

# Java

## Les 12

[jeroen.devos01@ap.be](mailto:jeroen.devos01@ap.be)

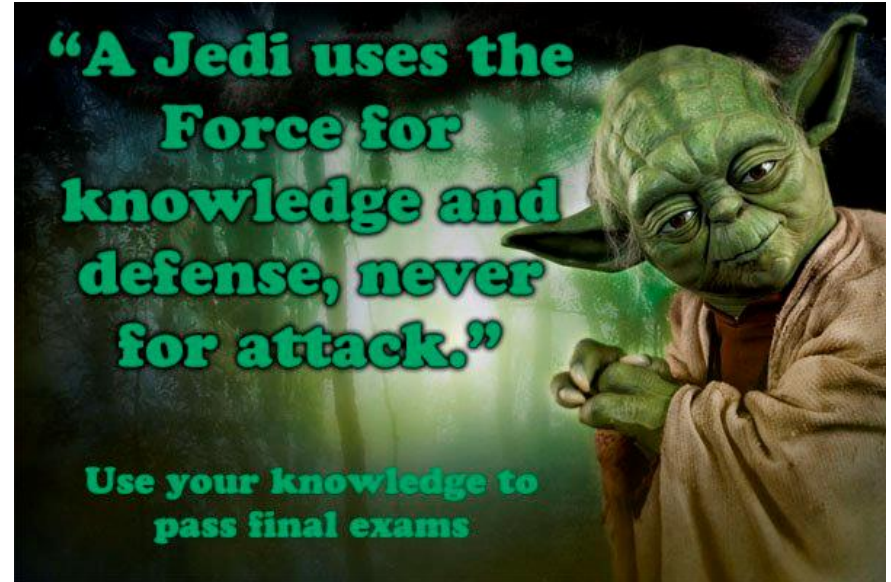
# Lesdoelen

- Examen informatie
- Leerstof overzicht
- Bevraging
- Q&A
- Oefenexamen



# Examen informatie

- Maandag 08 / 01 / 2023
- 09u00 – 12u00
- VIA 01.27
- Digitale test
- ⇒ Te laat = GEEN EXAMEN
- ⇒ 09u01 = te laat



# Examen (semi) open boek

## Toegestaan:

- Cursus en eigen notities
- Modeloplossingen, eigen oefeningen, voorbeeldexamen
- Websites:
  - **oracle.com**
  - **stackoverflow.com**
  - **w3schools.com**
  - **baeldung.com**
  - **geeksforgeeks.org**
  - **google translate**

## Niet toegestaan:

- **AI TOOLS**
  - Github copilot
  - IntelliCode
  - Tabnine
  - ChatGPT (ook bing)
  - Grepper
  - ... (en alle anderen)
- Vorige examens (is geen eigen oefening)
- Communicatiemiddelen
- Alle andere websites...

# Examen informatie



# Examen informatie

## OPGELET

- Examenvragen dupliceren of verspreiden is **fraude**.
- Examenoplossingen van vorige jaren gebruiken is **fraude**.

# Examen informatie

- Geen theorie, enkel praktijk
- Eén of meerdere oefeningen
  - Elke oefening is een apart Maven project
  - Elke oefening apart indienen
- Examen op 20
- Gebruik van boetebblad

# Boeteblad

- Project bevat nog fouten en compileert niet: - 1
- Iedere klasse niet in apart bestand: -1
- Veel onnodige of redundante code (geen goed gebruik loops, methoden of klassen, ongebruikte imports) : -3
- Bladspiegel overdreven slecht : -1
- Naamgeving van methoden en variabelen niet conform conventies : -2
- Naamgeving niet consistent (Engels en Nederlands door elkaar) : -1
- Geen zip van project, enkel codefile(s) ingeleverd of ander formaat als zip : -3
- Gebruik util.Date : -2



# Java Naming Conventions

Het correct toepassen van **naming conventions** onderscheidt de professional van de amateur !

<b>package</b>	lowercase	com.example
<b>class / interface</b>	PascalCase	AnimalProperties
<b>method</b>	camelCase	makeSound()
<b>attribute</b>	camelCase	dateOfBirth
<b>constant</b>	SCREAMING_SNAKE_CASE	MAX_AGE



# Maven

## Folder structure of Maven

- `src/main/java`
- `target`
- Pom sections
- Goals
- Dependency
- `~/.m2/repository`
- Defaults



# OOP

## Object Oriented Programming in Java



[www.educba.com](http://www.educba.com)

# Dates

Date-time types in Java & SQL	Legacy class	Modern class	SQL standard data type
Moment in <b>UTC</b>	java.util. <b>Date</b> java.sql. <b>Timestamp</b>	java.time. <b>Instant</b>	TIMESTAMP WITH TIME ZONE
Moment with <b>offset-from-UTC</b> ( hours-minutes-seconds )	( lacking )	java.time. <b>OffsetDateTime</b>	TIMESTAMP WITH TIME ZONE
Moment with <b>time zone</b> ( `Continent/Region` )	java.util. <b>GregorianCalendar</b> javax.xml.datatype. <b>XMLGregorianCalendar</b>	java.time. <b>ZonedDateTime</b>	TIMESTAMP WITH TIME ZONE
Date & Time-of-day ( no offset, no zone ) <b>Not a moment</b>	( lacking )	java.time. <b>LocalDateTime</b>	TIMESTAMP WITHOUT TIME ZONE
Date only ( no offset, no zone )	java.sql. <b>Date</b>	java.time. <b>LocalDate</b>	DATE
Time-of-day only ( no offset, no zone )	java.sql. <b>Time</b>	java.time. <b>LocalTime</b>	TIME WITHOUT TIME ZONE
Time-of-day with offset ( impractical, not used )	( lacking )	java.time. <b>OffsetTime</b>	TIME WITH TIME ZONE

JDBC 4.2

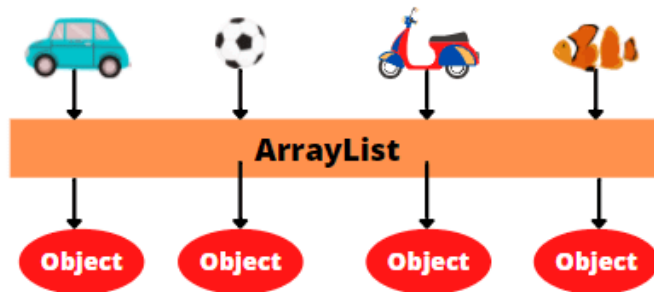
JDBC 4.2

# Generics

## WITHOUT GENERICS

Objects go IN as a reference to Car, Football, Scooter, and Fish objects

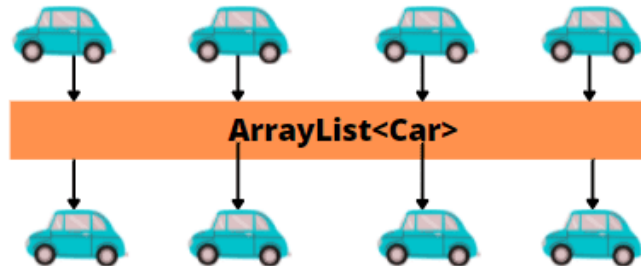
And come OUT as a reference of type Object.



## WITH GENERICS

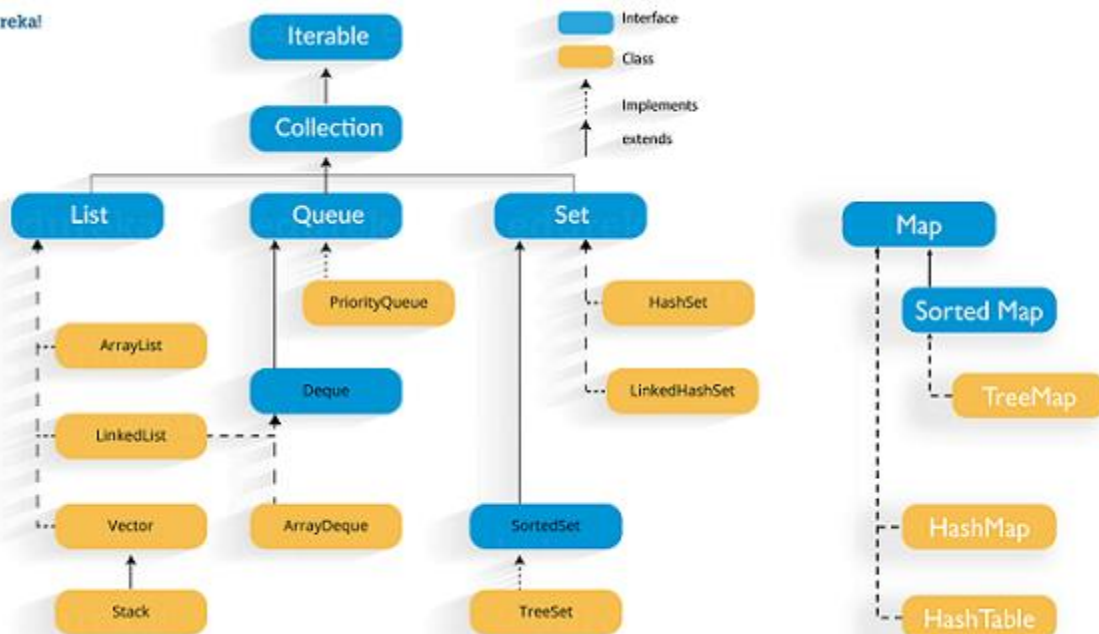
Objects go IN as a reference to only Car objects

And come OUT as a reference of type Car.



# Collections

edureka!



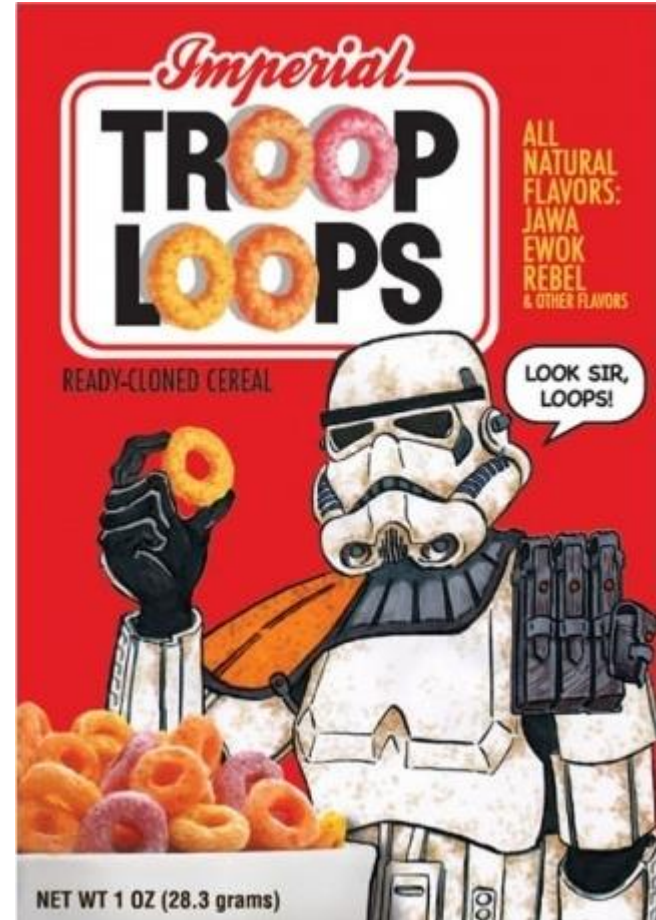
# Loops

Vier soorten :

- **for** loop
- **for each** loop
- **while** loop
- **do while** loop

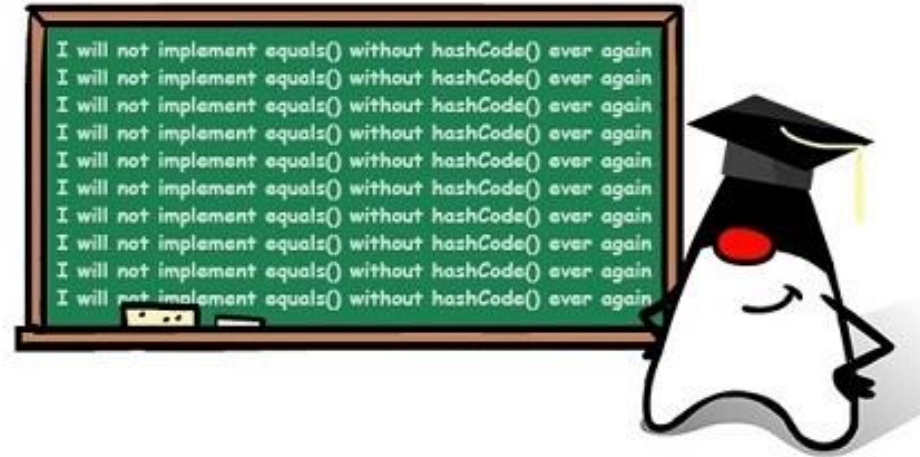
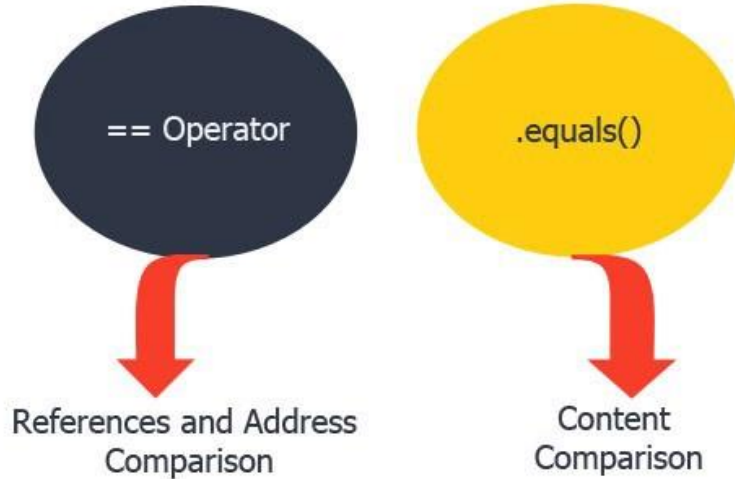
Keywords:

- break
- continue





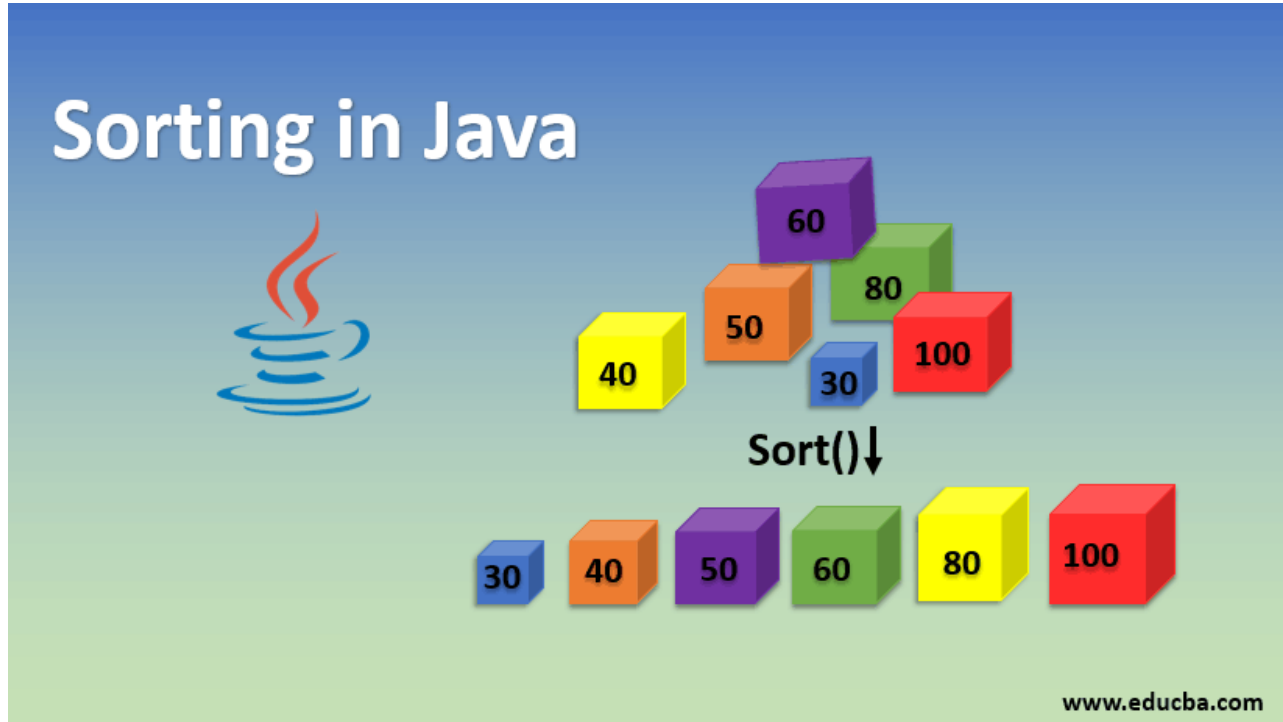
# equals & hashCode



# Comparable & Comparator



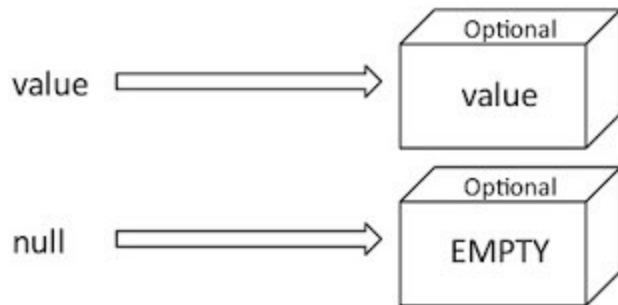
# Sorting



# Optional

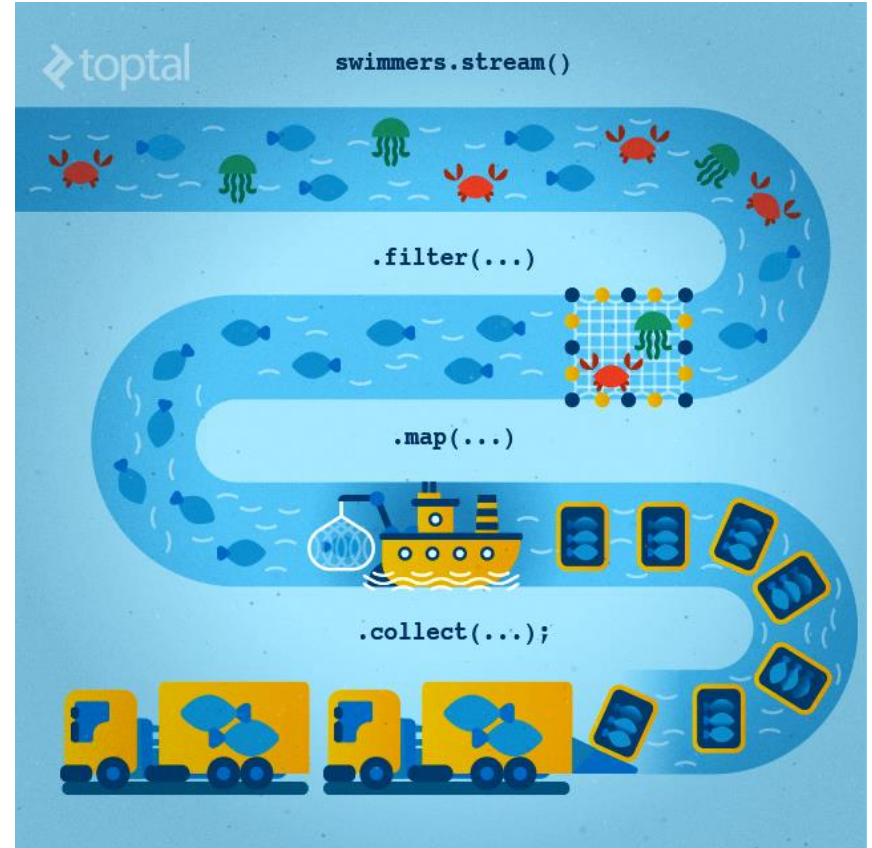
## Replacing nulls with Optionals

If nulls are so problematic why don't we just avoid them?



Optional is a type that models a possibly missing value

# Lambda & Stream



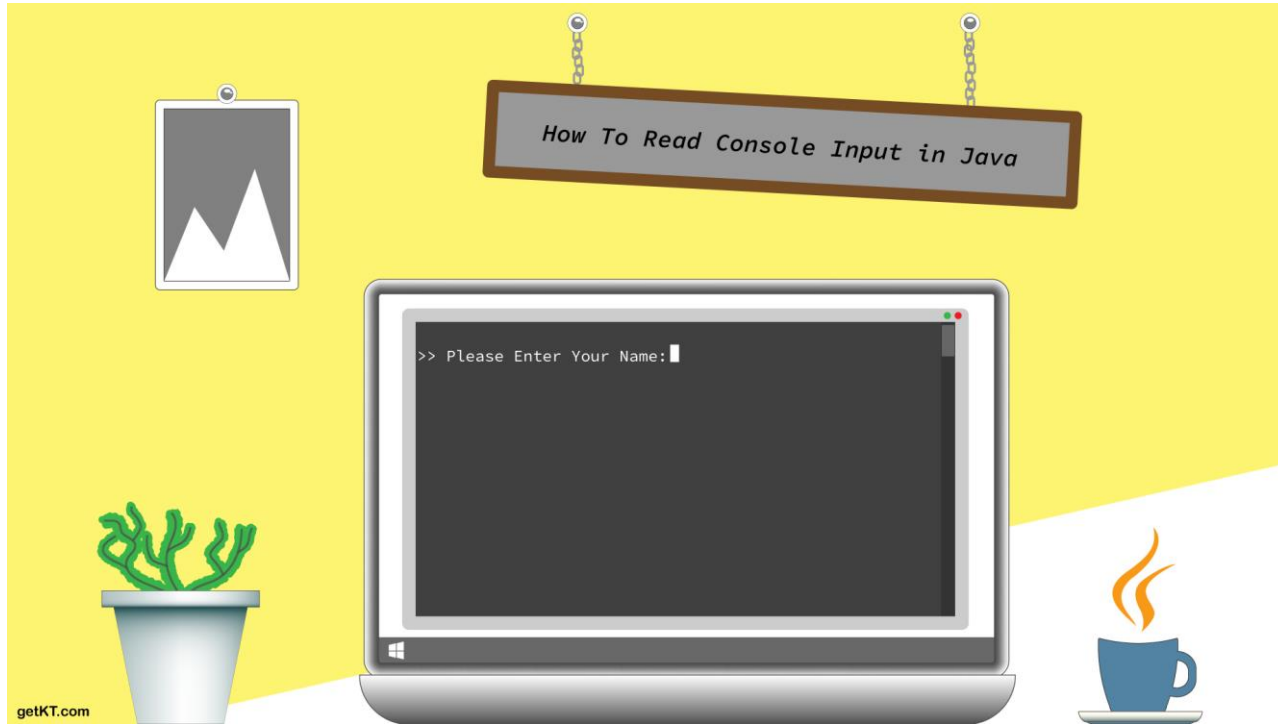
# File handling



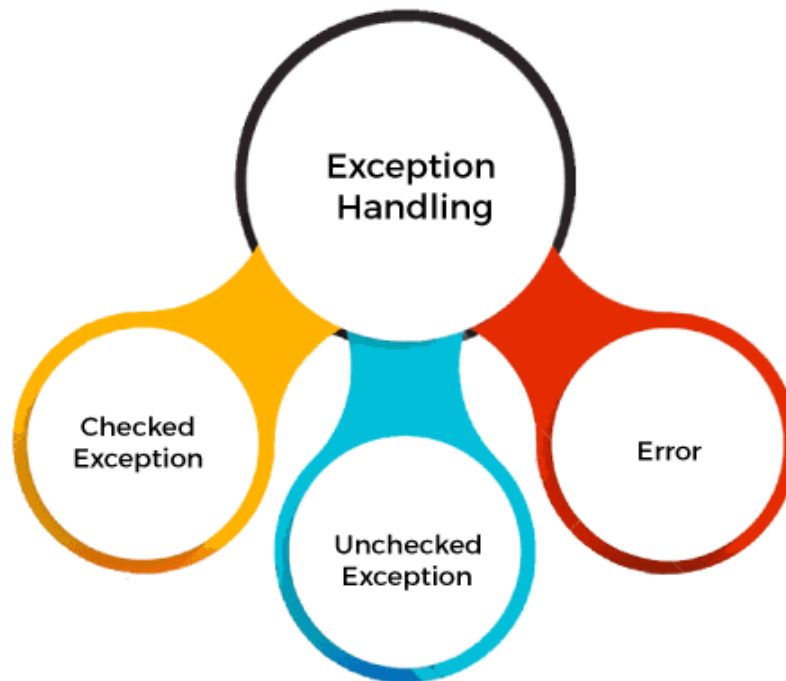
File IO with Java 8  Streams

nextptr

# Console : read / write



# Exceptions





# Stack trace lezen

How to read and  
understand a  
Java Stacktrace



# Annotations

@annotation



# Try with resources

```
try(  
    java.lang.AutoCloseable  
    java.io.Closeable  
) {  
    use_resources  
} catch (Exception ex) {  
    ex.printStackTrace();  
}
```

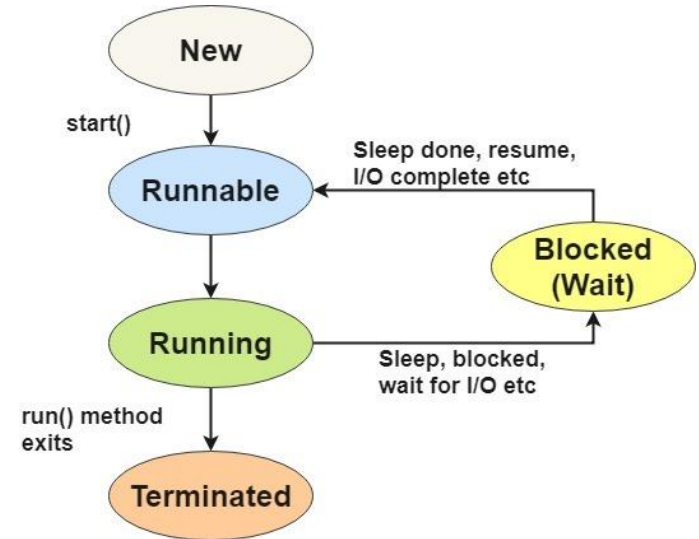
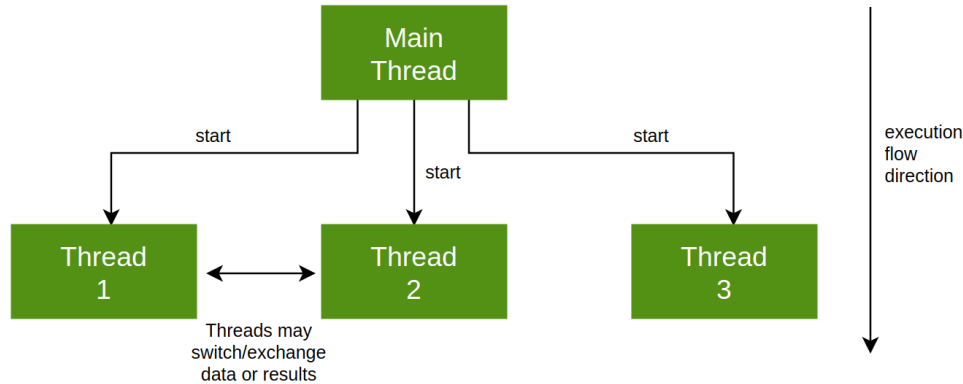


# Hibernate



# Threads / multithreading

## Multithreading Programming



# JavaDoc

[PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

[SUMMARY: NESTED | FIELD | CONSTR | METHOD](#) [DETAIL: FIELD | CONSTR | METHOD](#)

com.baeldung.javadoc

**Class SuperHero**

java.lang.Object  
com.baeldung.javadoc.Person  
com.baeldung.javadoc.SuperHero

```
public class SuperHero
extends Person
```

Hero is the main entity we will be using to ...

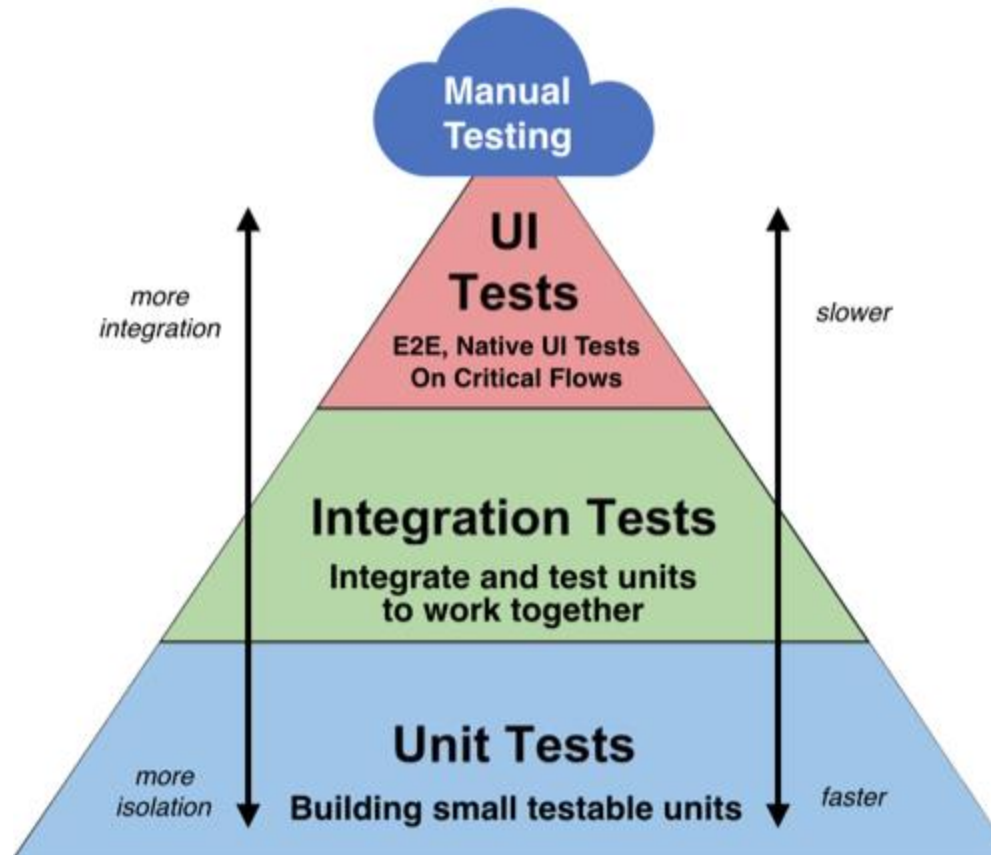
Author:  
Captain America

**Field Summary**

**Fields**

Modifier and Type	Field and Description
private int	<b>defense</b>
private int	<b>health</b>
private java.lang.String	<b>heroName</b> The public name of a hero that is common knowledge
private java.lang.String	<b>uniquePower</b>

# Testing



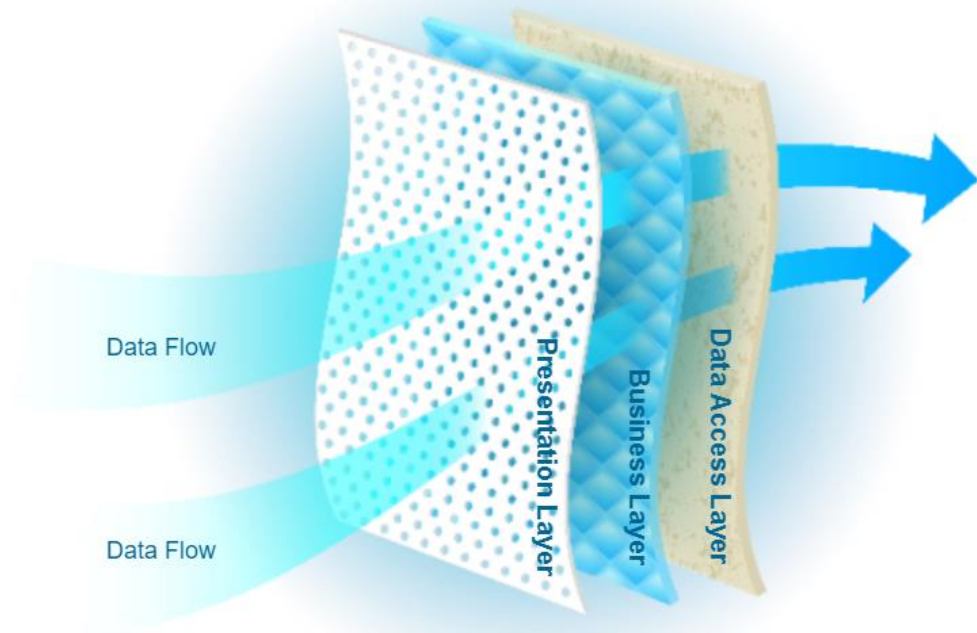
Log4J

**LOG4J**





# Layered architecture



# JAX-RS / Jersey / Jackson / Jetty





# Tips

- Lees de opgave grondig
- Doe eerst wat je goed kan
- Let op je code (boetebblad)
- Zorg ervoor dat je code compileert / werkt.  
Als je bepaalde zaken niet volledig aan de praat krijgt, zet deze in commentaar en schrijf erbij wat je geprobeerd hebt en waar je vermoedt dat het misloopt.

# Vrijblijvende bevraging

**OLOD bevraging (zie mail)**  
(vul dat nu in)





# Oefenexamen

Drie vragen:

1. Hibernate & file import & console
2. Sortering & streams
3. JavaDoc & unit testing

# Oefenexamen – hoe begin ik eraan ?

## 1. Time boxing

Blijf niet te lang plakken op een vraag... je hebt 'maar' 3u tijd. Moet je teveel opzoeken, ga naar de volgende vraag.

## 2. Doe eerst wat je goed kent / begin met de kleinere vragen.

## 3. Gebruik je oefeningen als inspiratie, kopieer niet zonder nadenken.



