

# Java 7: Was ist neu

---

21 Mai 2012

**Genc Mazlami**  
**Gionata Genazzi**  
Luca Longinotti

# Java 7: Übersicht

---

- Erschienen 06.2011 (aktuell update 4)
- Linux, MacOS X, Solaris und Windows
- **Neue Sprach-Funktionalität (Project Coin)**
- Neue I/O API
- Neue multicore-ready API
- Unterstützung von dynamischen Sprachen

# Switch auf Strings

---

```
final String str = "Java7";  
switch (str) {  
    case "Java6":  
        System.out.println("You couldn't do this!");  
        break;  
    case "Java7":  
        System.out.println("Now you can!");  
        break;  
}
```

# Binäre Literale

---

Java 6:

```
int binary = 8;
```

Java 7:

```
int binary = 0b1000;
```

# Unterschiedliche Numerische Literale

---

```
// New in Java7
```

```
int oneMillion_ = 1_000_000;
```

```
int oneMillion = 1000000;
```

# Diamant Syntax

---

Java 6:

```
Map<String, Integer> map =  
    new TreeMap<String, Integer>();
```

Java 7:

```
Map<String, Integer> map =  
    new TreeMap<>();
```

# Multi-Catch bei Exceptions

---

```
try {  
    fileInStream.read(fileContent);  
}  
catch (FileNotFoundException | IOException e) {  
    e.printStackTrace();  
}
```

# Automatisches Ressourcen Management

---

```
try (  
    OutputStream fos = new FileOutputStream("j7.txt");  
    OutputStream dos = new DataOutputStream(fos)  
) {  
    dos.write("Java7 cleans up after itself!");  
}  
catch (IOException e) {  
    e.printStackTrace();  
}
```



# IDE Status

---

- Eclipse: ab Indigo SR 1 Release (09.2011)
- Netbeans: ab 7.0.1 (08.2011)
- IntelliJ IDEA: ab 10.5 (05.2011)

# Java 7

---

- **Neue Sprach-Funktionalität**
  - **Switch auf Strings**
  - **Binäre Literale**
  - **Unterschrichene Numerische Literale**
  - **Diamant Syntax**
  - **Multi-Catch bei Exceptions**
  - **Automatisches Ressourcen Management**
- **Neue I/O API**
- **Neue multicore-ready API**
- **Unterstützung von dynamischen Sprachen**

# Neue I/O API (NIO 2.0)

---

- Verbesserte Pfad Handhabung (Path & FileSystem Klassen)
- Neue File Operationen (copy, move, delete)
- Verbesserte Unterstützung von Sym. Links
- Asynchrones I/O (via Futures & Callbacks)
- File Change Notification (WatchService API)

# Neue multicore-ready API

---

- Fork/Join Framework
  - Parallele Abarbeitung von Tasks
- Synchronisations Barrieren (Phaser)
- Thread-Lokale Zufällige Zahlen
- Verbesserte/neue Concurrent-Collections

# Weitere Infos

---

[https://blogs.oracle.com/javase/entry/java\\_7\\_has\\_released](https://blogs.oracle.com/javase/entry/java_7_has_released)

<http://openjdk.java.net/projects/jdk7/features/>

[http://www.eclipse.org/jdt/ui/r3\\_8/Java7news/whats-new-java-7.html](http://www.eclipse.org/jdt/ui/r3_8/Java7news/whats-new-java-7.html)