



# Lektionstillfälle 1

"Introduktion Java"

**NACKADEMIN**

# Lektionstillfällets mål och metod

## Mål med lektionen:

- Kursmål
- GitLab
- Installera Java & Utvecklingsmiljö
- Introduktion till Java

## Lektionens arbetsmetod/er:

- Föreläsning
- Labb

# Kurslitteratur (Frivilligt)

Rekommenderad bok:

- Java in a Nutshell, 7th edition.
  - Skriven för Java 11 LTS

Alternativ bok:

- Head First Java.
  - Detta är en gammal bok som utgår ifrån Java 5.0
  - Den är bra för generell programmeringskunskap och om ni gillar bilder och alternativa sätt att lära.
  - Det kommer snart en uppdaterad utgåva (efter denna kurs tyvärr)

**NACKADEMIN**

# Begreppsgenomgång

Java – Ett programmeringsspråk som Python

Dynamiskt typat / Statiskt typat

Starkt Typat – Python & Java

Git Repository - GitLab & GitHub

JRE - Java Runtime Environment

JDK - Java Development Kit

Oracle Java

OpenJDK - Öppen java

LTS – Long Term Support

# Typning

Dynamiskt typat – Python typer sätts när programmet körs

Statiskt typat – Java Typer måste sättas, String str = "hello"

Båda är "Starkt typat" – Språket tar hänsyn till typer, konvertering mellan typer är styrt och kan under vissa omständigheter ske automatiskt.

# Java VS Python

- Java är statiskt typat

```
String message = "hello world"
```

```
// Sätts tills String i koden innan det kompileras & körs
```

Python är dynamiskt typat

```
message = "hello world"
```

```
# Sätts till typen String i "runtime" när programmet körs
```

# Java & Python är starkt typat

- Java är starkt typat

```
String message = 1
```

```
// Type mismatch: cannot convert from int to String
```

Python är starkt typat

```
message = "hello world"
```

```
print(message + 1.0)
```

```
TypeError: can only concatenate str (not "float") to str
```

**NACKADEMIN**

# Java Vs Python

- Java är från början flertrådat
- Python har trådar, men användningen är i princip enkeltrådat genom GIL låsning. Det går istället att använda multiprocessing etc, men det kräver då ramverk för att göra det smidigt.
- Java har primitiva typer och referens typer
- I Python är allt objekt så det finns bara referens typer




# Hårt typat Java

```
lesson_1 > examples > java > Strong.java > ...  
1  ∨ public class Strong {  
2  
3  ∨  
4  public static void main(String[] args) {  
5      int number = 7;  
6      String person = "Pelle";  
7      System.out.println(number - person);  
8  }  
9  
10 }
```

```
rober ~/repos/devops21/devops21_java/lesson_1/examples/java main javac Strong.java  
Strong.java:6: error: bad operand types for binary operator '-'  
    System.out.println(number - person);  
                        ^  
first type:  int  
second type: String  
1 error
```

# Starkt typat Python

```
lesson_1 > examples > python >  strong.py > ...
```

```
1   person = "Pelle"  
2   number = 7  
3  
4   print(number - person)  
5
```

```
Traceback (most recent call last):
```

```
File "/Users/rober/repos/devops21/devops21_java/lesson_1/examples/python/strong.py", line 4, in <module>  
    print(number - person)
```

```
TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

**NACKADEMIN**

# Löst typat JavaScript

```
lesson_1 > examples > javascript > JS weak.js > ...
```

```
1  var person = "Pelle"  
2  var number = 7  
3  
4  console.log(number - person);  
5  // expected output: NaN
```

# Varför Java? Hur populärt är java?

- Tiobe index
  - Sammanvägt index <https://www.tiobe.com/tiobe-index/>
- PYPL Popularity of Programming Language index
  - Hur ofta tutorials är sökta på google för varje språk <https://pypl.github.io/PYPL.html>
- <https://octoverse.github.com/>
  - De mest använda programmeringspråken på GitHub

# Fördelar med Java

- Java är stabilt över tid
- Har traditionellt inte ändrats på ett sådant sätt att applikationer behöver skrotas eller skrivas om helt
- Ett öppet landskap, men ändå ägt av stora bolag som driver det
- Lång historik, skapades 1995
- Stort antal mogna projekt och paket finns att importera
- "Cross Platform", stödjer många plattformar <https://www.oracle.com/java/technologies/javase/products-doc-jdk17certconfig.html>

# Stort ekosystem

- Java för Android
- Java EE numera Jakarta EE
  - [https://en.wikipedia.org/wiki/Jakarta\\_EE](https://en.wikipedia.org/wiki/Jakarta_EE)
- Kotlin
- Scala
- Groovy (används av Jenkins)
- Spring

# Stort ekosystem

- <https://www.infoq.com/news/2021/07/snyk-jvm-2021/>
- <https://www.jrebel.com/blog/2021-java-technology-report>
- <https://github.com/docker-library/docs/blob/master/openjdk/README.md#supported-tags-and-respective-dockerfile-links>
- <https://www.jetbrains.com/idea/devecosystem-2021/java/>
- <https://snyk.io/jvm-ecosystem-report-2021/>

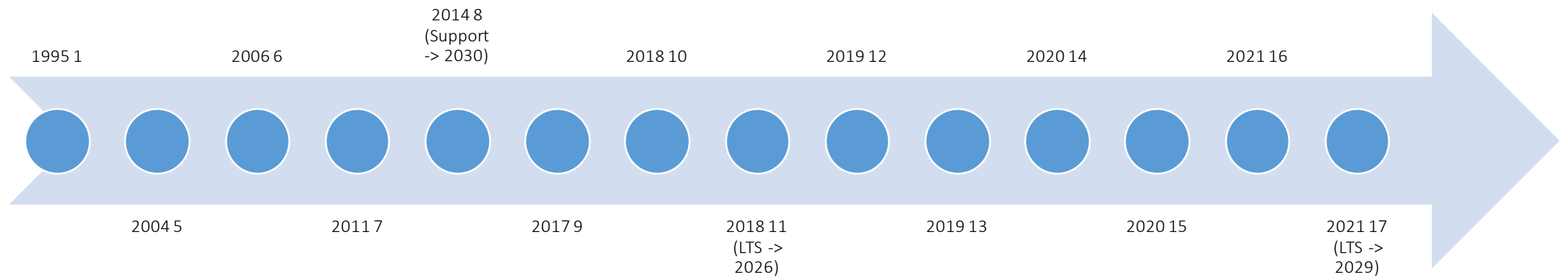
# Varför version använder vi 11/17?

- Vissa företag kör alltid det senaste (17+)
  - Vissa företag kör bara LTS eller supporterade versioner 8/11/17
  - Vissa företag kör vidare utan att uppdatera något (inte bra)
  - Vissa företag har ingen aning, det kan vara allt möjligt
- 
- Början ni på nytt och har ingen uttalad strategi för uppdateringar? Välj senaste LTS
  - Finns kod sedan tidigare? Då måste ni undersöka vilken LTS som fungerar.

**NACKADEMIN**



# Java historik



[https://en.wikipedia.org/wiki/Java\\_version\\_history](https://en.wikipedia.org/wiki/Java_version_history)

**NACKADEMIN**

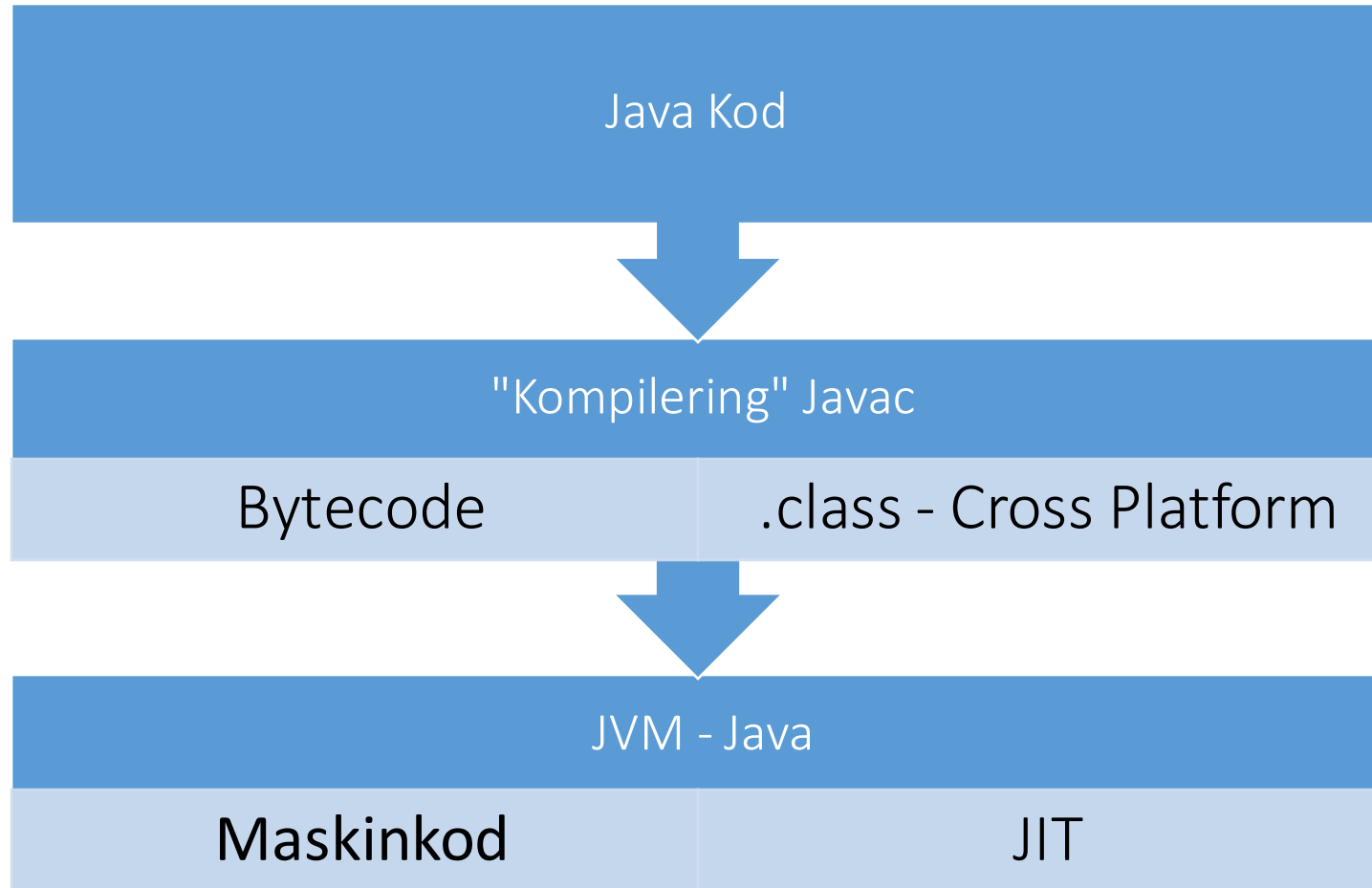
# Java Long Term Support

Oracle Java SE Support Roadmap <sup>*†</sup>				
Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Support
7 (LTS)	July 2011	July 2019	July 2022*****	Indefinite
8 (LTS)**	March 2014	March 2022	December 2030*****	Indefinite
9 (non-LTS)	September 2017	March 2018	Not Available	Indefinite
10 (non-LTS)	March 2018	September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	September 2026	Indefinite
12 (non-LTS)	March 2019	September 2019	Not Available	Indefinite
13 (non-LTS)	September 2019	March 2020	Not Available	Indefinite
14 (non-LTS)	March 2020	September 2020	Not Available	Indefinite
15 (non-LTS)	September 2020	March 2021	Not Available	Indefinite
16 (non-LTS)	March 2021	September 2021	Not Available	Indefinite
17 (LTS)	September 2021	September 2026****	September 2029****	Indefinite
18 (non-LTS)***	March 2022	September 2022	Not Available	Indefinite
19 (non-LTS)***	September 2022	March 2023	Not Available	Indefinite
20 (non-LTS)***	March 2023	September 2023	Not Available	Indefinite
21 (LTS)***	September 2023	September 2028	September 2031	Indefinite

<https://www.oracle.com/java/technologies/java-se-support-roadmap.html>

**NACKADEMIN**

# Java, javac, jvm



# Java, javac, jvm

- Vad motsvarar detta i python?
- `__pycache__`
- `.pyc`
- <https://en.wikipedia.org/wiki/Bytecode>
- <https://medium.com/swlh/a-performance-comparison-between-c-java-and-python-df3890545f6d>

# Flera språk kör på JVM

Java, Kotlin, Clojure, Scala

```
graph TD; A[Java, Kotlin, Clojure, Scala] --> B[Compilers -> .class]; B --> C[JVM];
```

The diagram consists of three blue rectangular boxes stacked vertically. The top box contains the text 'Java, Kotlin, Clojure, Scala'. A blue arrow points from the bottom center of this box to the top center of the middle box. The middle box contains the text 'Compilers -> .class'. Another blue arrow points from the bottom center of the middle box to the top center of the bottom box. The bottom box contains the text 'JVM'.

Compilers -> .class

JVM

**NACKADEMIN**

# Summering av dagens lektion

- Java och JVM:en är de facto en av världens viktigaste plattformar
- Det finns många intressenter i java
  - Android
  - Java
  - Jakarta EE
- Java är statiskt typat och starkt typat
- Java för DevOps är mer än att bara koda
  - Support / Uppdateringar
  - Säkerhet (Aktuellt nu t.ex. log4shell)
  - Kända verktyg såsom Jenkins, Tomcat, JetBrains