

# Informatica: Java reeks 1

van Python naar Java

Ruben De Smet

2<sup>de</sup> semester 2021

**WPO's** Ruben De Smet — rdesmeta@etrovub.be — K4.56  
Tobias Birnbaum — tbirnbau@etrovub.be — PL9.1.63  
Colas Schretter — cschrett@etrovub.be — PL9.2.26

**Project** spel maken

**Website** `parallel.vub.ac.be` → Education → Java programming → Practica

**Entropy** dezelfde oefeningen in Java.

Vijf oefeningenreeksen met extra's. Slides staan online.

Week	Omschrijving
23	oefeningen
24	oefeningen + projectuitleg
25	oefeningen + groepen en projectbeschrijving
26	project
27	oefeningen
28	project
29	oefeningen
30, 31	lentevakantie
32	project
33	deadline
34	projectevaluatie

# JAVA VERSUS PYTHON

## VERSCHILLEN

Schaalbaarheid:

- ▶ Python typisch gebruikt voor kleine **scripts**.
- ▶ Java typisch voor (heel) grote (vaak commerciële) **projecten**.

Typering

- ▶ Python is dynamisch getypeerd.
- ▶ Java is statisch getypeerd.

Java blokken in {}, Python blokken met verplichte indentatie.

Erg sterke focus op **portabiliteit** bij Java: “compile once, deploy anywhere”.

Bij Python gebruikten we Spyder

- ▶ simpel
- ▶ gemakkelijk
- ▶ eenvoudig

---

<sup>1</sup>...en later Eclipse.

Bij Python gebruikten we **Spyder**

- ▶ simpel
- ▶ gemakkelijk
- ▶ eenvoudig

Voor Java gebruiken we vandaag **Jupyter Notebooks**<sup>1</sup>

- ▶ Geen installatie, ga naar `https://jupyter-production.opencloudedge.be`,  
login met **VUB account**.
- ▶ Makkelijk om een taal te leren kennen.
- ▶ Slaat op op de server.
- ▶ Veel gebruikt in industrie met Python.

---

<sup>1</sup>...en later Eclipse.

# JAVA VERSUS PYTHON

## SYNTAX

```
1  int x;  
2  x = 3;  
3  int y = 5;  
4  boolean z = true;  
5  
6  if (x > 100) {  
7      System.out.println("x is groter dan 100");  
8  } else {  
9      System.out.println("x is gelijk aan " + x);  
10 }
```

# JAVA VERSUS PYTHON

## INDENTATIE

Indentatie is belangrijk...maar **niet voor het programmaverloop**.

```
1  if (x > 100) {  
2      a = 3;  
3  } else {  
4      a = 4;  
5      if (x < 20) {  
6          b += 2;  
7          c = 3;  
8      }  
9  }
```

```
1  if (x > 100){a=3;}  
2  else {a=4;if(x<20) {  
3      b+=2;c=3;}}
```

Bovenstaande zijn equivalent.



# JAVA VERSUS PYTHON

## INDENTATIE

Indentatie is belangrijk...maar **niet voor het programmaverloop**.

```
1  if (x > 100) {  
2      a = 3;  
3  } else {  
4      a = 4;  
5      if (x < 20) {  
6          b += 2;  
7          c = 3;  
8      }  
9  }
```

```
1  if (x > 100){a=3;}  
2  else {a=4;if(x<20) {  
3      b+=2;c=3;}}
```

Bovenstaande zijn equivalent.

Witruimte is inbegrepen in de prijs.

## LUSSEN IN JAVA

For loop:

```
1  for ([initialisatie]; [conditie]; [stap]) { ... }  
1  for (int i = 0; i < 10; i++) {  
2      System.out.println(2*i);  
3  }
```

While loop:

```
1  while ([conditie]) { ... }  
1  int n = 20;  
2  while (n > 0) {  
3      System.out.println(i);  
4      n -= 2;  
5  }
```

## JAVA IS STATISCH GETYPEERD

Python achterhaalt **zelf** de types van variabelen. In Java moet **jij** expliciet aangeven of iets een `String` of `int` is.

```
1  int power(int n, int k) {
2      int result = 1;
3      for (int i = 0; i < k; ++i) {
4          result *= n
5      }
6      return result;
7  }
8
9  void showPower(int i, int k) {
10     int p = power(i, k);
11     System.out.println("De " + k + "de macht van " + i + " is " + p);
12     // nothing is returned!
13 }
```

## STRINGS IN JAVA

Zoals in Python heeft een `String`<sup>2</sup> allerlei nuttige *methodes*.

```
1 String s = ""; // ALTIJD "" voor `String's
2 s.length(); // lengte van de string, zoals len(s)
3 s.charAt(n); // nde karakter van s
4
5 System.out.println("x\nx\nx"); // drie lijnen met x
```

## STRINGS IN JAVA

Zoals in Python heeft een `String`<sup>2</sup> allerlei nuttige *methodes*.

```
1 String s = ""; // ALTIJD "" voor `String's
2 s.length(); // lengte van de string, zoals len(s)
3 s.charAt(n); // nde karakter van s
4
5 System.out.println("x\nx\nx"); //
```

---

<sup>2</sup>Met een *hoofdletter*

## STRINGS IN JAVA

Zoals in Python heeft een `String`<sup>2</sup> allerlei nuttige *methodes*.

```
1 String s = ""; // ALTIJD "" voor `String's
2 s.length(); // lengte van de string, zoals len(s)
3 s.charAt(n); // nde karakter van s
4
5 System.out.println("x\nx\nx"); // drie lijnen met x
```

---

<sup>2</sup>Met een *hoofdletter*

# JAVA VERSUS PYTHON

## IDE

IDE: Integrated Development Environment.

Bij Python gebruikten we **Spyder**

- ▶ simpel
- ▶ gemakkelijk
- ▶ eenvoudig

# JAVA VERSUS PYTHON

## IDE

IDE: Integrated Development Environment.

Bij Python gebruiken we **Spyder**

- ▶ simpel
- ▶ gemakkelijk
- ▶ eenvoudig

Voor Java gebruiken we **Eclipse**

- ▶ geschikt voor **enorme projecten**
- ▶ vereist wat **gewenning**
- ▶ enorm krachtig, maar ook erg **veel opties**
- ▶ ook geschikt voor Python (d.m.v. **PyDev**)



# JAVA VERSUS PYTHON

## BESTANDSSTRUCTUUR

Python: bestanden en eventueel “modules”.

Java: project > packages > bestanden met klasse(s) > methoden.

Python voert je bestand **lineair** (van boven naar onder) uit.

Java werkt in twee stappen:

1. **compile** code
2. start **een klasse** in diens **public static void main**(String[] args) {...}.

# JAVA VERSUS PYTHON

## SYNTAX

```
1 public static void main(String[] args) {
2     int x;
3     x = 3;
4     int y = 5;
5     boolean z = true;
6
7     if (x > 100) {
8         System.out.println("x is groter dan 100");
9     } else {
10        System.out.println("x is gelijk aan " + x);
11    }
12 }
```