

# Modularisierung in Java mit Jigsaw

Henning Hontheim

# Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

MODULEPATH

JAR Module

Weiterführendes

# Was ändert sich mit Java 9?

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

`MODULEPATH`

JAR Module

Weiterführendes

- Verzeichnisstruktur hat sich geändert
- JRE nicht mehr in eigenem Verzeichnis des JDK
- Java-Libs `rt.jar` und `tools.jar` gibt es nicht mehr
- Einige interne APIs sind nicht mehr zugreifbar (z.B. einige `com.sun.*`- und `sun.misc.*`-Packages)
- Reflection ist eingeschränkt
- Linker `jlink`, der Distributionen erstellt, die ohne zusätzliches Java ausführbar sind

# Jigsaw

## Das Java Platform Module System (JPMS)

- Skalierbarkeit von Java (z. B. für IoT)
  - Erhöhung der Sicherheit und Wartbarkeit von Java
  - Verbesserte Performance
  - Förderung gut konstruierter und wartbarer Bibliotheken und Anwendungen
  - **Explizite Abhängigkeitskontrolle** (Definition lose gekoppelter Services und zuverlässige Konfiguration)
  - Einfache Handhabbarkeit
- 
- 1 Die Modularisierung von Java selbst (mit „Platform Modules“)
  - 2 Die Modularisierung von Bibliotheken und Anwendungen (mit „Application Modules“)

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

`MODULEPATH`

JAR Module

Weiterführendes

# Unterschiede zu OSGi

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

`MODULEPATH`

JAR Module

Weiterführendes

- OSGi unterstützt Versionierung
- OSGi unterstützt *Hot Deployment*, also Austausch der Module zur Laufzeit

**Jigsaw unterstützt beides nicht!**

# Verschiedene Arten von Modulen

- **Platform Explicit Modules:** *Platform Modules*  
Module der Java Runtime (siehe `$ java --list-modules`)
- **Application Explicit Modules**  
JARs, die einen *Module Descriptor* (`module-info.class`) enthalten
- **Open Modules**  
Zusätzlich zu *Application Explicit Modules*: Export der Module zur Laufzeit für *Deep Reflection*
- **Automatic Modules**  
JARs, die keinen *Module Descriptor* enthalten
- **Unnamed Modules**  
Alle JARs des CLASSPATH werden zu genau einem (!) unbenannten Modul zusammengesasst. Diese werden (selbst mit *Module Descriptor*) als *Nicht-Jigsaw-Module* behandelt.

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

`MODULEPATH`

JAR Module

Weiterführendes

# Die module-info.java-Datei

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

module-info.java

Beispiele

MODULEPATH

JAR Module

Weiterführendes

```
1 module org.example.app {
2
3     // Abhaengigkeiten zu anderen Modulen
4     requires org.example.util;
5     requires transitive com.company.service; //
        Transitive Abhaengigkeit
6
7     // Exportieren eigener Pakete
8     exports org.example.app;
9     exports org.example.secure to com.company.access; //
        Exklusive Freigabe
10
11 }
```

# Beispiele

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

**Beispiele**

MODULEPATH

JAR Module

Weiterführendes

- 1 Manuelle Verlinkung über den MODULEPATH
- 2 Manuelle Verlinkung als einzelne JAR Module



# Manuelle Verlinkung über den MODULEPATH

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

module-info.java

Beispiele

**MODULEPATH**

JAR Module

Weiterführendes

```
1 #!/bin/bash
2
3 javac -d classes --module-source-path src $(find src -name "*.
   java")
4
5 java -p classes -m net.hontheim.jigsaw.meineapp/net.hontheim.
   jigsaw.meineapp.MeineApp
```

# Manuelle Verlinkung als einzelne JAR Module

```
1 #!/bin/bash
2
3 javac -d classes --module-source-path src $(find src -name "*.
   java")
4
5 mkdir ./modules
6
7 jar --create --file modules/net.hontheim.jigsaw.meineapp-1.0.
   jar --module-version 1.0 --main-class net.hontheim.jigsaw.
   meineapp.MeineApp -C classes/net.hontheim.jigsaw.meineapp .
8
9 jar --create --file modules/net.hontheim.jigsaw.meinutil-1.0.
   jar --module-version 1.0 -C classes/net.hontheim.jigsaw.
   meinutil .
10
11 java -p modules -m net.hontheim.jigsaw.meineapp/net.hontheim.
   jigsaw.meineapp.MeineApp
```

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

module-info.java

Beispiele

MODULEPATH

JAR Module

Weiterführendes

# Manuelle Verlinkung als einzelne JAR Module

Analyse der Dependencies mit jdeps

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

module-info.java

Beispiele

MODULEPATH

JAR Module

Weiterführendes

```
1 | $ jdeps -s --module-path modules modules/*.jar
2 | net.hontheim.jigsaw.meineapp -> java.base
3 | net.hontheim.jigsaw.meineapp -> net.hontheim.jigsaw.meinutil
4 | net.hontheim.jigsaw.meinutil -> java.base
```

# Weitere Informationen

Intro

Inhalt

Java 9+

Jigsaw

OSGi

Modularten

`module-info.java`

Beispiele

`MODULEPATH`

JAR Module

Weiterführendes

Das Java Platform Module System (JPMS, "Jigsaw")

<https://www.torsten-horn.de/techdocs/Jigsaw.html>