

Gradle, ein Buildsystem

Rahel Arnold <rahel.arnold@unibas.ch>

Cristina Illi <cristina.illi@unibas.ch>

Nikodem Kernbach <nikodem.kernbach@unibas.ch>

Simon Peterhans <simon.peterhans@unibas.ch>

Jan Schönholz <jan.schoenholz@unibas.ch>

Departement Mathematik & Informatik, Universität Basel

Programmierprojekt – FS20

Lernziele

- Was ist ein Buildsystem und wofür brauchen wir es?
- Aufbau von Gradle
- Wichtigste Befehle von Gradle

Warum brauchen wir ein Build System?

```
> javac -cp libs/hadoop-core-1.2.1.jar:libs/guava-2.1.jar *.java &&  
jar cmf my_manifest project *.class
```

Was macht obiger Befehl?

Was sind die Schwächen?

Manuelle Builds: Schwächen

- › Unübersichtlich
- › Systempfadabhängig
- › javac hat eine unübersichtliche Anzahl an Optionen
- › Wer ruft den Build Befehl auf?
- › Manuelles Verwalten von verwendeten Libraries
- › Befehlskette für Kompilation, JAR-Erstellung, JavaDoc-Generation, ...

Was kann IntelliJ?

IntelliJ kann

- Automatisch JARs erstellen

IntelliJ kann nicht

- Libraries einfach verwalten
- Optionen von javac abstrahieren
- Von überall und jedem aufgerufen werden

Die Lösung



Gradle: Aufbau

- › Gradle besteht aus zwei Teilen: **Projekten** und **Tasks**

Projekt Sammlung von Aufgaben

Task Eine spezifische Aufgabe. Etwa das Kompilieren einer Java Klasse

- › Projekt und Sammlung von Tasks in so genannter *build.gradle* Datei
- › Buildprozess besteht aus Abfolge von Tasks
 - › Hierarchische Abhängigkeit
 - › Nur wenn vorgehender Task erfolgreich abgeschlossen wurde, wird der nächste ausgeführt.

Gradle: Ordnerstruktur

Vorgegebene Ordnerstruktur für cs108

```
src
├── main
│   ├── java
│   │   └── utils
│   │       └── SomeJavaFile.java
│   └── resources
├── test
│   ├── java
│   │   └── utils
│   │       └── SomeJavaFileTest.java
│   └── resources
```


Gradle: Java Plugin

- › Gradle kommt mit einem Plugin für Java Projekte

```
plugins {  
    id 'java'  
}
```

- › Der Kommandozeilenbefehl `gradlew build` erstellt das komplette Projekt: Kompilieren der `.java`-Dateien, JAR erstellen, JavaDoc generieren, Tests ausführen, ...

Gradle: Dependencies

- Größere (Java) Projekte benötigen Bibliotheken
- Welche Library Version hat Alice? Welche Bob?
- Zentrales Repository mit verschiedenen Versionen der Libraries

```
repositories {  
    mavenCentral()  
}
```

```
dependencies {  
    implementation "joda-time:joda-time:2.10.5"  
    testImplementation "junit:junit:4.12"  
}
```

- `implementation` und `testImplementation` ermöglichen kontextspezifische Abhängigkeiten.

Gradle: Namen und Main

- Benennung der erzeugten .jar Datei:

```
version = 0.1.0
// In der Datei "settings.gradle"
rootProject.name = 'example-game'
```

→ Erzeugt: *build/libs/example-game-0.1.0.jar*

- Festlegung der Main-Klasse:

```
mainClassName = 'ch.unibas.cs.p9.fs20.example.HelloWorld'
```

Gradle Wrapper

Alle benutzen Gradle ...

- › ... aber nicht alle die gleiche Version
- › ... aber verschiedene Projekte andere Versionen

Lösung: Gradle Wrapper

- › Kleines Programm mit Konfiguration im Repository
- › Lädt und verwendet die richtige Gradle-Version für das Projekt
- › macOS, Linux: `./gradlew`
- › Windows: `gradlew.bat`

Gradle: Wichtige Befehle

- Projekt bauen
 - `gradlew build`
- Tests ausführen
 - `gradlew test`
- Projekt bauen und ausführen
 - `gradlew run`
- Kompilierte und generierte Dateien löschen
 - `gradlew clean`

Im Programmierprojekt

build-cs108

- › (Ein) Jar erstellen
- › JavaDoc generieren
- › Tests ausführen

Gradle: Beispiel

Demo am Beispielprojekt

Netzwerktest-WiFi

- Die regulären WiFi-Netzwerke der Universität erlauben keine Kommunikation zwischen Endgeräten.
- Separates WiFi-Netzwerk für das Programmierprojekt:

WiFi-Credentials

- SSID: DMI PROGRAMMING
- Passphrase: StudyingIT@unibas=reallyGreat2019

Verfügbar: Spiegelgasse 1, Pharmazentrum, Kollegienhaus, Alte Universität

Hausaufgaben

- › `gradlew build`, Build sollte erfolgreich abschliessen, jar sollte ausführbar sein.

Fragen?

Ressourcen

Ressourcen zu Gradle:

- Gradle Documentation:
<https://gradle.org/docs/>