

Software engineering fundamentals



Maarten Troost – Jens Baetens



Doelstellingen





Doelstellingen

- Leer werken als een software engineer
 - Leer de Java-programmeertaal kennen en gebruiken in een Software Engineering context
 - Weet hoe u te werk gaat om vertrekkende van een probleemstelling op de juiste manier te komen tot efficiënte en productieve code
 - Leer object-georiënteerd werken
 - Leer hoe je software automatisch kunt testen
 - Leer hoe je code schrijft die lang kan gebruikt/herbruikt worden
 - Leer samenwerken aan code
 - Eerste voorbereiding op certificatie tot "Java Foundations Junior Associate"



Te ontwikkelen competenties – niveau 1

1.3 Complexe informatie en IT-behoeften kritisch analyseren.
1.4 IT-oplossingen bedenken en modelleren.
2.1 De aangewezen techniek hanteren om de gewenste gegevens efficiënt te verzamelen.
3.1 Inzicht verwerven in de behoeften van verschillende stakeholders.
7.1 Een oplossing voor een (complex) IT probleem kunnen ontwerpen
7.2 Een oplossing kunnen implementeren, optimaal gebruik makend van standaard- of modeloplossingen
7.3 Een integrale oplossing structureel kunnen testen en de resultaten kunnen gebruiken om het ontwerp en/of de oplossing continu te verbeteren
7.4 Een oplossing kunnen documenteren zodat anderen de oplossing kunnen begrijpen, aanpassen of instaan voor het onderhoud
8.2 Deadlines en afspraken kunnen respecteren
8.3 Kritisch kunnen reflecteren over het functioneren van het team, de leden en zichzelf

8.4 Verantwoordelijkheden of leiderschap opnemen in een team en waken over het welbevinden van de teamleden

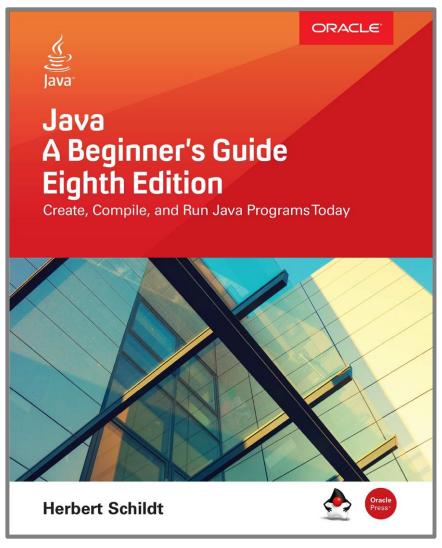


Interessant / Aanbevolen boeken



http://www.vanduurenmedia.nl/EAN/9789059403048/Handboek_Objectgeorienteerd_programmeren_2e_ed_





https://www.amazon.co.uk/Java-Beginners-Eighth-Herbert-Schildt/dp/1260440214/ 6

Voorlopige planning

- Leerverslag
 - Per 2 in te dienen
 - Geen vaste groep verplicht
- Demo + code
 - Vaste groep
 - Project + Presentatie



Planning SEF 2021-2022

	Week	Online/On Campus	Topics	Evaluatie
t	1	Op campus	Intro vakinhoud \ manier van evalueren Intro Java Installeren Java Java via commandline, commandline arguments Installeren en gebruik IntelliJ	
	2	Op campus	UML + project lifecycle + TDD + documentatie	Leerverslag
	3	Op campus	JavaFX + basics + TDD	Leerverslag
	4	Op campus	Groepswerk + encapsulation + overloading	Leerverslag
	5	Op campus	composition, overerving, polymorphisme	Leerverslag
	6	Op campus	abstract, interfaces, exceptions, packages	Leerverslag
	7	Op campus	Presentatie projecten	Groepswerk: demo+code

ECTS – fiche evaluatie

- Permanente evaluatie
 - 20% Proces
 - 40% Werkproduct
 - 20% Documentatie
 - 20 % Presentatie
- Optioneel openboek mondeling examen met open vragen
 - Opt-in = je kan zelf kiezen of je deelneemt
 - Indien je deelneemt dan telt dit examen mee voor 40% (permanente evaluatie in totaal dan 60%)



Leerverslag

- Beschrijft hoe je de opdracht hebt aangepakt, de genomen beslissingen en de ervaren problemen
- Schrijf het voortdurend: te maken tijdens de opdracht, niet erna!
- Het verslag en de code indienen via github classroom.
- Inhoud
 - Introductie met de namen van de studenten en de titel van de opdracht
 - Werkverslag
 - Reflectie
- Verzorg de vormgeving
- Zie Toledo voor meer informatie



Pair programming





Project

- In vaste groepen
- Opgave
 - Kies een aantrekkelijke toepassing voor een bedrijf
 - Maak een oplossing voor het gekozen probleem op te lossen
 - Pas zoveel mogelijk van de geziene technieken uit de lessen van SEF toe
- Link naar de opdracht
 - TI 1-2: https://classroom.github.com/a/5ogmSID
 - TI 3-4: https://classroom.github.com/a/eXJdsdrQ
 - TI 5-6: https://classroom.github.com/a/wNPqLmYl
 - TI 7-8: https://classroom.github.com/a/qhqwGMiU



Deadlines

- 01/05 23u59: Maak groepen, lees de artikels op Toledo uit je klasgroep en kies je top 3 topics of stel er zelf 1 voor
- 08/05 23u59: Stel een aantrekkelijke toepassing voor en commit dit naar github
- 29/05 23u59: Eerste oplossing indienen
- 06/06 23u59: Finale versie + presentatie + peer reviews indienen



Presentatie project

- Presentatie van 10 minuten
 - Demonstreer het resultaat van uw teamwork
 - Toon de code en leg ze uit
 - Zorg dat elk teamlid aan het woord komt
 - ledereen moet alle code kennen
- 5 minuten voor vragen



Peer evaluatie

- Individueel en vertrouwelijk
- In te dienen via Toledo
- Inhoud
 - Evaluatie door [uw naam]
 - Volgens mij heeft/is sinds vorige les [naam geëvalueerde]
 - [score] de afgesproken taken uitgevoerd
 - [score] bijgedragen tot ideeën en planning voor het werk
 - [score] bereikbaar/aanspreekbaar geweest voor overleg
 - [score] bijgedragen tot het succes van het project
 - Op te nemen in het verslag als feedback voor deze persoon [uw feedback in maximum een aantal regels]

1 = niet/nauwelijks

2 = af en toe / soms

3 = meestal / altijd



Vakinhoud





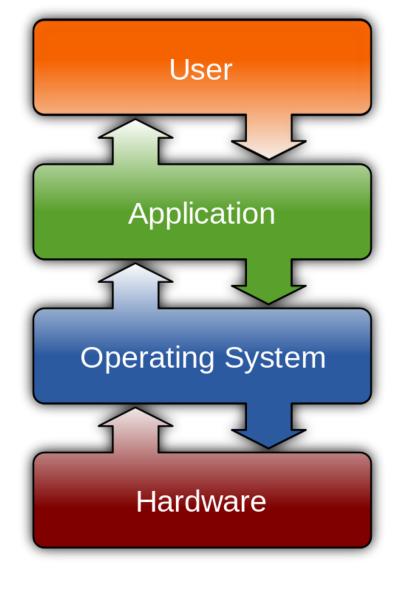
Overzicht van de SEF werkwijze

- Doorgronden van de probleemstelling
- Opstellen van lijst met te leveren publieke diensten
- Herhaal:
 - JUnit test maken (op basis van de lijst)
 - Schrijf code (eerst het skelet, daarna de details)
 - Refactor
 - Javadoc-documentatie schrijven
- En daarna ... de GUI dus



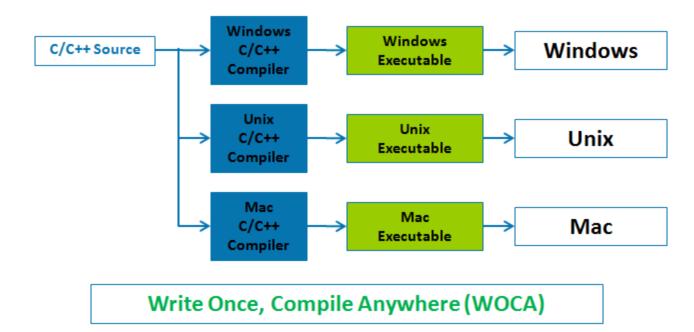
JAVA

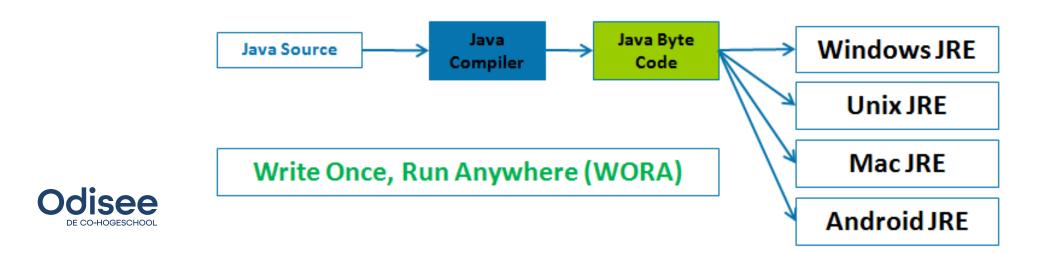
- Virtueel platform
- Hogere programmeertaal (gelijkaardig aan C#)
- Gecompileerd naar virtueel platform
 - Dit platform wordt aangeboden door OS
- Platformonafhankelijk





Java is platformonafhankelijk door





JRE vs JVM vs JDK

JRE

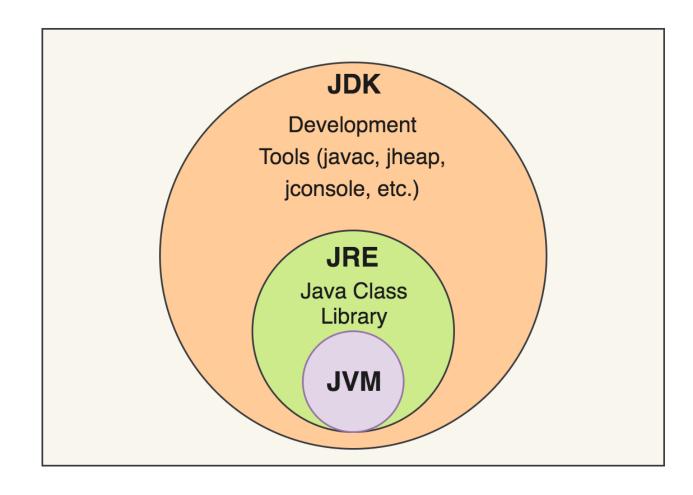
- JVM
- standaard software componenten
 - OS-afhankelijk (dialogs, ...)

JDK

Plus compiler, javadoc, jar-tool, ...

Types

- SE → standard edition
- ME -> mobile edition (oude gsm's)
- EE -> enterprise edition (servers)





Geschiedenis?

- Gemaakt door Sun
- Sun is in 2009 overgenomen door Oracle
- Nu elk jaar minstens 1 update
- Java gebruikt in grote bedrijven
 - Banken, Verzekeringen, ...
 - Backwards-compatibility is belangrijk!
- C# eerder in kleinere bedrijven



Installatie JDK

- Ga naar https://www.oracle.com/java/technologies/downloads/
- Kies de meest recente SE development kit voor jouw platform (windows, linux, mac)
- Download en installeer het

■ Kan zijn dat het reeds geinstalleerd is door Visual Studio Code





Java(TM) SE Development Kit 18 (64-bit) - Setup





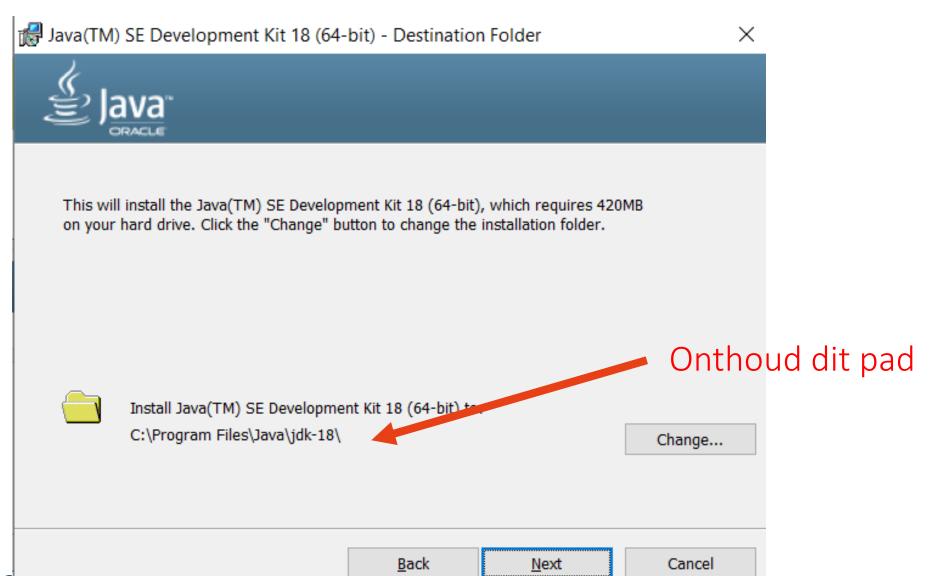
Welcome to the Installation Wizard for Java SE Development Kit 18

This wizard will guide you through the installation process for the Java SE Development Kit 18.



Next >

Cancel







Command Prompt

```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.
C:\Users\jens.baetens3>java -version
iava version "18" 2022-03-22
Java(TM) SE Runtime Environment (build 18+36-2087)
Java HotSpot(TM) 64-Bit Server VM (build 18+36-2087, mixed mode, sharing)
C:\Users\jens.baetens3>javac -version
iavac 18
C:\Users\jens.baetens3>
```



Installatie Linux

\$ sudo apt update

\$ sudo apt install default-jdk



Installatie Mac

https://www.java.com/nl/download/help/mac_install.html



Eerste JAVA programma

- Maak een textfile aan met volgende code.
 - Naam van de file is JavaTest.java

```
public class JavaTest {

    public static void Main(){
        System.out.println("Hello World");
    }
}
```



Compileer en voer het uit

C:\Users\jens.baetens3\OneDrive - ODISEE\SEF\Leerstof\Leerstof\Week 1>javac JavaTest.java

C:\Users\jens.baetens3\OneDrive - ODISEE\SEF\Leerstof\Leerstof\Week 1>java JavaTest Hello World



Oefening

■ Pas de tekst aan uit het eerste programma en voer het programma opnieuw uit



Belangrijke terminologie

- U leert user software te maken.
- Een Java-compiler produceert bytecode. Dit is platformonafhankelijke code die door een interpreter wordt uitgevoerd.
- Java-programma's worden uitgevoerd op een Java Virtual Machine (JVM). Dat is een operationele computer waaraan een Java Runtime Environment (JRE) is toegevoegd.
- Om zelf Java-programma's te maken, hebt u een Java Software Development Kit (SDK) nodig.

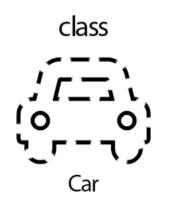
Nieuwe termen en begrippen		
User software	Bytecode	
Operating System	Java Runtime Environment (JRE)	
Virtuele machine	Java Virtual Machine (JVM)	
Platform	Software Development Kit (SDK)	
Platform(on)afhankelijkheid		



Refresher OOP



Object-georiënteerde concepten









Audi

Nissan

Volvo

- Programmeertaal-onafhankelijk
 - Klassen, objecten, methoden, properties, ...
 - Inkapseling
 - Overerving, Interfaces, Packages
 - Software architectuur

Work phases in building planning (HOAI ⁴)		Phases in software engineering
1	Basic evaluation	Analyzes
2	Preliminary planning	Definition of requirements
3	Blueprint planning	Software architecture design
4	Approval planning	
5	Implementation planning	Design of components & classes
6	Preparation in award of contract	
7	Participation in award of contract	
8	Supervision/controlling of the construction	Implementation
		Integration and tests
9	Handling of warrantee claims and documentation	Documentation (introduction/training)
		Maintenance



Programma's en realiteit

- Doel van ontwikkelen software is
 - Het maken van een bruikbaar model van de realiteit dat net genoeg is om het probleem op te lossen
- Voorbeeld loonberekening

Relevant	Niet-relevant
Naam	Geloofsovertuiging
Voornaam	Haarkleur
Adres	Schoenmaat
Functie	Geaardheid
Aantal dienstjaren	Kleur van ogen
Gehuwd	Lievelingseten



Programma's en realiteit

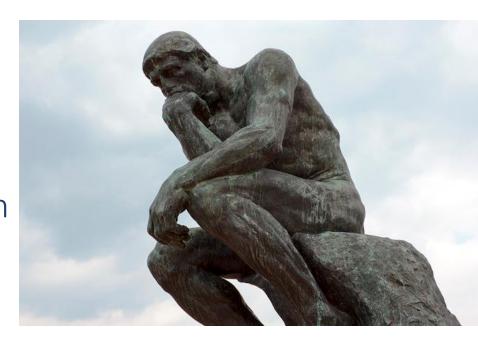
- Voorbeeld 2: Administratie huisarts
 - Welke kenmerken zijn relevant?

Kenmerken	
Naam	Aantal dienstjaren
Voornaam	Gehuwd
Adres	Lichaamsgewicht
Geaardheid	Schoenmaat
Kleur van ogen	Gevaccineerd tegen
Beoefende sporten	Lievelingseten
Ziektehistoriek	



Programma's en realiteit

- In beide voorbeelden gaat het over een Persoon
 - Relevante informatie verschilt van toepassing
 - Soms relevant soms irrelevant
- Vertrek vanuit de werkelijkheid
 - Is een verzameling van objecten die met elkaar samenwerken
 - Objecten verzamelen een reeks data over "iets"
 - Bvb: Een object is een driehoek dat informatie bevat over de drie punten waaruit het bestaat





Oefening: Observeer een kruispunt en merk het volgende op

Auto	kleur = groen, merk = Volvo	vertraagt	
Auto	kleur = blauw, merk = Audi	rijdt aan 30 km/u	
Auto	kleur = rood, merk = BMW	staat stil	
Voetganger	geslacht = vrouw, haarkleur = blond, kleding = jas	steekt straat over	
Voetganger	type = kind, leeftijd = 10 jaar	wacht voor rood licht	
Fietser	geslacht = man, voertuig = fiets (kleur = rood)	wacht voor rood licht	
Fietser	geslacht = man, voertuig = fiets (kleur = blauw)	wacht voor rood licht	
Fietser	type = meisje, voertuig = fiets (kleur = geel)	rijdt door het groene licht	
Verkeerslicht	hoogte = 4m, aantal lichten = 3	rood licht brandt	
Verkeerslicht	hoogte = 4m, aantal lichten = 3	groen licht brandt	

■ Welke klassen zijn er? Hoeveel instanties zijn er van elke klasse? Welke eigenschappen hebben de verschillende klassen? Welke operaties zijn er?

■ Welke interacties tussen objecten zijn er?



Oplossing

■ Klassen:

- Auto met 3 instanties
 - Eigenschappen: kleur, merk, ...
 - Operaties: rijden aan een snelheid, vertragen, stilstaan
- Voetganger met 2 instanties
- Fietser met 3 instanties
 - Eigenschappen: type, geslacht, voertuig
 - Operaties: wachten, rijden
- Verkeerslicht met 2 instanties

Interacties

- Voetganger wacht bij een verkeerslicht
- auto vertraagt omdat het een verkeerslicht nadert



Belangrijke terminologie

- Een programma is de realisatie van een model van een stukje werkelijkheid.
- OOP is een methode van programmeren die het toelaat zo dicht mogelijk bij concepten van de te modelleren realiteit te blijven.
- Een klasse is de beschrijving van een reeks gelijksoortige objecten.
- Een dergelijk object heet dan een instantie van de klasse.

Nieuwe termen en begrippen

Klasse

Object

Instantie



Wat gaan we zien in de loop van dit vak over OOP?

Principle	What it is	Why we use it	How to do it
Data encapsulation	Technique that hides information from the users of that class	To reduce the level of software development complexity	Use access modifiers such as public, private, and protected
Inheritance	Technique to allow a derived or child class to use parts of a base or parent class	To promote the re-use of the software	Use the extends keyword
Polymorphism	Technique which supports different behavior of methods that is dependent on the object the method is executing against	To make an application more maintainable	Inherent to the Java language

IDE Integrated Developer Environment



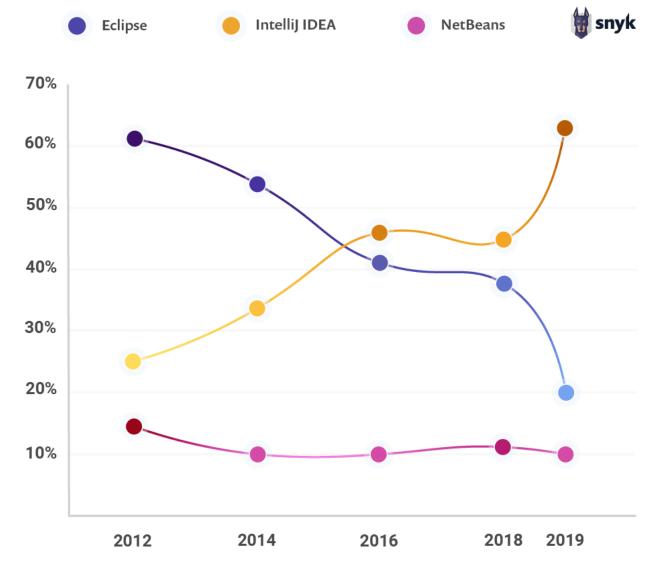
Hoe Java programmeren?

- Java programma's bestaan uit .java bestanden
- Deze kunnen geopend en bewerkt worden door een standaard tekst editor
 - Of Notepad++
 - Of Atom
 - Of Visual Studio Code
- Functionaliteiten beperkt
 - Maak gebruik van een IDE voor programmeren (zoals Visual Studio)



Welke IDE gebruiken voor Java?

- Er zijn drie veelgebruikte
 - Eclipse
 - Netbeans
 - IntelliJ



https://snyk.io/blog/intellij-idea-dominates-the-ide-market-with-62-adoptionamong-jvm-developers/



- Maak een student account aan
 - Gebruik je odisee mail
- Vul de gegevens in en klik op apply

Individual licenses for students and teachers

Get free access to all JetBrains IDEs for personal use at school or at home.

Who can get free individual licenses for education

Students and faculty from accredited educational institutions (high schools, colleges, and universities) are welcome to apply.

Students need to be enrolled in an accredited educational program that takes one or more years of full-time study to complete.

Not sure about the license terms? Check out the FAQ or read the full terms here.

Apply now

For students and teachers



Installeren IDE

- Klik op de link in de mail om je account te activeren
- Volg de stappen om de EULA te accepteren
 - Je moet naar beneden scrollen voor de knop actief wordt

Hi,

You're receiving this email because your email address was used to register or update a JetBrains Educational Pack.

Please click this link to proceed:

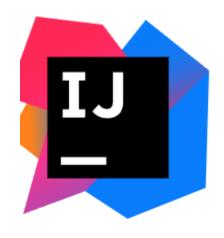
https://www.jetbrains.com/shop/eform/students/request?code=agoljn6jnqsvay1i7b2hxzx5s

After accepting the License Agreement, you will be asked to sign up for a JetBrains Account. You will need to use this account whenever you want to access JetBrains tools.



Installeren IDE

- https://www.jetbrains.com/idea/download/
 - Installeer ultimate edition (gratis met je student account)



ersion: 2021.3.3 uild: 213.7172.25 7 March 2022

odise€ elease notes ≯

Download Intelly IDEA

Windows macOS Linux

Ultimate

For web and enterprise development



Free 30-day trial

Community

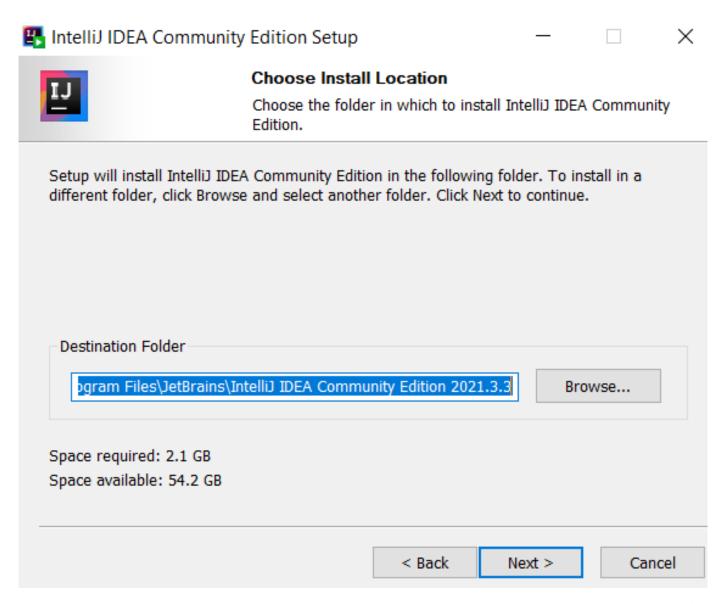
For JVM and Android development



Free, built on open source

Volg de installatiestappen

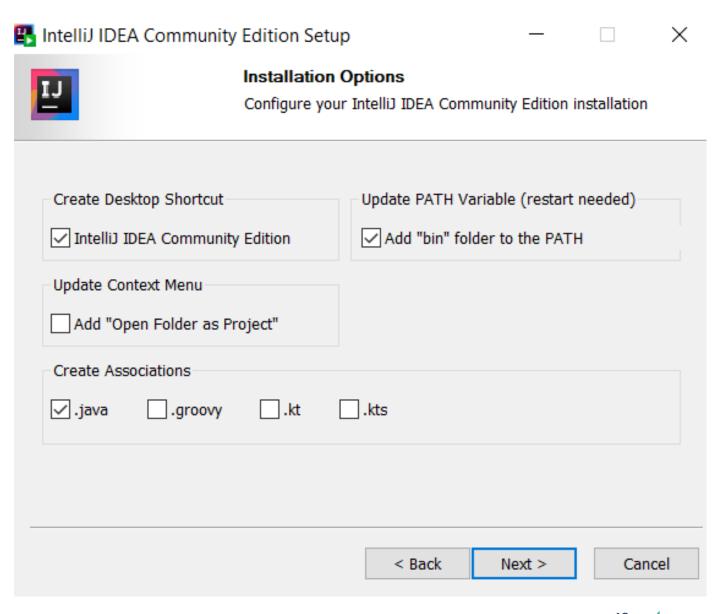
■ Kies installatielocatie



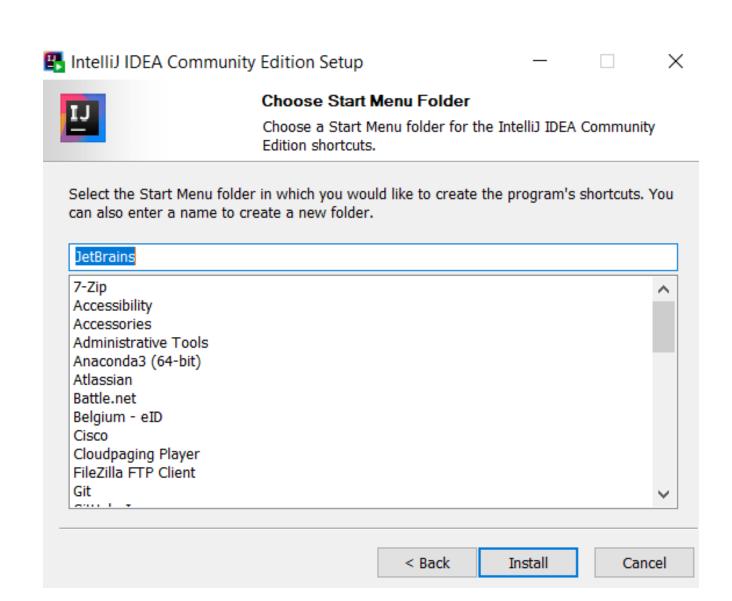


Installatie

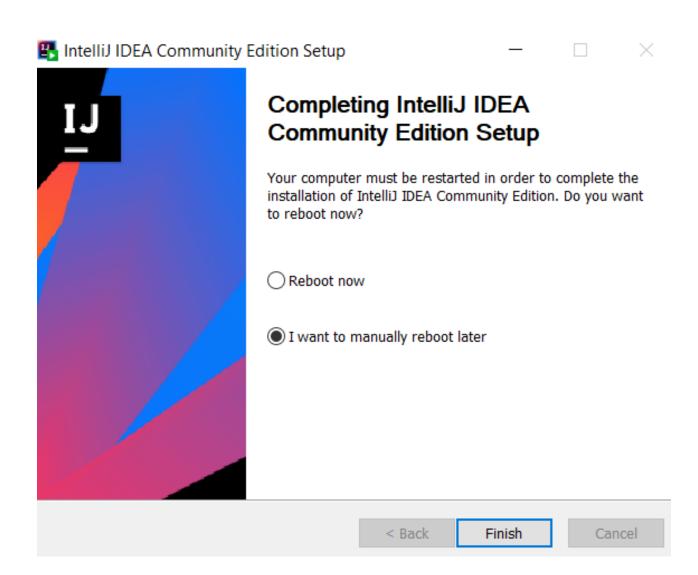
■ Selecteer de volgende opties













Installeren IDE



 \times

JETBRAINS USER AGREEMENT

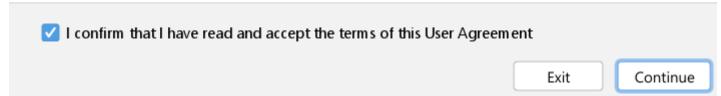
Version 1.4, effective as of September 22, 2021

IMPORTANT! READ CAREFULLY:

THIS IS A LEGAL AGREEMENT. BY CLICKING ON THE "I AGREE" (OR SIMILAR) BUTTON THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE JETBRAINS SOFTWARE, SUPPORT, OR PRODUCTS, YOU BECOME A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL CAPACITY TO ENTER INTO SUCH AGREEMENT, AND YOU CONSENT TO BE BOUND BY ALL THE TERMS AND CONDITIONS SET FORTH BELOW.

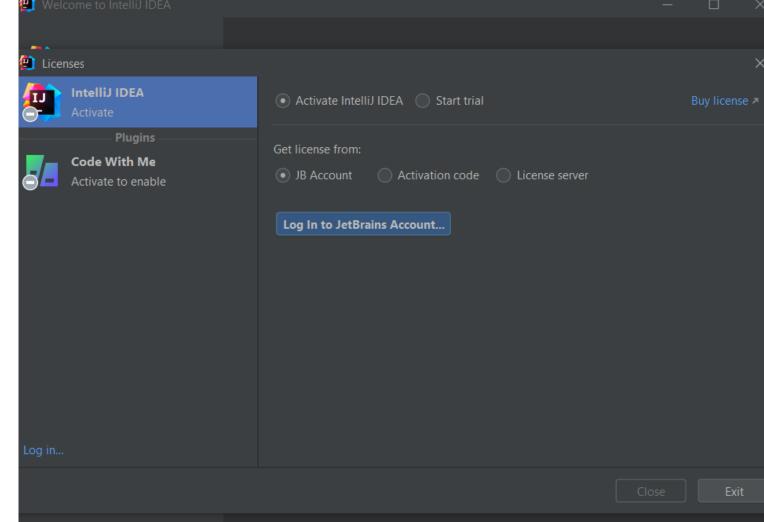
1. PARTIES

1.1. "JetBrains" or "we" means JetBrains s.r.o., having its principal place of business at Na Hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague, Section C, File 86211, ID No.: 265 02 275.



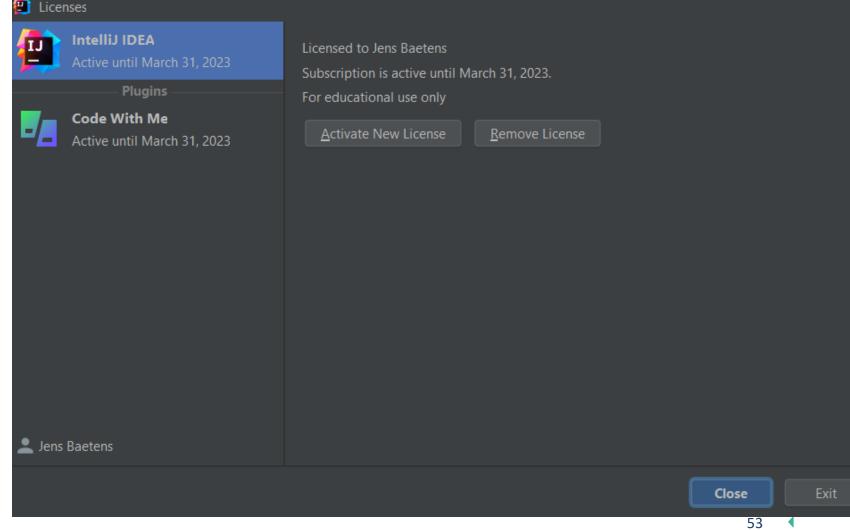


- Log in met je account
- Klik op activate





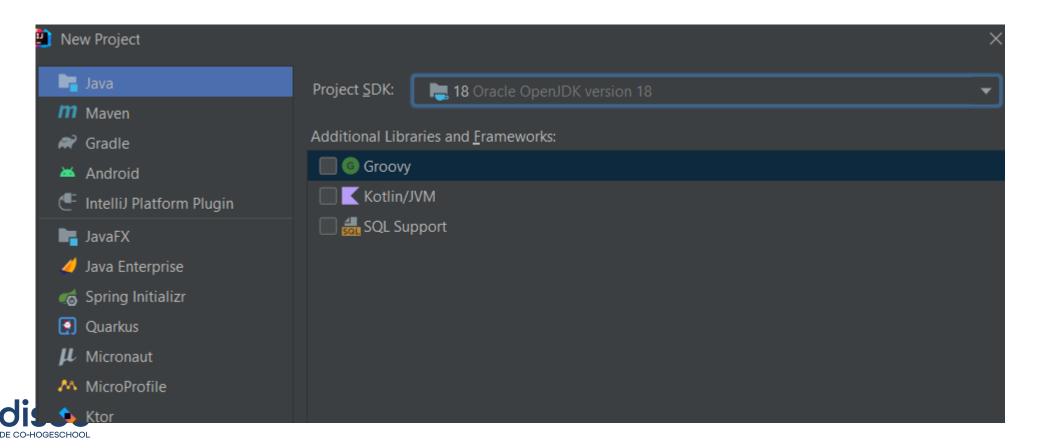
■ Klik op close



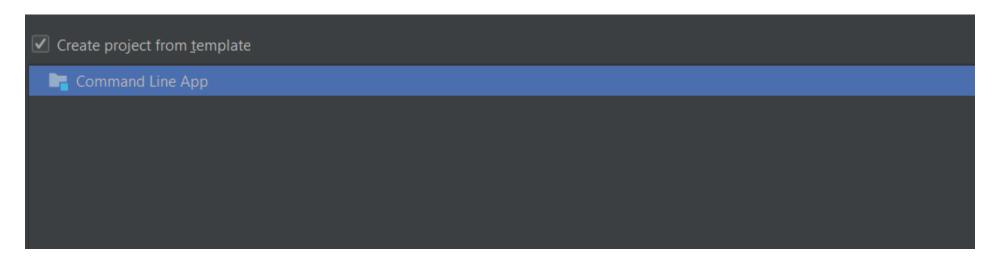


Maak een nieuw project

■ Selecteer Java en klik op next

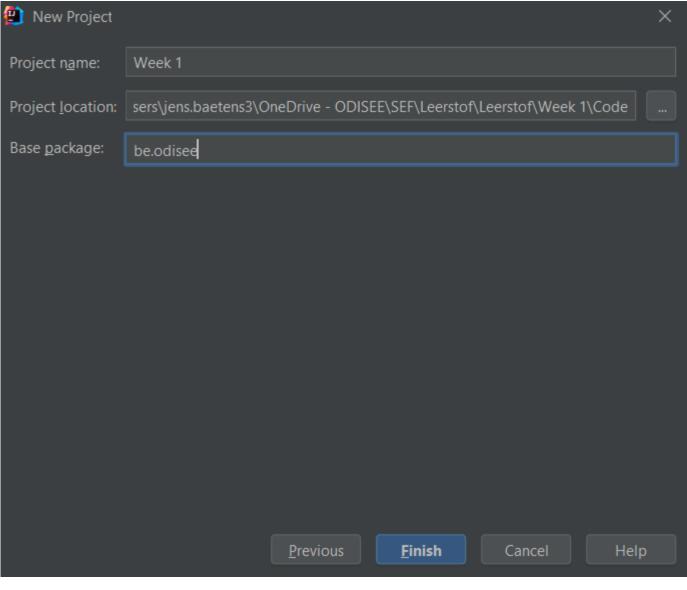


- Selecteer create project from template
 - Kies command line app

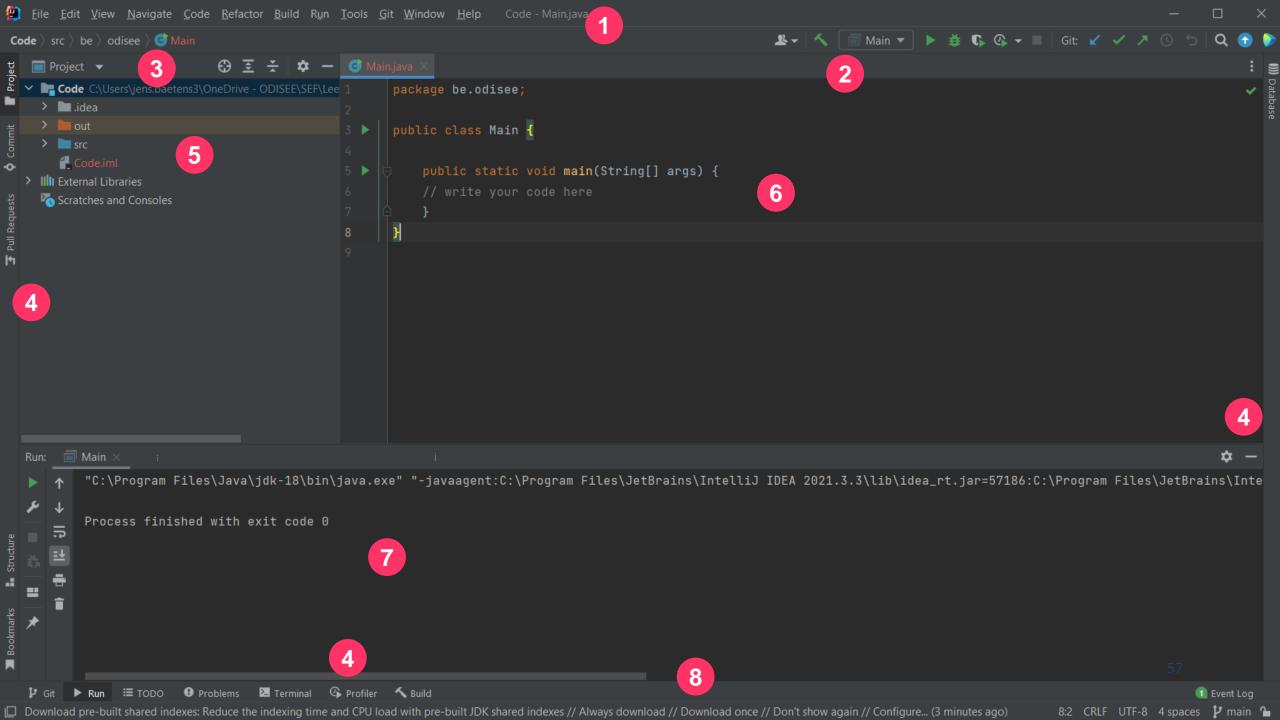




- Kies een naam
- Kies een locatie voor het project
- Kies een start package
 - Bvb be.odisee
- Klik op finish



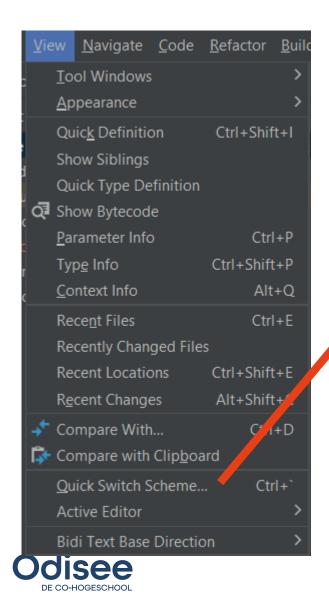


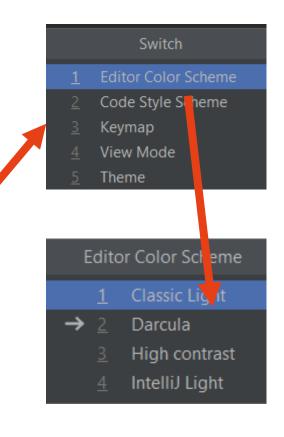


- Hoofdmenu -> projecten beheren, refactoring, intellij aanpassen
- Hoofdtoolbar -> Knoppen om taken uit te voeren binnen het project
- Navigatiebar -> Welk betand actief is + pad in het project 3.
- Toolbladbladen -> Links, onderaan, rechts met tools binnen het project. Kunnen ook geopend worden via view
- Toolbladvenster -> Vensters die opengaan bij klikken op een tool
- Editor -> Venster met de code van de actieve file
- Run-scherm -> Belangrijk scherm voor console applicatie voor input/ouput
- Statusbar -> Huidige toestand van de IDE



Keuze thema van Intellij





Hello world applicatie

- Voeg een lijn toe dat output print naar console
 - Tip: type "sout" en kijk wat de intellisense aanbeveelt
 - Andere handige afkorting psvm
 - Dit maakt de main functie aan

```
package be.odisee;

public class Main {

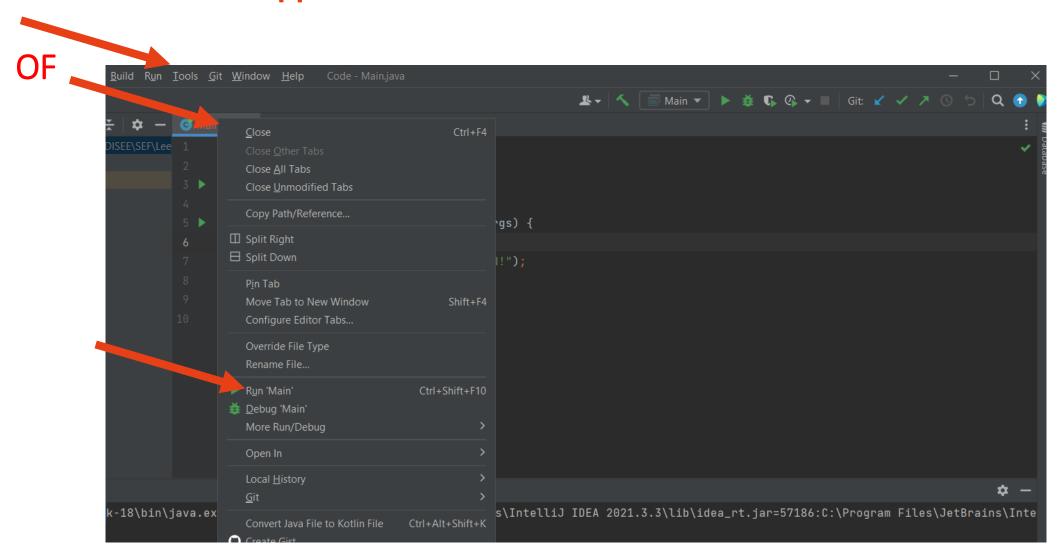
public static void main(String[] args) {
    // write your code here
}

}

}
```



Uitvoeren van de applicatie





Uitvoeren van de applicatie

- In de achtergrond doet IntelliJ de commando's
 - javac
 - java
- Met als resultaat in het run-view

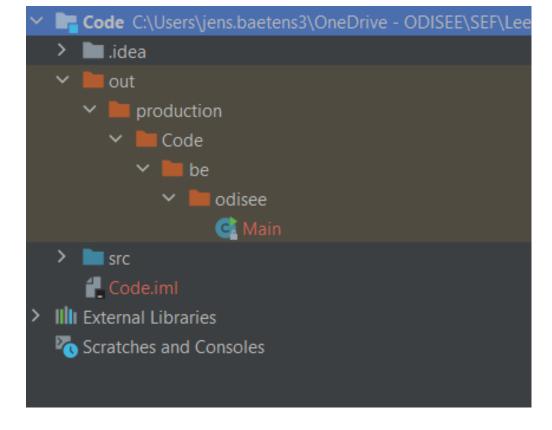
```
Main ×
Run:
        "C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.3\lib\idea_rt.ja
        Hello world!
        Process finished with exit code 0
```



Plaats bytecode

■ Al de .class files aangemaakt door javac worden in een aparte folder geplaatst

Standaard in de "out" folder





Wat gebeurt er bij het uitvoeren van de applicatie via IntelliJ

- Alle files worden bewaard
- Alle .java files die veranderd zijn worden gecompileerd
- De .class file dat de main functie bevat wordt uitgevoerd

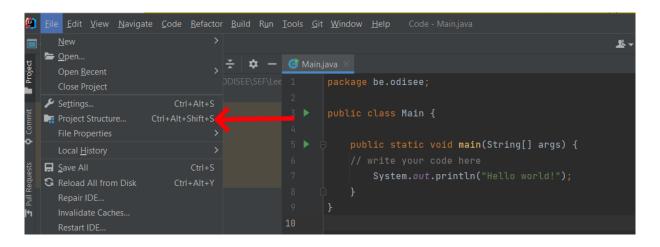


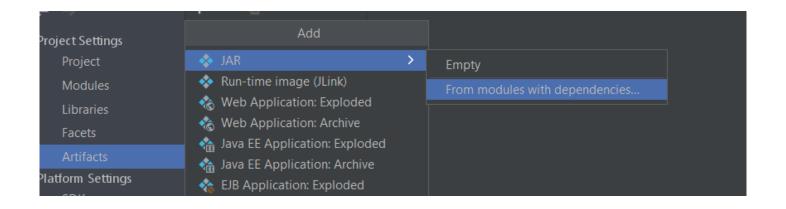
Hoe verspreiden van de applicatie

- Aan de hand van een Jar file
 - Java Archive
 - Een package/zip met alle .class files
- In IntelliJ zijn de volgende stappen nodig
 - Maak een artifact aan
 - Build de artifact
 - https://www.jetbrains.com/help/idea/compiling-applications.html#package_into_jar



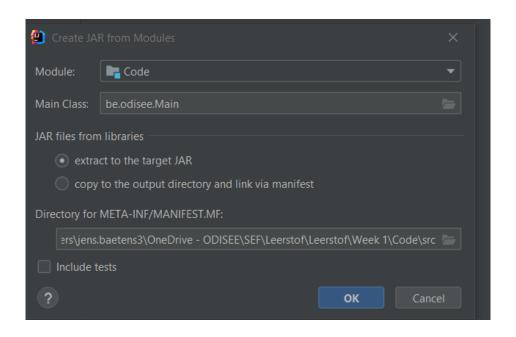
Maken van een artifact





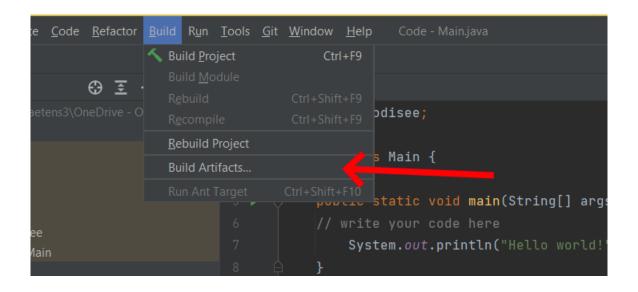


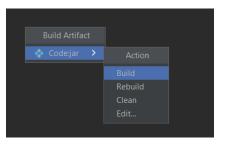
Maken van een artifact





Builden van een artifact







Uitvoeren van de jar

- Deze jar file kan verplaatst worden naar elke computer beschikkende over een JRE
- Kan uitgevoerd worden door middel van het java –jar commando

C:\Users\jens.baetens3\OneDrive - ODISEE\SEF\Leerstof\Leerstof\Week 1\Code\out\artifacts\Code_jar>java -jar Code.jar Hello world!



Aantal handige tips

- Dubbelklik op de bestandsnaam om het editor-scherm zo groot mogelijk te maken
- Rechtsklik op de bestandsnaam om meerdere bestanden naast elkaar te zetten
 - Verticaal: split-down
 - Horizontaal: split-right



Uitgebreider voorbeeld

■ Maak twee klassen aan:

- Square
 - Bevat twee functies
 - Square -> berekend en returned het kwadraat
 - SquareRoot -> berekend en returned de vierkantswortel
- Calculator
 - Bevat de main functie
 - Vraagt een getal als input via commandline
 - Berekent het kwadraat en vierkantswortel met behulp van de Square klasse
 - Print alle berekende gegevens uit naar commandline



Java – Naming conventions

Name	Convention	
class name	should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.	
interface name	should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.	
method name	should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.	
variable name	should start with lowercase letter e.g. firstName, orderNumber etc.	
package name	should be in lowercase letter e.g. java, lang, sql, util etc.	
constants name	should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.	



Opdrachten





Leerobjecten bekijken

- Beschikbaar op Toledo
- Herhaling van de les



Opdrachten

- De opdracht staat op github classroom en kan bereikt worden via de link
 - TI 1-2: https://classroom.github.com/a/jJZeQtP3
 - TI 3-4: https://classroom.github.com/a/YDMCa wq
 - TI 5-6: https://classroom.github.com/a/WPnn5g9n
 - TI 7-8: https://classroom.github.com/a/FwhwWf 5
- Dien de code in door het te pushen naar github
 - eventueel via IntelliJ git integratie
 - of externe git applicatie zoals github desktop
- Vergeet ook het leerverslag niet toe te voegen als pdf in de git repository

