

Inleiding: Java Programming

Table of contents

- Inleiding
- Java Lifecycle
- Java Development Kit
- Hello World
- Java Programmeertaal
- OOP: intro

Inleiding

Java Programming

Inleiding

1995: SUN → 2010: Oracle

Object-Oriented Language

Java 8 → Functional features

Soorten programmeertalen

Structured Programming (C ...):

- Functies
- Variabelen
- Begin en eindpunt

Object-oriented programming (Java, C# ...):

- Encapsulation – Inheritance – Polymorphism
- Herbruikbaarheid
- Onderhoudbaar
- Aanpasbaar

Soorten Programmeertalen

Gecompileerde programmeertalen(C++ ...):

- Produceren van een uitvoerbaar bestand
- Binaire code inladen en uitvoeren (processor)

Geïnterpreteerde programmeertalen (Javascript ...):

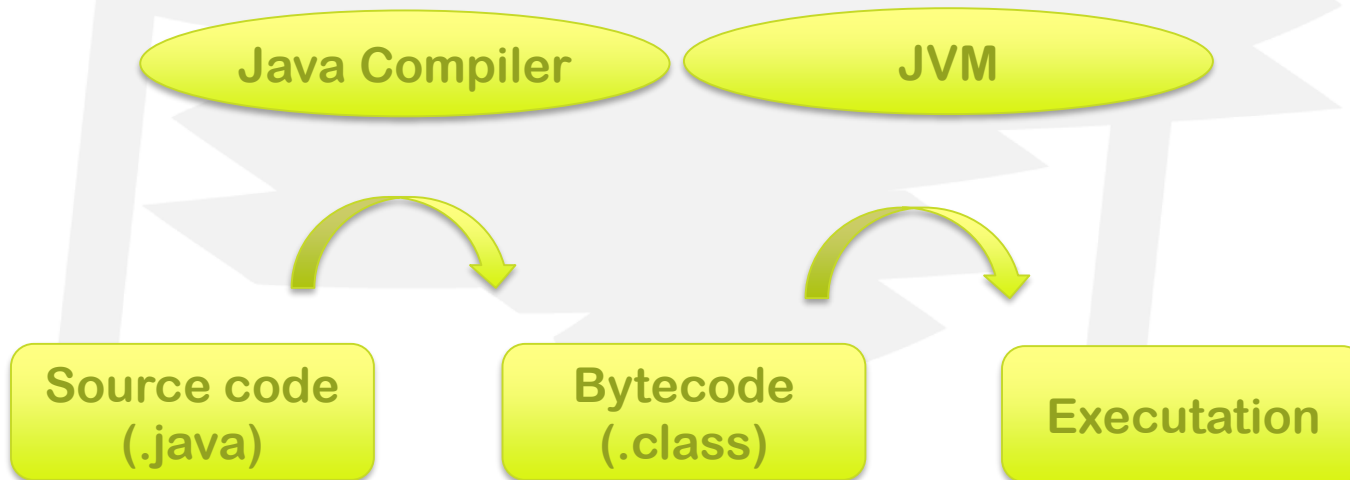
- Vertalen van code tijdens uitvoeren
- Browser = Interpreter

Java Lifecycle

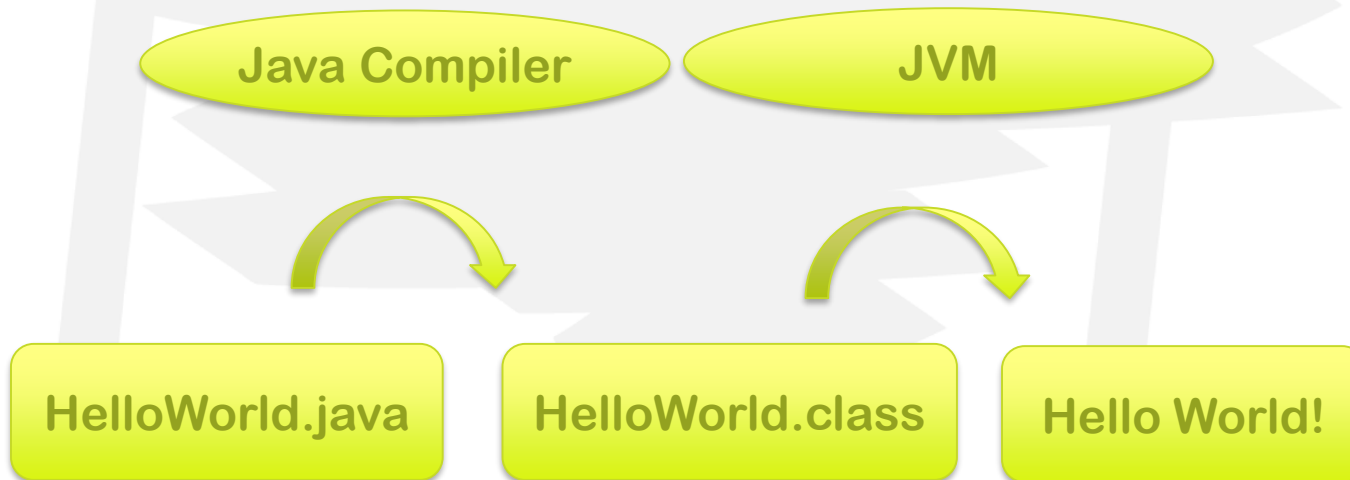
Java Programming

Java Lifecycle

Java: Combinatie van beide



Development Cycle



Java Platforms

JAVA SE: Standard Edition (Desktop apps)

JAVA EE: Enterprise Edition (Web apps)

JAVA ME: Micro Edition (subset of SE)



JDK

Java Programming

Java Development Kit

JDK 8:

- JRE: Java Runtime Environment (bevat JVM en Java API)
- API documentatie
- Compiler, debugger ...

IDE

- Eclipse
- IntelliJ
- Netbeans
- Notepad++
- ...

Hello World

Java Programming

HelloWorld

Oefening 1: HelloWorld app in java

- Notepad ++
- Command Prompt

Java Programmeertaal

Java Programming

Java Programmeertaal

Variabelen en literals

Operatoren

Expressions, statements en blokken

Conditional en flow-control statements

Variabelen en literals

- Type: int, double, String ...

→ **Strongly typed**

- Naam: duidelijke naam
- Scope: elke variabele heeft een bepaalde scope.

Variabelen declareren

Voorbeelden:

```
int getal;
```

```
int getal1, getal2, getal3;
```

```
int getal = 10;
```

```
int getal1 = 10, getal2 = 20, getal3 = 30;
```

```
int getal;  
getal = 10;
```

```
int getal1 = getal;
```

Variabelen declareren

PRIMITIVE DATA TYPE in JAVA

Type	Contains	Default	Size	Range
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
float	IEEE 754 floating point	0.0f	32 bits	$\pm 1.4\text{E-}45$ to $\pm 3.4028235\text{E}+38$
long	Signed integer	0L	64 bits	-9223372036854775808 to 9223372036854775807
double	IEEE 754 floating point	0.0d	64 bits	$\pm 4.9\text{E-}324$ to $\pm 1.7976931348623157\text{E}+308$
boolean	true or false	FALSE	1 bit	NA
char	Unicode character	'\u0000'	16 bits	\u0000 to \uFFFF

Literals

= Letterlijke waarden

boolean: true / false

character: 'a', '\u0061', 97

int: 1, 2 ...

long: 1L, 2L ...

float: 1.0F, 2.0F, 1F ...

double: 1.0, 2.0, 1

Literals

Escape characters te gebruiken in print of String.

Escape Sequence	Character
\n	newline
\t	tab
\b	backspace
\f	form feed
\r	return
\"	" (double quote)
\'	' (single quote)
\\	\ (back slash)

Oefening 2

Gebruik elk primitief datatype in je programma en print dit uit op het scherm.

Oefening 3

Print het volgende 'mopje' uit in java, maak gebruik van escape characters.

My Java "Joke":

After I drink coffee, I like to show the empty mug to the IT guy to tell him that I've successfully installed Java.

He hates me!

Constanten

```
final double PI = 3,1415;
```

```
final int MIJN_GETAL= 10;
```

Casting

`double d = 10; (=implicit casting)`

`float f = (float) 10.0 (=explicit casting)`

`int a = 10;`

`byte b = (byte) a;`

Operatoren

Soorten:

Rekenkundige

Relationele

Logische

Shift/ Bit

Toekenningsoperatoren

Rekenkundige Operatoren

Operator	Gebruik	Resultaat
+	<i>op1 + op2</i>	<i>De som van op1 en op2</i>
-	<i>op1 – op2</i>	<i>Het verschil van op2 en op1</i>
*	<i>op1 * op2</i>	<i>Het product van op1 en op2</i>
/	<i>op1 / op2</i>	<i>Het quotient van op1 gedeeld door op2</i>
%	<i>op1 % op2</i>	<i>Op1 modulo op2 (restwaarde)</i>
++	<i>++op</i>	
++	<i>op++</i>	
--	<i>--op</i>	
--	<i>op--</i>	
-	<i>-op</i>	<i>De negatieve waarde van op</i>
+	<i>+op</i>	<i>Zet een byte,short of char om naar een integer</i>

Opdracht 4 en 5

Maak beide opdrachten in de cursus.

Extra:

```
public class OperatorPlusPlus{  
    public static void main(String[] args){  
        int v= 0;  
        v++;  
        int amount= v++;  
        System.out.println(++v + " " + amount);  
        System.out.println(v);  
    }  
}
```

Relationele Operatoren

OPERATOR	GEBRUIK	RESULTAAT
>	<i>op1 > op2</i>	<i>True als op1 groter is dan op2</i>
>=	<i>op1 >= op2</i>	<i>True als op1 groter of gelijk is dan op2</i>
<	<i>op1 < op2</i>	<i>True als op1 kleiner is dan op2</i>
<=	<i>op1 <= op2</i>	<i>True als op1 kleiner of gelijk is dan op2</i>
==	<i>op1 == op2</i>	<i>True als op1 gelijk is aan op2 (DUBBELE =)</i>
!=	<i>op1 != op2</i>	<i>True als op1 verschillend is aan op2</i>

Logische Operatoren

OPERATOR	GEBRUIK	RESULTAAT
&&	<i>op1 && op2</i>	<i>True als op1 EN op2 waar zijn</i>
 	<i>op1 op2</i>	<i>True als op1 OF op2 waar is</i>
!	<i>!op</i>	<i>True als op NIET WAAR is (inversie)</i>

Toekenningsoperator

=

Gecombineerde toekenningsoperator:

`+=, -=, %=, /=, *= ...`

```
int a += 10;
```

```
int a = a + 10;
```


Conditionele operator

? : verkorte if-else

Bv:

```
int a = 10>9? 1 : 2;
```

```
System.out.println(10>9?“Groter”：“Kleiner”);
```

Oefening 4

Een cursist heeft moeten deelnemen aan 4 java examens. Vraag met behulp van de Scanner zijn resultaten (op 20).

Maak een programma dat het gemiddelde berekent (als decimaal getal) en het behaalde percentage toont.

Conditional statements

- if – else
- Switch-case

De if-instructie

Syntax

```
if (conditie) { ...  
}  
else { ...  
}
```

Voorbeeld

```
if (getal == 23) {  
    System.out.println("getal is gelijk aan 23");  
}  
else getal = 23;
```

De switch

```
switch(integer) {  
    case literal1: statements;  
                    break;  
    case literal2: statements;  
                    break;  
    ...  
    default: statements;  
}
```

De switch

Voorbeeld

```
switch(month) {  
    case 1: System.out.println("januari");  
            break;  
    case 2: System.out.println("februari");  
            break;  
    ...  
    default:  
        System.out.println("Onbekende maand");  
}
```

Flow Control statements

- While
- Do – While
- For

De while-lus

Syntax

```
while (condition) statement;  
of  
while (condition) {  
    statement;  
    statement;  
    ...  
}
```

Voorbeeld

```
While (getal < 8)  
    System.out.println(getal);
```


De do..while lus

Syntax

```
do  
    statement;  
while (condition);
```

```
do {  
    statement;  
    statement;  
    ...  
} while (condition);
```

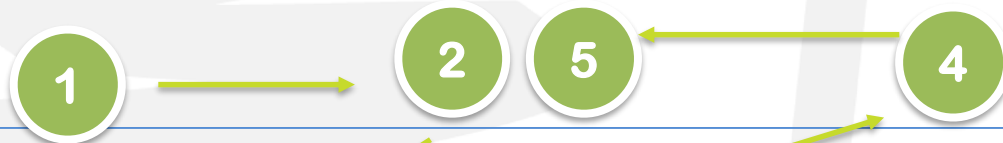
De do..while lus

Voorbeeld

```
int number = 1;  
do  
    System.out.println(number++);  
while (number < 10);
```

De for-lus

Syntax



```
for(initialisation; condition; increment ) {  
    statement;  
    statement;  
    ...  
}  
...
```

Opdracht 15 tem 18

1^{ste} deel van elke opdracht samen

Object Oriented Programming

Java Programming

Hoofdtype (base class)

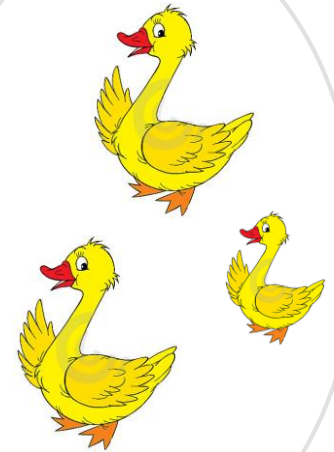
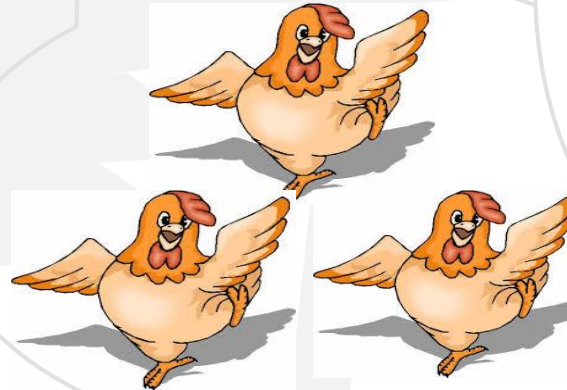
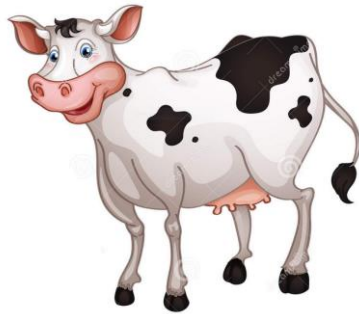
Dieren

klasse of type
(class)

Vogel

Kip

Gans



objecten



Voorbeeld van een class (klasse):

Kip

- snavel
- pluimen
- poten

- verplaatsen (rennen, vliegen)
- communiceren (kakelen)



Class = object type



Properties =
eigenschappen



Methods = gedragingen,
methodes

Objecten van de class kip:



Louise



Sofie



Claire

Objecten noemt men ook wel eens
instanties van een bepaalde class

Access Modifiers

Niveau	Eigen klasse	Package	Subklassen	Alle klassen
private	X			
package	X	X		
protected	X	X	X	
Public	X	X	X	X

Peilers van OOP

Pijlers van OOP:

- Inkapseling (**encapsulation**)
- Overerving (**inheritance**)
- Meervormigheid (**polymorfism**)

Encapsulation

= variabelen afschermen vd buitenwereld.

Via getters en setters zelf beslissen wat aanpasbaar is.

Inheritance

= Overerving

Gemeenschappelijke code samenplaatsen.

Polymorphism

= één object kan meerdere vormen aannemen.

Bv: `Auto mijnAuto = new BMW();`
`mijnAuto = new Honda();`

Boodschappen

Boodschappen

Object Kip

Object Mens



Eet (



)



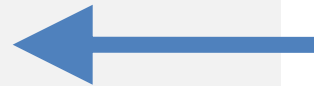
Boodschappen

Object Kip

Object Mens



Bestemming



eet (



)



Boodschap



Resultaat (= optioneel)

Voorbeeld

```
Kip claire = new Kip();  
Ei mijnEi = claire.eet("mais");
```

Instance variables en methods **vs** **Class** variables en methods

Class

Definitie: Een klasse is een blauwdruk die de eigenschappen en methoden van objecten van dezelfde soort bepaalt.

De variabelen en methoden van een klasse noemt men ook wel **klasse** variabelen en klasse methoden.

De variabelen en methoden van een object (of instantie) noemt men ook wel **instance** variabelen en methoden

Voorbeelden van **class** en **instance** variabelen

Eigenschappen

- A. 4 wielen
- B. stuur
- C. kleur

Class Auto



ABC-123



GPS-987



TYD-555



LOL-007

Auto objecten

Hebben allemaal 4 wielen en een stuur = klasse variabelen.

Hebben allemaal een kleur, maar deze verschilt = instance variabele

Opdracht 1

<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

Opdracht 1 in cursus maken met behulp van de Java documentatie.