



LATVIJAS  
UNIVERSITĀTE

ANNO 1919

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



EIROPAS SAVIENĪBA

Eiropas Sociālais  
fonds

IEGULDĪJUMS TAVĀ NĀKOTNĒ

# Java programmēšanas pamati

14. NODARBĪBA

# GRADLE RĪKS



**BUILD**  
**AUTOMATIZĀCIJA**

# GRADLE

- ▶ Elastīgs programmas veidošanas (build) automatizācijas rīks, kas piemērots tam, lai sagatavotu jebkuru programmatūru palaišanai
- ▶ Atvērtā koda
- ▶ Augsta veiktspēja – izpilda tikai tos uzdevumus, kuros ir bijušas izmaiņas
- ▶ Darbojas ar JVM
- ▶ Konvencijas – viegli veidot Java projektus
- ▶ Paplašināmība – vari veidot savus uzdevumu tipus un build modeļus
- ▶ IDE atbalsts - Android Studio, IntelliJ IDEA, Eclipse, and NetBeans
- ▶ Piedāvā paplašinātu ieskatu par builda palaišanu – Build scans

# GRADLE IR UNIVERSĀLS

- ▶ Var izmantot ar dažādām programmēšanas valodām
- ▶ Atbalsta Maven un Ivy saderīgas krātuves un failu sistēmas
- ▶ Ļauj ģenerēt no nulles dažāda tipu projektus
- ▶ Atbalsta spraudņus (plugins)

# GRADLE MODELIS

- ▶ Gradle kopējais builds sastāv no atsevišķi veidotiem uzdevumiem, kuru savstarpēji ir atkarīgi un veido uzdevumu koku
- ▶ Gradle uzdevumu izpildes kārtību nosaka šīs atkarības (dependencies)
- ▶ Uzdevumi sastāv no:
  - ▶ Darbībām – piemēram, failu kopēšana vai pirmkoda kompilēšana
  - ▶ Ievaddatiem – vērtības, faili un direktorijas, kuras «darbību» uzdevumi izmanto
  - ▶ Izvaddatiem – faili un direktorijas, kurus «darbību» uzdevumi modificē vai ģenerē

# GRADLE BUILD FĀZES

- ▶ Inicializācija – iestata vidi, kurā projekts tiks izpildīts un nosaka, kurš projekts tiks veidots
- ▶ Konfigurācija – izveido un nokonfigurē uzdevumu koku un nosaka uzdevumus un to kārtību kādā tie ir jāizpilda (atkarīgs no lietotāja palaistā uzdevuma)
- ▶ Izpilde – izpilda atlasītos uzdevumus



# GRADLE UZSTĀDĪŠANA



# INSTALĀCIJA

- ▶ Lejuplādē gradle jaunāko laidienu - <https://gradle.org/releases/>
- ▶ Izveido mapi – C:\gradle
- ▶ Atarhivē lejuplādēto failu iekš jaunās «gradle» mapes
- ▶ Atver komandrindu
- ▶ Izpilda komandu: «setx path "%path%; c:\Gradle\gradle-6.8.1\bin"»
- ▶ Pārbauda gradle versiju ar komand: «gradle -v»

# JAVA PROJEKTS

## PROJEKTA MAPES IZVEIDE

```
$ mkdir demo  
$ cd demo
```

## (1) PALAIŽ UZDEVUMU INIT

```
$ gradle init
```

```
Select type of project to generate:
```

```
1: basic
```

```
2: application
```

```
3: library
```

```
4: Gradle plugin
```

```
Enter selection (default: basic) [1..4] 2
```

```
Select implementation language:
```

```
1: C++
```

```
2: Groovy
```

```
3: Java
```

```
4: Kotlin
```

```
5: Scala
```

```
6: Swift
```

```
Enter selection (default: Java) [1..6] 3
```

## (2) PALAIŽ UZDEVUMU INIT

Split functionality across multiple subprojects?:

1: no - only one application project

2: yes - application and library projects

Enter selection (default: no - only one application project) [1..2] 1

Select build script DSL:

1: Groovy

2: Kotlin

Enter selection (default: Groovy) [1..2] 1

Select test framework:

1: JUnit 4

2: TestNG

3: Spock

4: JUnit Jupiter

Enter selection (default: JUnit 4) [1..4]

Project name (default: demo): MyDemo

Source package (default: MyDemo): mydemo

## (1) PROJEKTA STRUKTŪRA

```
| — gradle (1)
|   | — wrapper
|       | — gradle-wrapper.jar
|       | — gradle-wrapper.properties
| — gradlew (2)
| — gradlew.bat (2)
| — settings.gradle (3)
| — app
|   | — build.gradle (4)
|   | — src
|       | — main
|           | — java (5)
|               | — demo
|                   | — App.java
| — test
|   | — java (6)
|       | — demo
|           | — AppTest.java
```

# (2) PROJEKTA STRUKTŪRA

1. Mape wrappera failu glabāšanai
2. Gradle wrappera palaišanas skripts
3. Konfigurācijas fails builda nosaukuma un apakšprojektu definēšanai
4. Builda skripts projektam «app»
5. Noklusētā Java pirmkoda mape
6. Noklusētā Java testu pirmkoda mape



## PROJEKTA FAILI - SETTINGS

**settings.gradle**

```
rootProject.name = 'myDemo' (1)  
include('app') (2)
```

1. Projekta nosaukums
2. Buildā iekļautie projekti

## (1) PROJEKTA FAILI - BUILD

**app/build.gradle**

```
plugins {  
    id 'application' (1)  
}  
  
repositories {  
    jcenter() (2)  
}  
  
dependencies {  
    testImplementation 'junit:junit:4.13' (3)  
  
    implementation 'com.google.guava:guava:29.0-jre' (4)  
}  
  
application {  
    mainClass = 'demo.App' (5)  
}
```

## (2) PROJEKTA FAILI - BUILD

1. Pielieto «Application» spraudni, lai izveidotu konsoles programmu
2. Izmanto «JCenter» atkarīgo bibliotēku ielādei
3. Izmanto «JUnit» testa sistēmu
4. Šo atkarību (dependency) izmanto programma
5. Nodēfinē programmas «main» klasi

## PROGRAMMAS PALAIŠANA

```
$ ./gradlew run
```

```
> Task :app:run  
Hello world!
```

```
BUILD SUCCESSFUL
```

```
2 actionable tasks: 2 executed
```

# PROGRAMMAS SAPAKOŠANA

```
$ ./gradlew build
```

```
BUILD SUCCESSFUL in 0s
```

```
7 actionable tasks: 7 executed
```

# ATSAUCES

- ▶ <https://docs.gradle.org/current/userguide>
- ▶ [https://www.tutorialspoint.com/gradle/gradle\\_overview.htm](https://www.tutorialspoint.com/gradle/gradle_overview.htm)
- ▶ <https://www.baeldung.com/gradle>



LATVIJAS  
UNIVERSITĀTE  
ANNO 1919