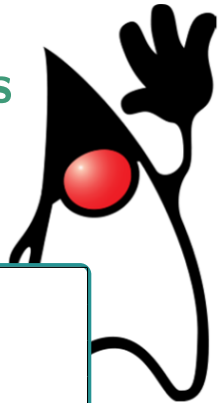


# Programming Fundamentals

## Inleiding










Klasgroep	1EO-ICT
Opleiding	Bachelor Elektronica-ICT
Theorie	DI: 8:45 - 9:45 WOE: 11:45 - 12:45
(Werk)Labo	DI: 9:45 - 12:45 + 13:30 - 14:30 WOE: 13:30 - 15:30 + 15:30 - 17:30
Docent	Katja Verbeeck
Contact	<a href="mailto:katja.verbeeck@odisee.be">katja.verbeeck@odisee.be</a>



# Inhoud


- 1 Praktisch
- 2 Evaluatie
- 3 Lesmateriaal
- 4 Java : hoe werkt het ?
  - Editeren, compileren, uitvoeren
  - Ontwikkelomgeving en installatie
  - Fouten at compile-time
  - Fouten at runtime
  - Stijl
  - Javadoc







 Programming Fundamentals [OGI02q]


     


 Uitleg bij het cursusmenu 


 Mededelingen



 E-BOOK 


 Online theorie les 



 Inhoud


 Opdrachten


 Mijn cijfers



 Feedback 

 Evaluatie

 Discussieruimte 

 ECTS

 Overzicht studiemateriaal


 Tools 

## Mededelingen

Mededeling maken 

### Welkom @ Programming fundamentals

Gepost door: Katja Verbeek

 Gepost: donderdag 15 september 2022 8:00:00 uur CEST

Beste studenten,

Welkom in het vak Programming Fundamentals ! Vanaf volgende week op **dinsdag 20 september** verwachten we de groepen 1 ICT1 tot 1ICT5 in de les - Voor de groepen 1ELO1 tot 1ELO3 en 1ICT6 tot 1ICT7/8 is het wachten op de volgende module die zal starten op **dinsdag 22 november**.

We gaan het hebben over de basisprincipes van programmeren en zullen dit doen aan de hand van de programmeertaal Java.

We organiseren dit als volgt:

- **theorie** : dinsdag : 8u45 - 9u45 - lokaal EO32 + woensdag : 9u45-10u45 - lokaal JOO9
- **labo+ werklabo** : dinsdag 9u45 - 11u45 - D-gang + 13u30-15u30 Cafeteria Zone3  
woensdag: 13u30 - 15u30 D-gang + 15u30-17u30 Cafeteria Zone 3

Julie kunnen je alvast voorbereiden op het eerste labo door vooraf de ontwikkelomgeving IntelliJ te installeren op jullie laptop, zo kunnen we het eerste labo meteen van start gaan.

Tip: How to Install IntelliJ IDEA on Windows 10 + Creating First Hello World Java Application (<https://www.jetbrains.com/help/idea/installation-guide.html> en <https://www.youtube.com/watch?v=EMlTOMdlz4w>).  
Installeer de gratis versie (Community Edition).

alvast veel programmeerplezier,

*Katja Verbeek*

# Teams

The screenshot displays the Microsoft Teams application interface. At the top, a dark blue header bar contains navigation icons, a search bar, and user information. Below the header, a light blue banner indicates that 113 students are waiting to join, with an 'Activate' button. The main content area shows the team's name and a blue icon with a code symbol. Below this, there are tabs for 'Members', 'Pending Requests', 'Channels', 'Apps', and '3 more'. The 'Members' tab is active, showing a search bar and an 'Add member' button. A list of team owners is displayed, including Yves Blancqua..., Evert-Jan Jaco..., Kristien Van A..., Thomas Van d..., Katja Verbeeck, and Mario Wyls. The left sidebar contains various navigation options like Activity, Chat, Teams, Assignments, Calendar, Calls, Files, and Apps.

113 students are waiting to join. [Activate](#)







## Programming Fundamentals [OGI02q] - 2223

[Members](#) [Pending Requests](#) [Channels](#) [Apps](#) [3 more](#)

Search for members

[Add member](#)

**Owners (6)**

Name	Title	Location	Tags	Role
 Yves Blancqua...		KAHOSL_IPTC		Owner
 Evert-Jan Jaco...				Owner
 Kristien Van A...				Owner
 Thomas Van d...				Owner
 Katja Verbeeck				Owner
 Mario Wyls				Owner

**Members and guests (113)**

# Werkwijze

Periode van 7 weken

Onderdelen : Java theorie - Java oefeningen - Java project

Java theorieles 2 x 1u di + woe

Java labo 2 x 2u

Java (zelfstandig)werklabo 2 x 2u De werklabo's zijn **niet vrijblijvend**. Je begeleidende labodocent beslist op basis van je werk of je vrijgesteld bent.

Het Teams kanaal is aanvullend en is de plaats om vragen te posten.

# Begeleidende docenten

## Module 2 :

- **1ICT6(LINFO3)** Evert-Jan Jacobs (evertjan.jacobs@odisee.be)
- **1ICT7(LINFO4)** Katja Verbeeck (katja.verbeeck@odisee.be)
- **1ICT8-1ELO1 (LINFO5)** Kristien Van Assche  
(kristien.vanassche@odisee.be)
- **1ELO2 (LINFO6)** Yves Blanquaert (yves.blancquaert@odisee.be)
- **1ELO (LINFO3-LINFO6)** Thomas Van den Bossche  
(thomas.vandenbossche@odisee.be)

# 1ste examenkans : permanente evaluatie

<b>Theorie (20%)</b> Week 3 Week 5 Week 7	<b>scantesten</b> : korte theoretische denkvragen t.e.m. Java API en prim operaties (4%) t.e.m. iteraties en selecties (6%) alles :- ) (10%)
<b>Praktijk (80%)</b> week 3 week 5 week 7 week 7	<b>Individuele Java oefening</b> op PC + project t.e.m. Java API en prim operaties (10%) t.e.m. iteraties en selecties (20%) alle geziene topics (30%) <b>Project</b> : Java oefening (20%)

# Evaluatie : 2de examenkans

Je beslist zelf welk onderdeel (theorie - individuele oef - project) je hermaakt. Bij voorkeur alle onderdelen waarvoor je niet geslaagd was, en minstens 1 uiteraard.

Theorie (20%)	Toledotest tijdens examenperiode (20%)
Praktijk (80%)	Individuele Java laboproef (60%) + project Java (20%)



# Lesmateriaal

- Alle materiaal : slides, opgaven, testen, ... beschikbaar op Toledo
- Java API online
- Goed boek : **Java A Beginner's Guide** , 6th Edition  
by **Herbert Schildt**  
Publisher: McGraw-Hill Osborne Media  
ISBN: 9780071809252
- Trail: Learning the Java Language, online @  
<https://docs.oracle.com/javase/tutorial/java/index.html>

# Inhoud

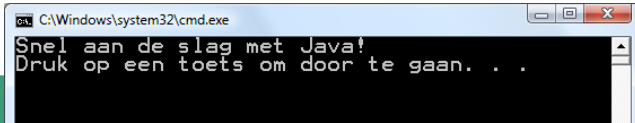
## Java

- First cup of Java
- Primitieve types en operaties
- Java API (oa String, Math, Scanner)
- Selecties
- Iteraties
- Methoden
- Intro OO
- Arrays

# MijnEersteVoorbeeld.java

MijnEersteVoorbeeld.java

```
public class MijnEersteVoorbeeld{  
    public static void main(String[] args){  
  
        System.out.println("Snel aan de slag  
            met Java!");  
    }  
  
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window contains the output of the Java program: 'Snel aan de slag met Java!' followed by a prompt 'Druk op een toets om door te gaan. . .'. The text is displayed in a monospaced font on a black background.

# Stappen

- 1 editeren
- 2 opslaan
- 3 compileren

```
javac MijnEersteVoorbeeld.java
```

```
-> MijnEersteVoorbeeld.class
```

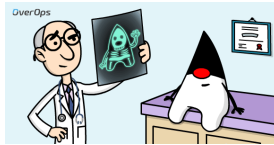
- 4 uitvoeren

```
java MijnEersteVoorbeeld
```

# De Java Virtuele Machine

Java wordt gecompileerd naar bytecode voor een virtuele machine. De Java Virtuele Machine (Java VM of kortweg JVM). Er is een JVM beschikbaar voor allerlei verschillende soorten computers. Hierdoor is de gecompileerde bytecode platformonafhankelijk, d.w.z. de code kan op elke willekeurige computer draaien, onafhankelijk van het besturingssysteem (zoals Windows, Unix of Mac OS X).

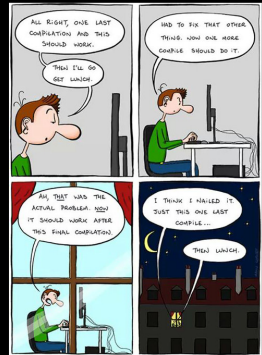
*“Write once, run anywhere”*



Om een gegeven .class bestand te kunnen gebruiken moet wel een compatibele JVM geïnstalleerd zijn.

# Compileren ....

```
99 little bugs in the code,  
99 bugs in the code,  
1 bug fixed...compile again,  
100 little bugs in the code.
```



# Garbage Collection ....

This is Java

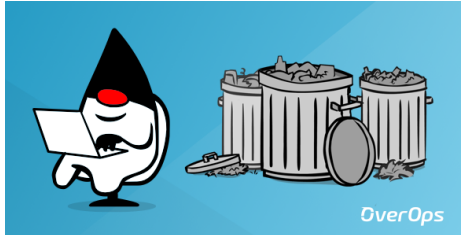


Java does not leave it's  
mess for others to clean.

It collects it's own garbage.

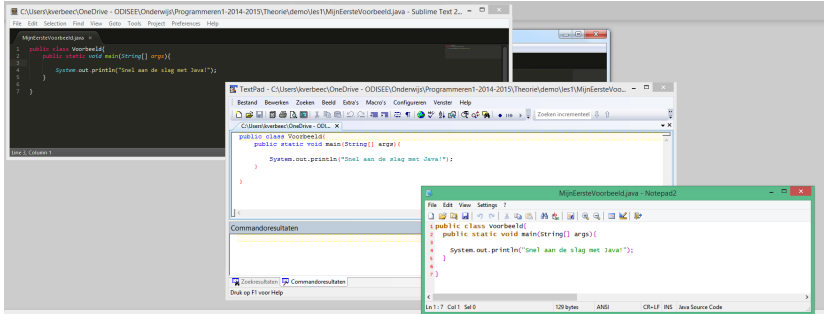
Be like Java.  
Keep your country clean.

@anshimeme's



# Ontwikkelomgeving of IDE

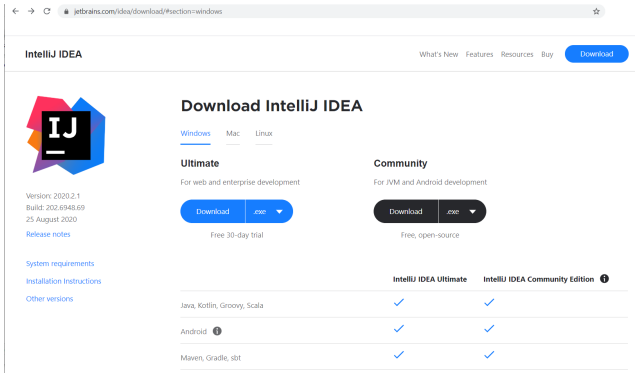
De ontwikkelomgeving of IDE (integrated development environment) kan een simpele tekstverwerker zijn :





# IntelliJ

Wij werken in het labo met de community edition van IntelliJ IDEA



IntelliJ IDEA

What's New Features Resources Buy [Download](#)

## Download IntelliJ IDEA

[Windows](#) [Mac](#) [Linux](#)

### Ultimate

For web and enterprise development

[Download](#) [.exe](#) ▼

Free 30-day trial

### Community

For JVM and Android development

[Download](#) [.exe](#) ▼

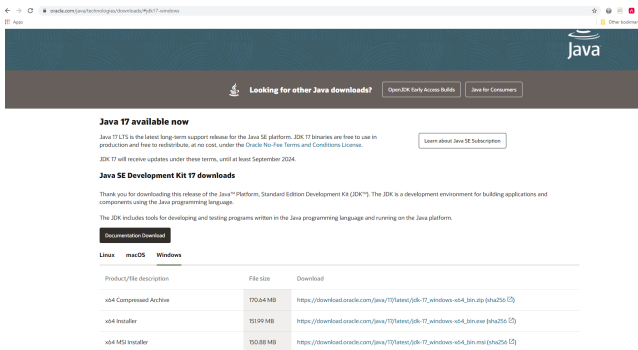
Free, open-source

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition ⓘ
Java, Kotlin, Groovy, Scala	✓	✓
Android ⓘ	✓	✓
Maven, Gradle, sbt	✓	✓

# Java installeren

Je kan rechtstreeks de jdk installeren, zie :

<https://www.oracle.com/java/technologies/downloads/>



The screenshot shows the Oracle Java Downloads page. At the top, there's a navigation bar with the Java logo. Below it, a section titled "Looking for other Java downloads?" includes links for "OpenJDK Early Access Builds" and "Java for Consumers". The main content area features a "Java 17 available now" announcement, stating that Java 17 LTS is the latest long-term support release. Below this, there's a section for "Java SE Development Kit 17 downloads" with a "Learn about Java SE Subscription" link. A "Documentation Download" button is also present. At the bottom, there's a table with download links for Linux, macOS, and Windows. The "Windows" tab is selected, showing three download options: x64 Compressed Archive (170.64 MB), x64 Installer (151.99 MB), and x64 MSI Installer (150.88 MB).

**Java 17 available now**

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions License.

JDK 17 will receive updates under these terms, until at least September 2024.

**Java SE Development Kit 17 downloads**

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

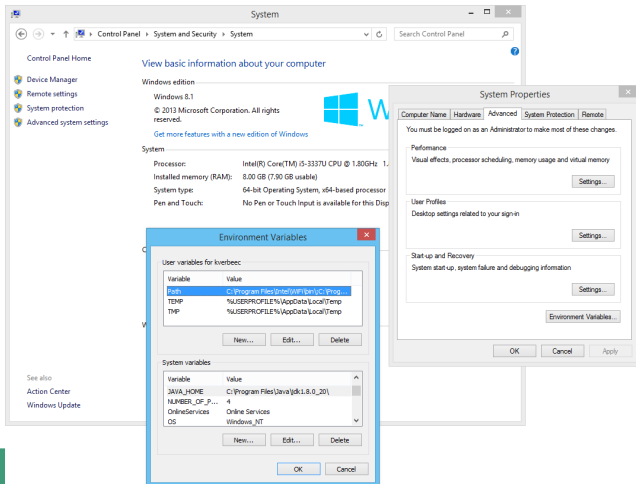
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

**Documentation Download**

**Linux macOS Windows**

Product/file description	File size	Download
x64 Compressed Archive	170.64 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip</a> (sha256)
x64 Installer	151.99 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	150.88 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi</a> (sha256)

# Instellen omgevingsvariabelen (path)



# Oracle Installation Guide

The screenshot shows the Oracle JDK 12 installation guide for Windows. The browser address bar displays the URL: docs.oracle.com/en/java/javase/12/install/installation-jdk-microsoft-windows-platforms.html#GUID-96EB3E76-BC7A-4A25-9F3A-A298FEC016A. The page title is "Table of Contents". The left sidebar contains a table of contents with the following items: "Title and Copyright Information", "Preface", "1 Overview of JDK Installation", "2 Version-String Format", "3 Installation of JDK on Oracle Solaris", "4 Installation of the JDK on Linux Platforms", "5 Installation of the JDK on macOS", "6 Installation of the JDK on Microsoft Windows Platforms", "System Requirements for Installing the JDK on 64-Bit Windows Platform", "JDK Installation Instruction Notation for Windows", "JDK Installation Instructions for Windows", "Downloading the JDK Installer", "Running the JDK Installer", "Installing the JDK Silently", "Setting the PATH Environment Variable" (highlighted), "Beginning to Use the JDK", "Uninstalling the JDK on Windows", "JDK Installation Troubleshooting", and "7 Installed Directory Structure of JDK". The main content area is titled "Setting the PATH Environment Variable". It states: "It is useful to set the PATH variable permanently for JDK 12 so that it is persistent after rebooting. If you do not set the PATH variable, then you must specify the full path to the executable file every time that you run it. For example: C:\> "C:\Program Files\Java\jdk-12\bin\javac" MyClass.java". It then says: "To set the PATH variable permanently, add the full path of the jdk-12\bin directory to the PATH variable. Typically, the full path is C:\Program Files\Java\jdk-12\bin". It then says: "To set the PATH variable on Microsoft Windows: 1. Select **Control Panel** and then **System**. 2. Click **Advanced** and then **Environment Variables**. 3. Add the location of the bin folder of the JDK installation to the PATH variable in **System Variables**." A note box contains the following text: "Note: The PATH environment variable is a series of directories separated by semicolons (;) and is not case-sensitive. Microsoft Windows looks for programs in the PATH directories in order, from left to right. You should only have one bin directory for a JDK at a time. Those following the first instance are ignored. If you are not sure where to add the JDK path, append it. The new path takes effect in each new command window that you open after setting the PATH variable."

# Installatie IntelliJ

- How to Install IntelliJ IDEA on Windows 10 + Creating First Hello World Java Application  
(<https://www.youtube.com/watch?v=EMLTOMdIz4w>)
- Creating your first Java application with IntelliJ IDEA (2020)  
([https://www.youtube.com/watch?v=H\\_XxH66lm3U](https://www.youtube.com/watch?v=H_XxH66lm3U))

# Fouten at compile-time = syntaxfouten

Naam van een klasse

Hallo.java

```
public class hallo {  
    public static void main(String[] args) {  
        System.out.println("Dag 1 EO-ICT !!!");  
    }  
}
```

# Fouten at compile-time = syntaxfouten

Naam van een klasse

```
C:\java\>javac Hallo.java
Hallo.java:1: class hallo is public, should be declared in a file
named hallo.java

public class hallo {
            ^
1 error
```

# Fouten at compile-time = syntaxfouten

Naam van een klasse

Hallo.java

```
public class Hallo {  
    public static void main(String[] args) {  
        System.out.println("Dag 1 EO-ICT !!!");  
    }  
}
```



# Fouten at compile-time = syntaxfouten

Naam van een klasse

- Java is case-sensitive
- bestandsnaam = naam van de klasse

# Fouten at compile-time = syntaxfouten

Hallo.java

```
public class Hallo {  
    public static void main(String[] args) {  
        System.out.println("Dag 1 EO-ICT !!!")  
    }  
}
```

# Fouten at compile-time = syntaxfouten

```
C:\java\>javac Hallo.java
Hallo.java:3: ';' expected
    System.out.println("Hallo 1ICT!")
                                ^
1 error
```

# Fouten at runtime = semantische fouten

KennisMaking.java

```
import java.util.Scanner;
public class KennisMaking {

    public static void main(String [] args) {
        System.out.println("Zeg, geef eens je naam :");
        Scanner scan = new Scanner(System.in);
        String naam = scan.next();
        // variabele naam van type String
        System.out.println("Dag "+ naam);
        System.out.println("Enne, hoe oud ben jij
            eigenlijk?");
        int leeftijd = scan.nextInt();
        // variabele leeftijd van type int
    }
}
```

# Fouten at runtime = semantische fouten

The screenshot shows the IntelliJ IDEA IDE with a project named 'KennisMaking'. The 'Project' view on the left shows the directory structure: 'KennisMaking' (containing 'idea', 'out', 'src', and 'src' subdirectories), 'External Libraries', and 'Scratches and Consoles'. The 'src' directory is highlighted, showing 'odisee.prog.theorie' (100% classes, 85% lines covered) and 'KennisMaking' (100% methods, 85% lines covered). The 'Run' view at the bottom shows the execution of the 'KennisMaking' class. The output is as follows:

```
Run: KennisMaking (1) x
C:\Users\katja.verbeeck\.jdk\openjdk-14.0.2\bin\java.exe -javaagent:C:\Users\katja.verbeeck\AppData\Local\JetBrains\Ide
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
odisee\prog\theorie\...
exclude patterns:
Zeg, geef eens je naam :
Katja
Dag Katja
Enne, hoe oud ben jij eigenlijk?
euhhh
Exception in thread "main" java.util.InputMismatchException: Create breakpoint
```

The code in the main editor is as follows:

```
public class KennisMaking {
    public static void main(String[] args) {
        System.out.println("Zeg, geef eens je naam :");
        Scanner scan = new Scanner(System.in);
        String naam = scan.next();
        // variabele naam van type String
        System.out.println("Dag "+ naam);
        System.out.println("Enne, hoe oud ben jij eigenlijk?");
        int leeftijd = scan.nextInt();
        // variabele leeftijd van type int
    }
}
```

# Stijl

variabelen steeds met kleine letter! laat een blok code inspringen !

```
import java.util.Scanner;
public class KennisMaking2 {

    public static void main(String [] args) {
        System.out.println("Zeg, geef eens je naam :");
        Scanner SCAN = new Scanner(System.in); // FOUT !
        String NAAM = SCAN.next(); // FOUT !!
        // variabele naam van type String
        System.out.println("Dag "+ NAAM);
        System.out.println("Enne, hoe oud ben jij
            eigenlijk?");
        int leeftijd = SCAN.nextInt();
        // variabele leeftijd van type int
    } // Gebruik indentatie !!
}
```

# Stijl : indentatie !

There are two types of people:

```
if (Condition) {  
    Statement  
    /* ....  
    */  
}
```

```
if (Condition)  
{  
    Statement  
    /* ....  
    */  
}
```

# Stijlfouten

- Stijlregel 1 : Naam van een klasse steeds met een Hoofdletter beginnen! Is ook het geval voor **ALLE** andere klassen in de java klassenbibliotheek!!! – (bvb. System, Math, String, ...)
- Stijlregel 2 : Laat elke Java methodenaam beginnen met een kleine letter – Is ook het geval voor **ALLE** andere methoden in de java klassenbibliotheek!!! – (bvb. println, ... )
- Stijlregel 3 : Laat elke Java variabele beginnen met een kleine letter
- Stijlregel 4 : Gebruik CamelCasing voor lange namen : klasse **KennisMaking**, **EnergieVerbruik**, methode **beschrijfJezelf**, ...
- er volgen nog stijlregels voor indentatie, constanten, ...



# Commentaar

```
/*  
    Commentaar Stijl 1 : meerdere lijnen  
    Dit is een eenvoudig Java voorbeeld.  
*/  
  
public class Hallo {  
    // Commentaar Stijl 2 : slechts 1 lijn  
  
    public static void main(String[] args) {  
        System.out.println("Dag 1 EO-ICT !!!");  
    }  
}
```

# Javadoc

Javadoc is een programma, bijgeleverd bij de jdk, waarmee je html-documentatie pagina's kan genereren gebruik makend van **javadoc** tags. Merk op, de volledige java API is eveneens op deze manier gedocumenteerd : <https://docs.oracle.com/javase/8/docs/api/> zie ook boek, appendix B

De tags die wij alvast hanteren voor elke gemaakt oefening zijn **@author** en **@version** om de header van elk programma te beschrijven. Later zullen meer tags gebruikt worden.

https://docs.oracle.com/javase/8/docs/api/

Overview PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

## Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

### Profiles

- compact1
- compact2
- compact3

### Packages

Package	Description
<b>java.applet</b>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<b>java.awt</b>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<b>java.awt.color</b>	Provides classes for color spaces.
<b>java.awt.datatransfer</b>	Provides interfaces and classes for transferring data between and within applications.
<b>java.awt.dnd</b>	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
<b>java.awt.event</b>	Provides interfaces and classes for dealing with different types of events fired by AWT components.
<b>java.awt.font</b>	Provides classes and interface relating to fonts.
<b>java.awt.geom</b>	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
<b>java.awt.im</b>	Provides classes and interfaces for the input method framework.
<b>java.awt.im.spi</b>	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
<b>java.awt.image</b>	Provides classes for creating and modifying images.
<b>java.awt.image.renderable</b>	Provides classes and interfaces for producing rendering-independent images.
<b>java.awt.print</b>	Provides classes and interfaces for a general printing API.

```
/**
 * Deze klasse print een welkomstwoord
 * op het scherm
 * @author Katja Verbeeck
 * @version september 2015
 */

// let op een javadoc header start steeds
// met /** en eindigt met */

public class Header {
    public static void main(String[] args) {
        System.out.println(" De eerste
                           javadoc-tags ");
    }
}
```

# Javadoc header in IntelliJ

