



# Programming Basics

## Hoofdstuk 3

### Mijn eerste Java-toepassing

#### **DE HOGESCHOOL MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.pxl.be/facebook)



# Gebruikte symbolen



**Eclipse Tips:** handig om vlot te programmeren.

Opgelet: je moet de syntax wel vanbuiten kennen voor het schriftelijk examendeel!!!



**Programmeertips**



**Ter informatie;** wordt niet geëvalueerd op het examen



# Inhoud

1. Inleiding
2. De broncode schrijven
3. De broncode compileren
4. De bytecode uitvoeren
5. De opbouw van het programma



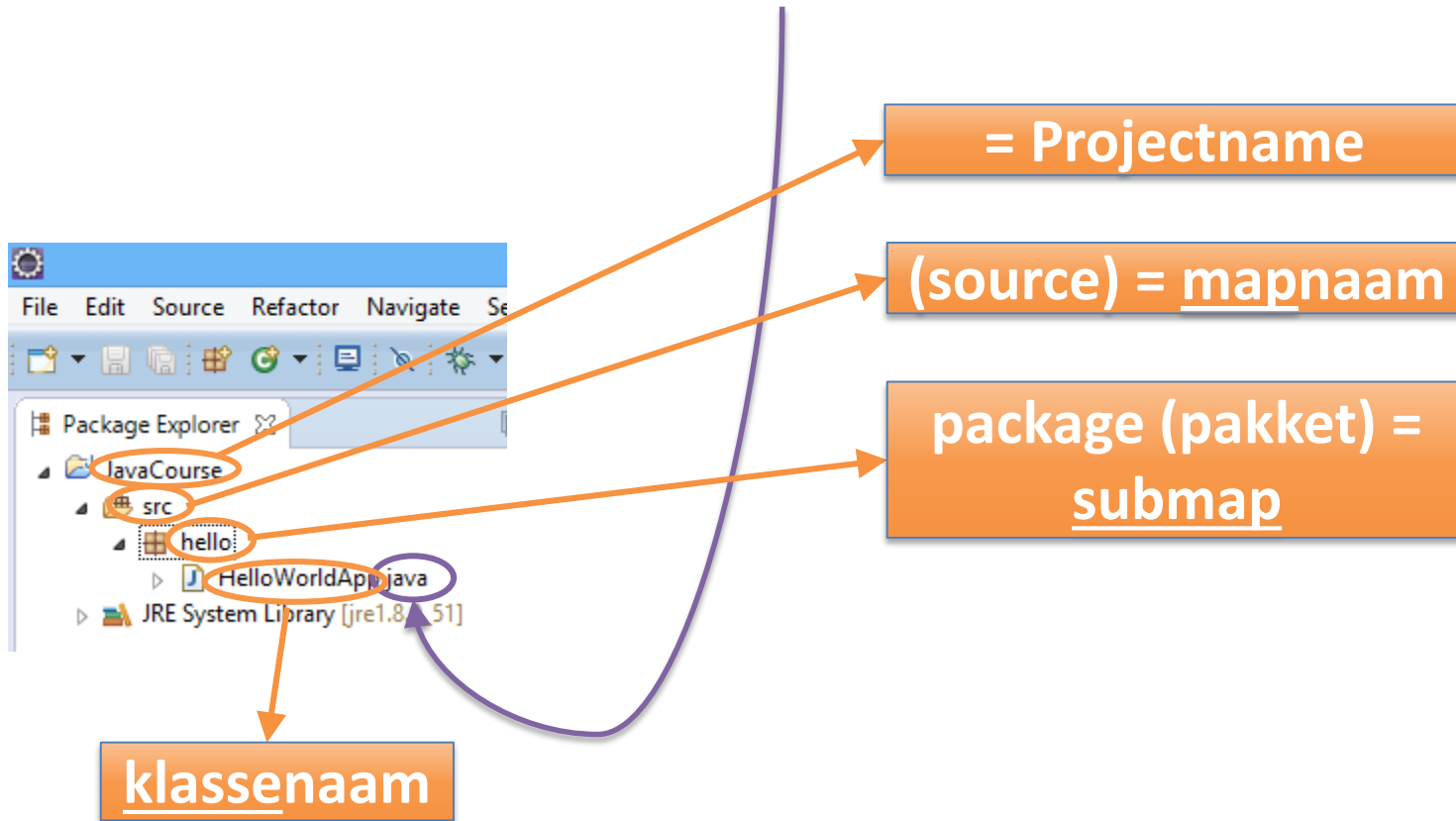
# 1. Inleiding

Een Java-programma schrijven gebeurt in 3 fasen:

- Het maken van de broncode
- Broncode compileren naar bytecode
- Bytecode uitvoeren met de JVM



## 2. De broncode schrijven



→ C:\workspacename\JavaCourse\src\hello\HelloWorldApp.java

# Opdracht 1: Mijn eerst Java-programma!

- Maak een package *hello*:

- selecteer de map *src*

- Kies vanuit het context-menu *New > Package*

- Geef naam *hello* →  *hello*

Package name: kleine letter

= rechtermuisklik (RM)

Class name: hoofdletter

- Maak een klasse *HelloWorldApp*

- Selecteer de map *src/hello*

- *RM > New > Class*

- Geef naam *HelloWorldApp* →  *HelloWorldApp.java*

-  *hello* wijzigt in  *hello*

- Tik onderstaande broncode in (case sensitive!)

```
/* This Java application shows the text 'Hello World!' on  
the screen. */  
package hello;  
  
public class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); //Show the text  
    }  
}
```



Bij het aanmaken van de klasse kan je deze zin aanvinken  
zodanig dat je deze niet zelf moet typen!



# 3. De broncode compileren

**Broncode** (.java) vertalen naar **bytecode** (.class)

- Kan manueel `(javac HelloWorldApp.java)`
- Bij gebruik van een IDE (Eclipse) gebeurt dat automatisch

In map *src*



In map *bin*

= complex (moet je zelf niet kunnen)



## Opdracht 2: Zoek het .class bestand

- M.b.v. het bestandssysteem (Explorer/Verkenner)
- M.b.v. Eclipse



*Window > Show View > Navigator*

## 4. De bytecode uitvoeren

De gecompileerde bytecode wordt uitgevoerd door de interpreter:

- Kan manueel `(java hello.HelloWorldApp)`
- Eclipse: *Run* > *Run*



= complex (moet je zelf niet kunnen)

## Opdracht 3: Voer het programma uit

Voer het programma HelloWorldApp uit in Eclipse.

- Waar zie je de output?
- Wat gebeurt er als je fouten typt in het programma?
  - wijzig println eens in printline



**je kan je fout ook oplossen met 'quick fixes'**



# 5. De opbouw van het programma

## 5.1 Drie soorten commentaar

`/* indien er meerdere regels commentaar  
in een programma geschreven worden */`

`/** documentation JAVADOC */`

`// een regel commentaar`



## 5.2 Het pakket definiëren

```
package hello;
```

Pakketten gebruiken we om orde / structuur te brengen in de code (cfr. mappenstructuur Verkenner)

➔ `..\JavaCourse\src\hello\HelloWorldApp.java`



## 5.3 De klasse definiëren

Hierin schrijf je de eigenlijke code.

## 5.4 De methode main()

```
public static void main(String[] args) {...}
```

Klasse met methode main() = het hoofdprogramma.

*(De details van deze code worden in volgende hoofdstukken toegelicht)*

## 5.5 Het eigenlijke werk

```
System.out.println("Deze tekst wordt  
Letterlijk afgedrukt!");
```



## Opdracht 4: Wijzig het programma

- Voeg volgende regel commentaar toe  
author: naam voornaam
- Wijzig de output in  
Dit was mijn eerste programma



Probeer ook eens uit wat er gebeurt als je  
*syso* <Ctrl> + <Spatie> typt en daarna een  
<Enter>!

