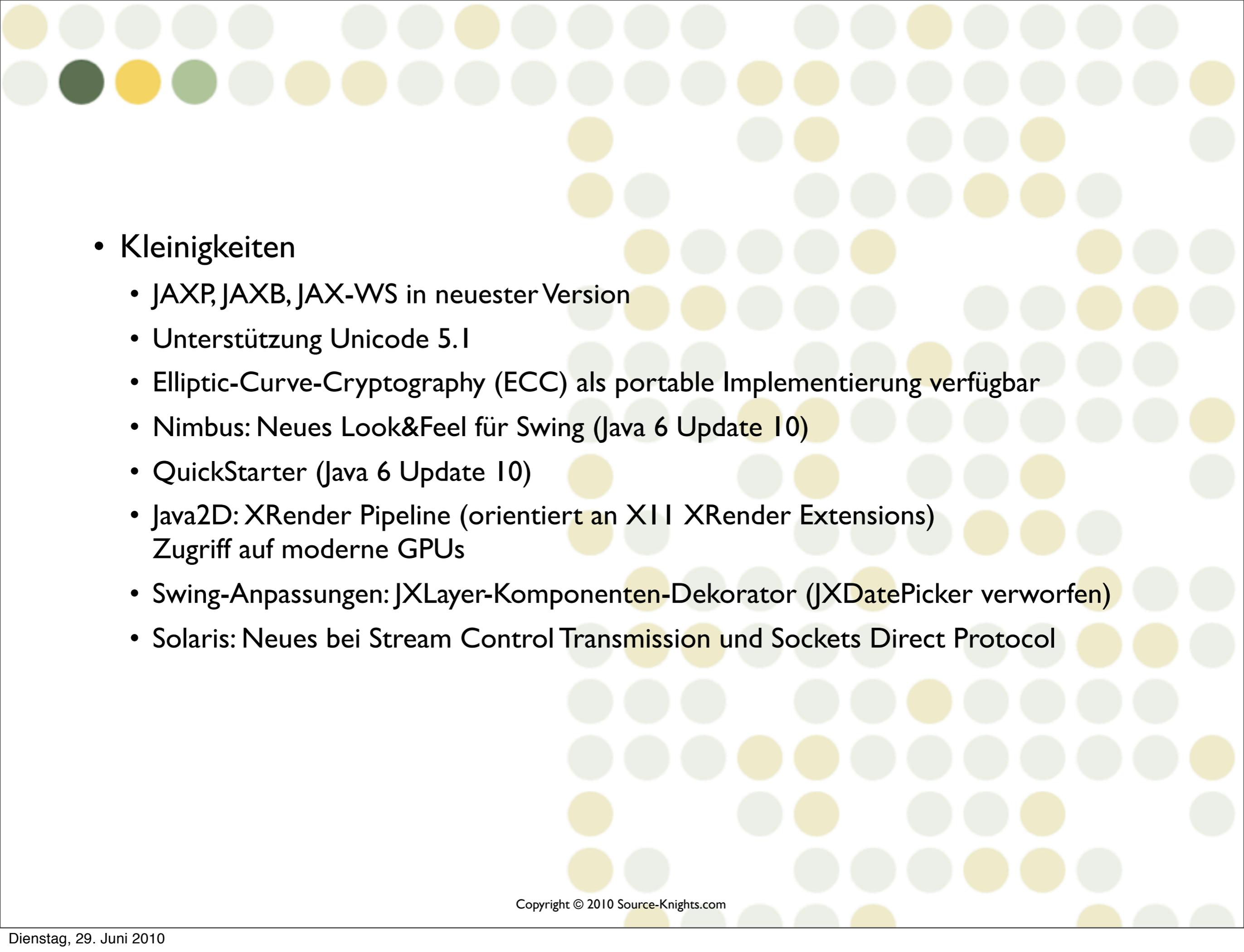
Ersetzt Concurrency Mark-Sweep

Kürzere Wartezeiten



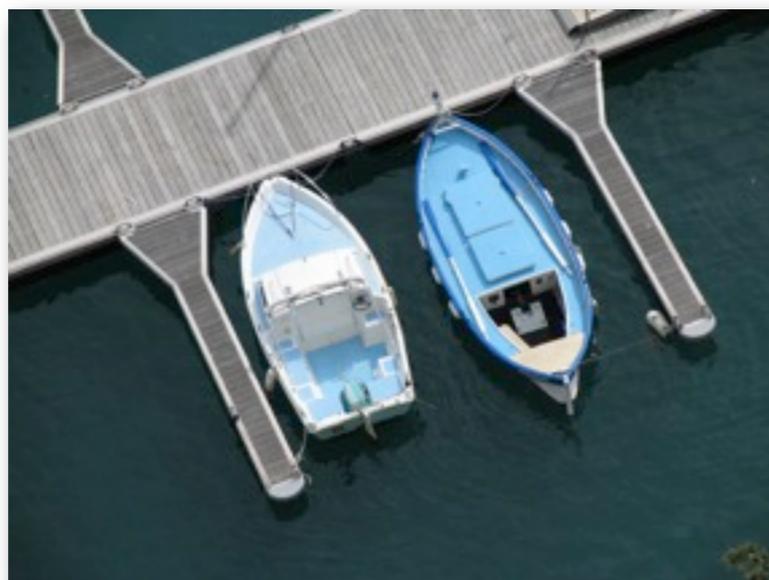


Copyright © 2010 Source-Knights.com



- Kleinigkeiten

- JAXP, JAXB, JAX-WS in neuester Version
- Unterstützung Unicode 5.1
- Elliptic-Curve-Cryptography (ECC) als portable Implementierung verfügbar
- Nimbus: Neues Look&Feel für Swing (Java 6 Update 10)
- QuickStarter (Java 6 Update 10)
- Java2D: XRender Pipeline (orientiert an X11 XRender Extensions)  
Zugriff auf moderne GPUs
- Swing-Anpassungen: JXLayer-Komponenten-Dekorator (JXDatePicker verworfen)
- Solaris: Neues bei Stream Control Transmission und Sockets Direct Protocol

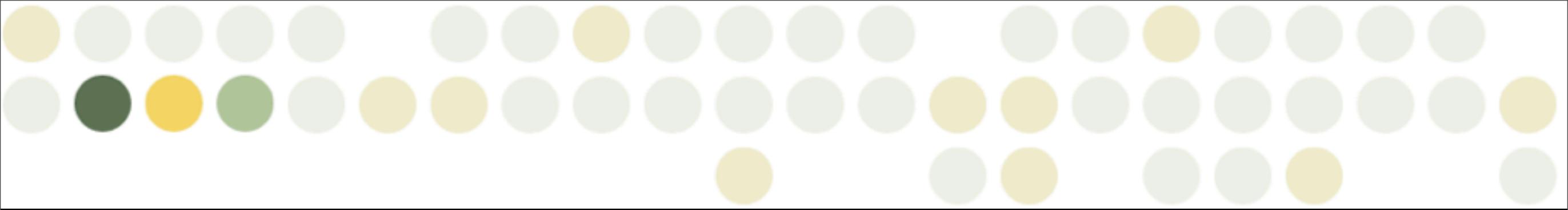


- Wir müssen weiterhin unsere Probleme selbst lösen



- Oder im Team ;)





# Fragen? Fragen!

[oliver.szymanski@source-knights.com](mailto:oliver.szymanski@source-knights.com)



**SOURCE-KNIGHTS.COM**