##### [marcobehlerjetbrains/maven-tutorial (github.com)](https://github.com/marcobehlerjetbrains/maven-tutorial)

[backend-java-lesopdracht-person-test/src/test/java/nl/novi/PersonTest.java at main · hogeschoolnovi/backend-java-lesopdracht-person-test · GitHub](https://github.com/hogeschoolnovi/backend-java-lesopdracht-person-test/blob/main/src/test/java/nl/novi/PersonTest.java)

# Praktisch Maven

Gebasseer op de les van 20-6-23

### Voorbeeld

### Te gebruiken bij:

-

### Eigenschappen:

-

### Buildtool

#### websites

<https://docs.google.com/document/d/1Y-bJSWGyrRESM71MQuCFoYcABlyKwWZulGoYvBu5RRQ/edit>

<https://books.sonatype.com/mvnref-book/reference/index.html>

<https://maven.apache.org/what-is-maven.html>

[AssertJ - fluent assertions java library](https://assertj.github.io/doc/)

#### Maven POM.xml

* Dependecy managment
* Geautomatische taken.
* Vaste structuur
  + /Pom.xml
  + /src/main/java
  + /src/main/resources
  + /src/test/java
  + /target
* Lifecycle/phases gerichte structuur

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

#### voorbeeld

Calculator.java

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int subtract(int a, int b) {

return a - b;

}

public int multiply(int a, int b) {

return a \* b;

}

public int divide(int a, int b) {

if (b == 0) {

throw new IllegalArgumentException("Cannot divide by zero");

}

return a / b;

}

}

CalculatorTest.java

import org.junit.Test;

import static org.junit.Assert.\*;

public class CalculatorTest {

@Test

public void testAddition() {

Calculator calculator = new Calculator();

int result = calculator.add(5, 3);

assertEquals(8, result);

}

@Test

public void testSubtraction() {

Calculator calculator = new Calculator();

int result = calculator.subtract(10, 4);

assertEquals(6, result);

}

@Test

public void testMultiplication() {

Calculator calculator = new Calculator();

int result = calculator.multiply(7, 2);

assertEquals(14, result);

}

@Test

public void testDivision() {

Calculator calculator = new Calculator();

int result = calculator.divide(10, 2);

assertEquals(5, result);

}

@Test(expected = IllegalArgumentException.class)

public void testDivisionByZero() {

Calculator calculator = new Calculator();

calculator.divide(10, 0);

}

}

#### Maven lifecycle

##### Clean lifecycle

Clean

##### Default lifecycle (built-lifecycle)

1. Valideren
2. Compileren
3. Testen
4. Package (maak er een (Jar “java application runtime”) van)
5. Verify
6. Install
7. Deploy

Deze phases zijn opbouwend

* Install = package in lokale maven repository zetten
* Deploy = package in romate maven repository zetten (servers, kubernetes/AWS)

#### Dependencies (pom.xml)

85% van code is externe code

Door:

* Libraries
* Frameworks

#### Demo

Intelij

1. POM.xml documenten in alle klasse aanmaken en opslaan binnen de package.
2. <project>
3. 

Dit moet boen elke POM.xml staan.

1. <modelversion>x.x.x</modelversion>

<groupid>org.xx.matthijs</groupid> //moeten uniek zijn

<artifactid>afdeling-project-xx</artifactid> //moeten uniek zijn

<version>0.0.1</version>

1. Op Load maven project drukken.

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

1. <depenencies>

<dependency>

<groupid>junit.jupiter</groupid>

<artifactid>junit-jupiter-engine</artifactid>

<version>5.9.2</version>

<scope>”test” of </scope> // alleen voor test beschikbaar

</dependency>

</dependencies>

1. Afbeelding met tekst, software, Webpagina, Computerpictogram

   Automatisch gegenereerde beschrijving
2. New -> directory
3. Afbeelding met tekst, schermopname, software, Webpagina

   Automatisch gegenereerde beschrijving
4. </project>

#### Unit tests

* Aftesten van methode/ classes
* Gebruik van junit (library toe te voegen aan dependencies)
  + JUnit is een populaire open-source testframework voor Java-programma's. Het biedt een eenvoudige en gestructureerde manier om unit tests te schrijven voor Java-code. Met JUnit kun je testcases maken die de functionaliteit van individuele eenheden (zoals methoden, klassen of modules) van je Java-programma testen.
  + JUnit maakt gebruik van annotaties om testmethoden en asserties te definiëren. Een testmethode wordt gemarkeerd met de annotatie @Test, en asserties worden uitgevoerd met behulp van verschillende assertie-methoden, zoals assertEquals, assertTrue, assertFalse, enzovoort.
  + Hier zijn enkele kenmerken van JUnit:
  + Eenvoudige syntaxis: JUnit heeft een eenvoudige en leesbare syntaxis die het gemakkelijk maakt om testcases te schrijven.
  + Testfixtures: JUnit ondersteunt het opzetten van testfixtures, zoals het initialiseren van objecten, voorbereiden van testgegevens en opruimen na het uitvoeren van tests.
  + Assertions: JUnit biedt een reeks assertiemethoden waarmee je de verwachte resultaten kunt vergelijken met de werkelijke resultaten van je code.
  + Testuitvoering en rapportage: JUnit voert testcases uit en rapporteert over de resultaten, zoals welke testcases zijn geslaagd of mislukt.
* Negatief testen// zolang alle methodes maar getest worden
* Positief test // zolang alle methodes maar getest worden

#### Junit test

Afbeelding met tekst, elektronica, schermopname, scherm

Automatisch gegenereerde beschrijving

* // given of arrange
* // when of act
* // the of assert

##### Asserts

* Outpute controle op voorwaarden
* Nooit zonder:
  + assertsEquals
  + assertsTrue
  + assertsFalse
  + assertsNotnull
  + assert
  + Afbeelding met tekst, schermopname, Lettertype

    Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, software, Besturingssysteem

Automatisch gegenereerde beschrijving

#### Mocking(mockito)

##### Mock interfaces

(recorded behaviour)

* Wanner je het volledige gedrag niet wilt uitprogammeren
* Vastleggen van gedrag van methode
* In Unittest vaak gebruikt.

Afbeelding met tekst, schermopname, software, Besturingssysteem

Automatisch gegenereerde beschrijving

##### Voorbeeld

*<?*xml version="1.0" encoding="UTF-8"*?>*<**project** xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <**modelVersion**>4.0.0</**modelVersion**>  
  
 <**groupId**>nl.novi</**groupId**>  
 <**artifactId**>Family</**artifactId**>  
 <**version**>1.0</**version**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.junit.jupiter</**groupId**>  
 <**artifactId**>junit-jupiter-api</**artifactId**>  
 <**version**>5.8.2</**version**>  
 </**dependency**>  
 </**dependencies**>  
  
 <**properties**>  
 <**maven.compiler.source**>17</**maven.compiler.source**>  
 <**maven.compiler.target**>17</**maven.compiler.target**>  
 </**properties**>  
  
</**project**>

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving