

Full Stack Developer Bootcamp

Basis Programmeren

Les 3

Basis Programmeren

Les 3



```
public class Lesson3 {  
    public static void main (String[] args) {  
        welcomeMessage();  
    }  
    public static void welcomeMessage() {  
        int les = 3;  
        for(int i = 0; i <= les; i++) {  
            if(i == les) {  
                System.out.println("Welkom bij de "+i+"e les van Basis Programmeren");  
            }  
        }  
    }  
}
```

Lesplanning

- Terugblik les 1+2
- Verzamelingen van informatie
- Methoden
- Java demonstratie
- *Opdracht: Euro/dollar converter - deel 3*

Terugblik les 1+2

Wat weten jullie nog?

Flinga

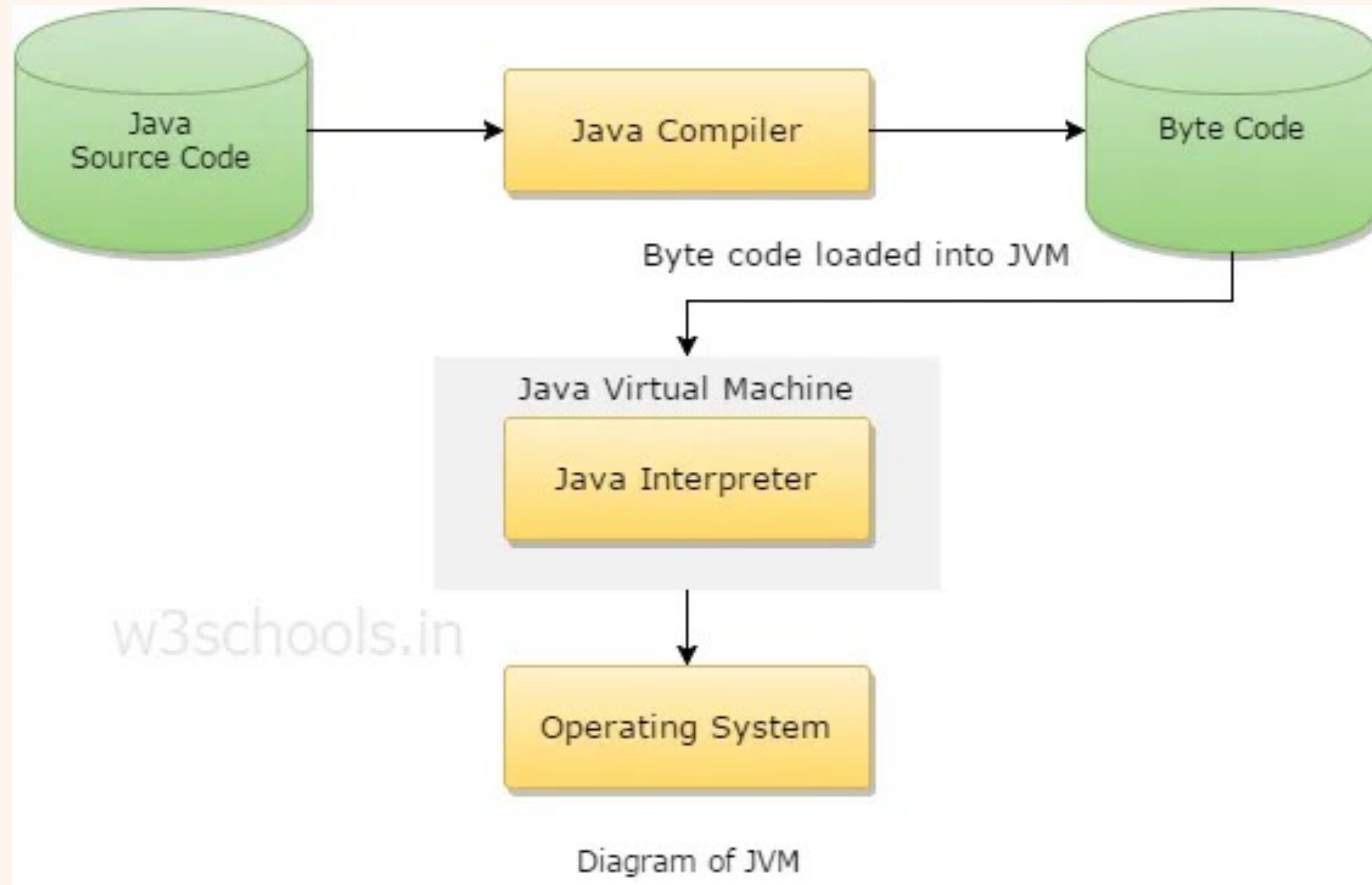
- Ga naar flinga.fi/s/FLGJHZD
- Bekijk de 4 cirkels
- Voeg per cirkel **woorden of code** toe wat gerelateerd is aan het onderwerp
- Bijvoorbeeld "Java Virtual Machine" bij de cirkel met "Java"

Java

- Java 1.0 in 1995 uitgebracht door SUN
 - Hedendaags Java 18 (uitgebracht in maart 2022)
 - Er wordt veel gewerkt met Java versie 11 of 16 vanwege lange support
- Object georiënteerde programmeertaal
- Java is platform onafhankelijk, het draait op Windows, Apple, Linux, Android, etc.
- Java-code moet gecompileerd worden naar byte-code
- Gecompileerde byte-code wordt door de Java Virtual Machine ingeladen
- Voor de cursus Basis Programmeren zal Java versie 16 of versie 17 beide geschikt zijn.

Java Virtual Machine

JVM



→ Bron: <https://www.w3schools.in/java/java-virtual-machine>

Java

Theorie

- .java-bestanden voor Java broncode
- .class-bestanden voor gecompileerde Java-code (= bytecode)

- Diverse soorten applicaties mogelijk met Java:

 - Console (command line interface) applicaties

 - Grafische desktop applicaties

 - Web applicaties

 - Android applicaties

- Afkortingen:

 - IDE = Integrated Development Environment, JVM = Java Virtual Machine

 - JDK = Java Development Kit (compileren), JRE = Java Runtime Environment (uitvoeren)

Informatie opslaan

Variabelen

→ Informatie wordt opgeslagen in variabelen

Denk aan wiskunde, bijv stelling van Pythagoras $a^2 + b^2 = c^2$

→ Variabelen hebben bij Java een **type**, een **naam** en een **waarde**

→ type geeft aan wat voor informatie in die variabele wordt opgeslagen

→ naam beschrijft de informatie die opgeslagen wordt

→ waarde is de opgeslagen informatie zelf

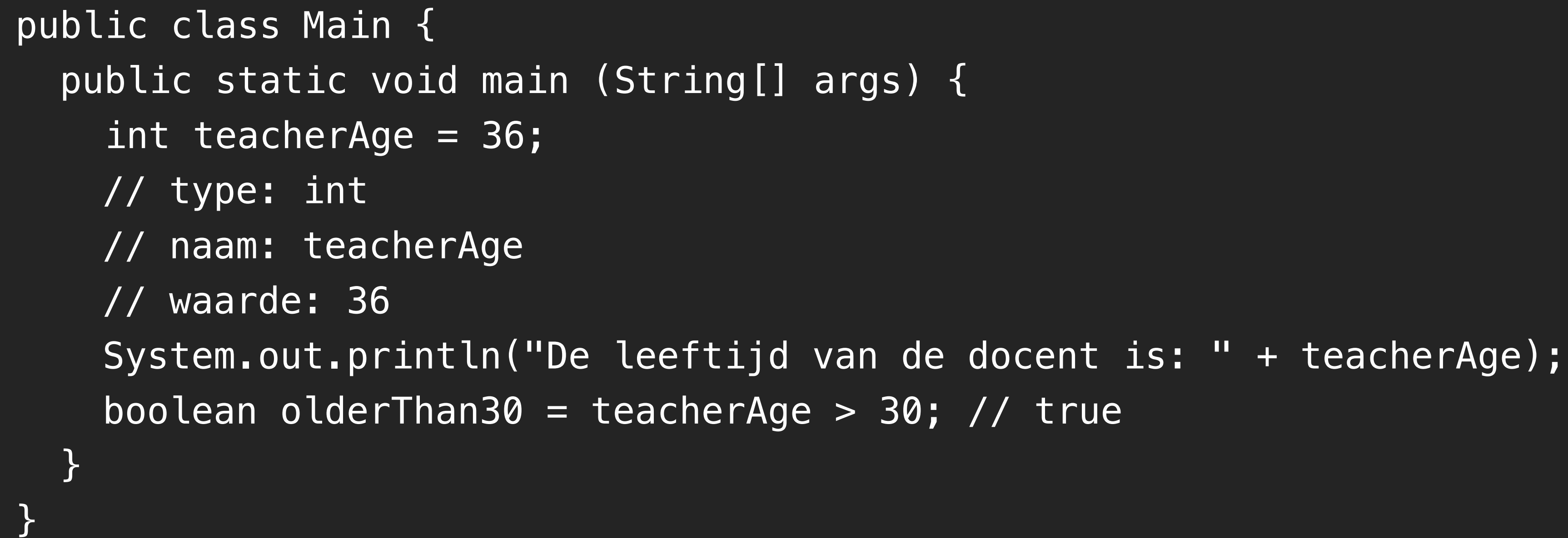
Primitieve typen

Variabelen

- Gehele getallen: byte, short, **int**, long
- Reëel getallen: float, **double**
- 1 karakter: char
- Logisch: boolean (true / false)

Informatie printen

Voorbeeld-code



```
public class Main {  
    public static void main (String[] args) {  
        int teacherAge = 36;  
        // type: int  
        // naam: teacherAge  
        // waarde: 36  
        System.out.println("De leeftijd van de docent is: " + teacherAge);  
        boolean olderThan30 = teacherAge > 30; // true  
    }  
}
```

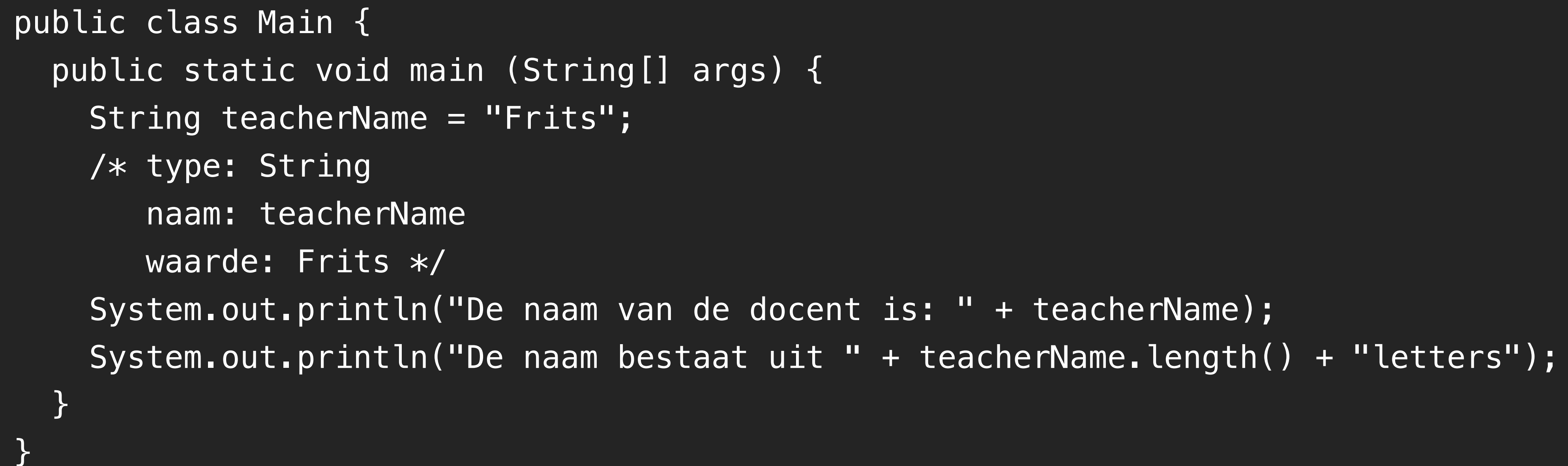
Tekst opslaan

String-variabelen

- Tekst wordt opgeslagen in een variabele van het type **String**
- String is geen primitieve datatype
- Bij een String-variabele worden objecten gebruikt van het type String
- Daardoor is het mogelijk om methoden uit te voeren, bijvoorbeeld om de lengte van de tekst op te vragen (*length*) of om de tekst om te zetten naar hoofdletters (*toUpperCase*)
- Zie String-methoden op: https://www.w3schools.com/java/java_ref_string.asp

Tekst opslaan en printen

Voorbeeld-code



```
public class Main {  
    public static void main (String[] args) {  
        String teacherName = "Frits";  
        /* type: String  
           naam: teacherName  
           waarde: Frits */  
        System.out.println("De naam van de docent is: " + teacherName);  
        System.out.println("De naam bestaat uit " + teacherName.length() + "letters");  
    }  
}
```

Keuzeconstructies

If/else statement vs switch statement

- Zie code op: <https://replit.com/@NOVI/Huisnummers>
- Klik op 'Show Files' om de code van Main.java te bekijken
- Druk op de groene Run-knop om de code uit te voeren
- Switch statement: <https://replit.com/@NOVI/Switch-statement>

Herhalingsconstructies

For, while, do-while

for
loop

The Java for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.

while
loop

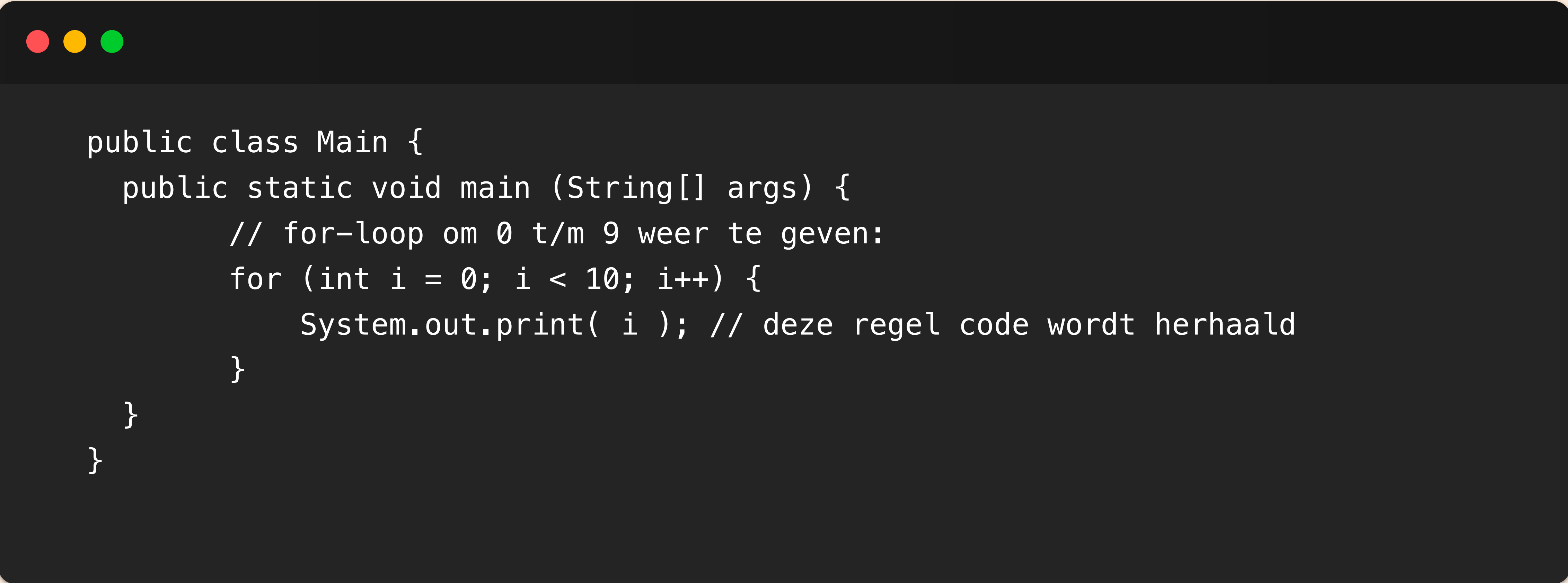
The Java while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

do-while
loop

The Java do-while loop is used to iterate a part of the program several times. Use it if the number of iteration is not fixed and you must have to execute the loop at least once.

For-loop

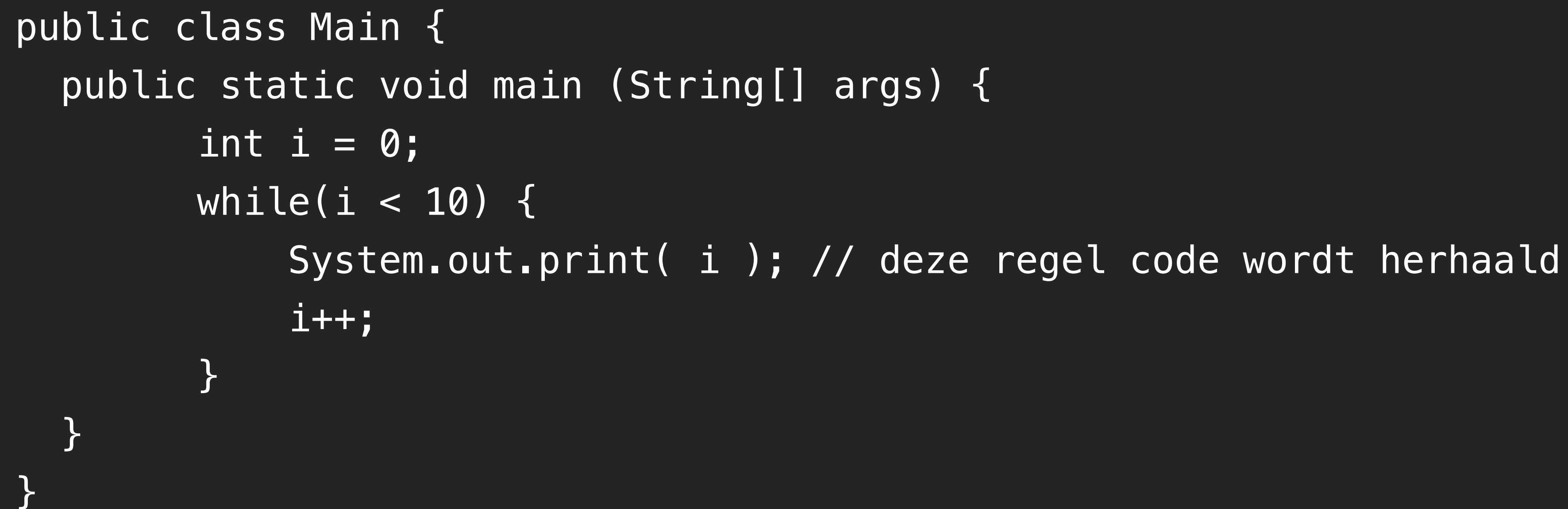
Voorbeeld-code



```
public class Main {  
    public static void main (String[] args) {  
        // for-loop om 0 t/m 9 weer te geven:  
        for (int i = 0; i < 10; i++) {  
            System.out.print( i ); // deze regel code wordt herhaald  
        }  
    }  
}
```


While-loop

Voorbeeld-code



```
public class Main {  
    public static void main (String[] args) {  
        int i = 0;  
        while(i < 10) {  
            System.out.print( i ); // deze regel code wordt herhaald  
            i++;  
        }  
    }  
}
```

Gelukt met Opdracht??

Euro/Dollar converter

- Ga verder met de Java-code van de opdracht van gisteren.
- Pas een keuzeconstructie (if/else-statement) en een herhalingsconstructie (bijv. while) toe zodat de invoer van de gebruiker wordt gecontroleerd (het programma vraagt net zolang om invoer totdat een goed getal is ingevoerd).
- Pas een herhalingsconstructie toe zodat de gebruiker vaker dan 1 keer een bedrag kan invoeren.
- *Bonus*: Zorg dat de gebruiker bij het begin kan kiezen tussen *euro-naar-dollar* of *dollar-naar-euro*.
- *Bonus 2*: Voeg meer valuta's toe: pond, yen, bitcoin, etc.

Verzamelingen

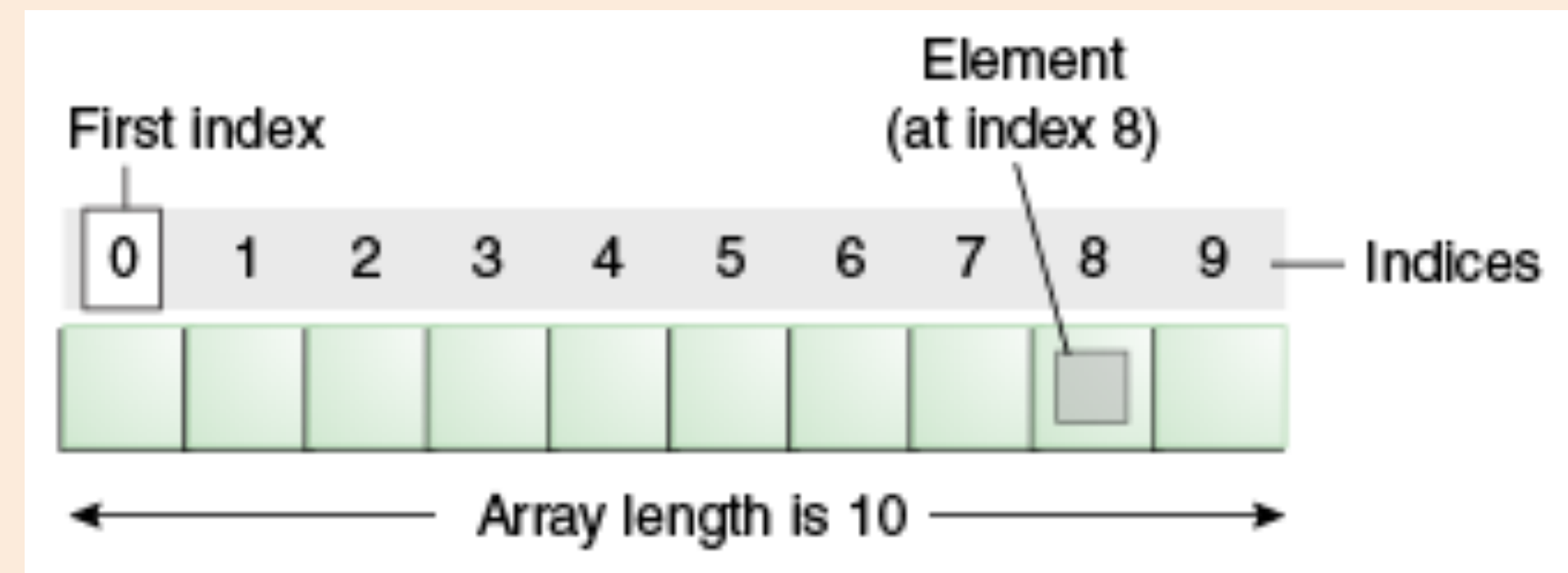
Arrays



Verzamelingen

Theorie

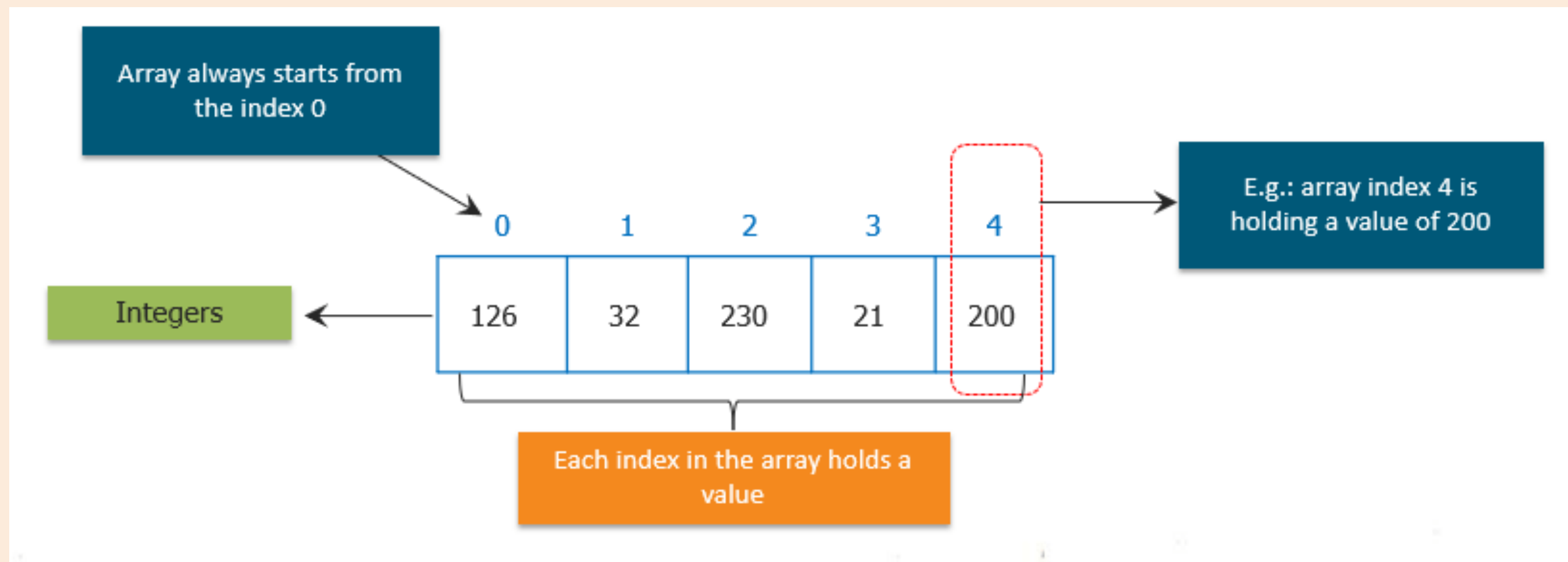
- Een array is een variabele die een **rij waarden** kan bevatten
- Zie plaatje. Elk vak kan een waarde krijgen (bijv een getal, of een stuk tekst).
- Het 1e vak bij een array is altijd 0



int Array


Voorbeeld van een verzameling van getallen

Bron: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>



String Array

Voorbeeld-code

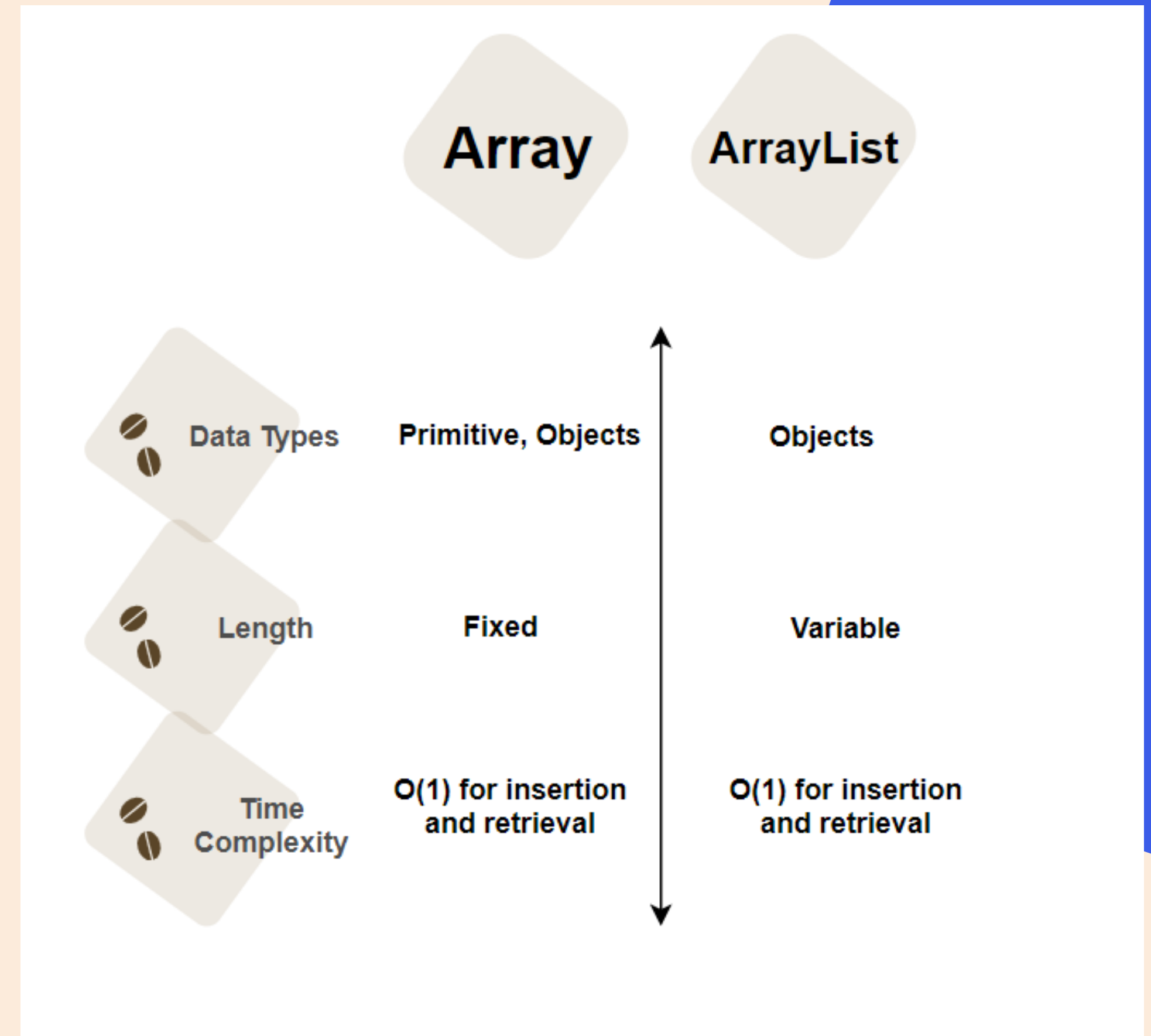


```
public class Main {  
    public static void main (String[] args) {  
        String[] artists = new String[4];  
        artists[0] = "Diggy Dex";  
        artists[1] = "Kraantje Pappie";  
        artists[2] = "Thomas Acda";  
        artists[3] = "Heideroosjes";  
  
        System.out.println("Artiesten in verzameling:");  
        for(int i=0; i < artists.length; index++) { // via artists.length kan je de lengte opvragen  
            System.out.println(artists.length[i]);  
        }  
    }  
}
```

Verzamelingen

Array vs ArrayList

- Arrays gebruik je als je vooraf de lengte weet van de verzameling.
- Arrays werken met primitieve data typen (int, double, char, boolean) en objecten.
- Bij ArrayLists is de lengte variabel.
- ArrayLists werken alleen met objecten.
Er zijn wel *wrapper-classes* beschikbaar (bijv Integer-klasse ipv int).



Verzamelingen

Theorie

→ Voordeel van gegevens in een array plaatsen:

→ Arrays sorteren:

<https://www.geeksforgeeks.org/arrays-sort-in-java-with-examples/>

→ Arrays vergelijken:

<https://www.geeksforgeeks.org/compare-two-arrays-java/>

Methoden

```
public static void doSomething() {}
```


Methoden

Theorie

- Methoden worden gebruikt om bepaalde acties te groeperen.
- Een methode wordt uitgevoerd als deze wordt aangeroepen.
 - De main-methode wordt automatisch aangeroepen bij het starten van de applicatie.
- Het is mogelijk om gegevens mee te geven aan de methoden: *parameters* (input).
- Het is mogelijk om een resultaat terug te geven na het uitvoeren van de methoden: *return-waarde* (output).

Waarom methoden gebruiken?

Theorie

- Opdeling van code zorgt voor meer overzicht
 - Het voorkomt onnodig dubbele code te typen
 - Daardoor ook tijdbesparend
- 

Methoden

main()-methode

→ Voorbeeld: <https://repl.it/@NOVI/Hallo-Novl>

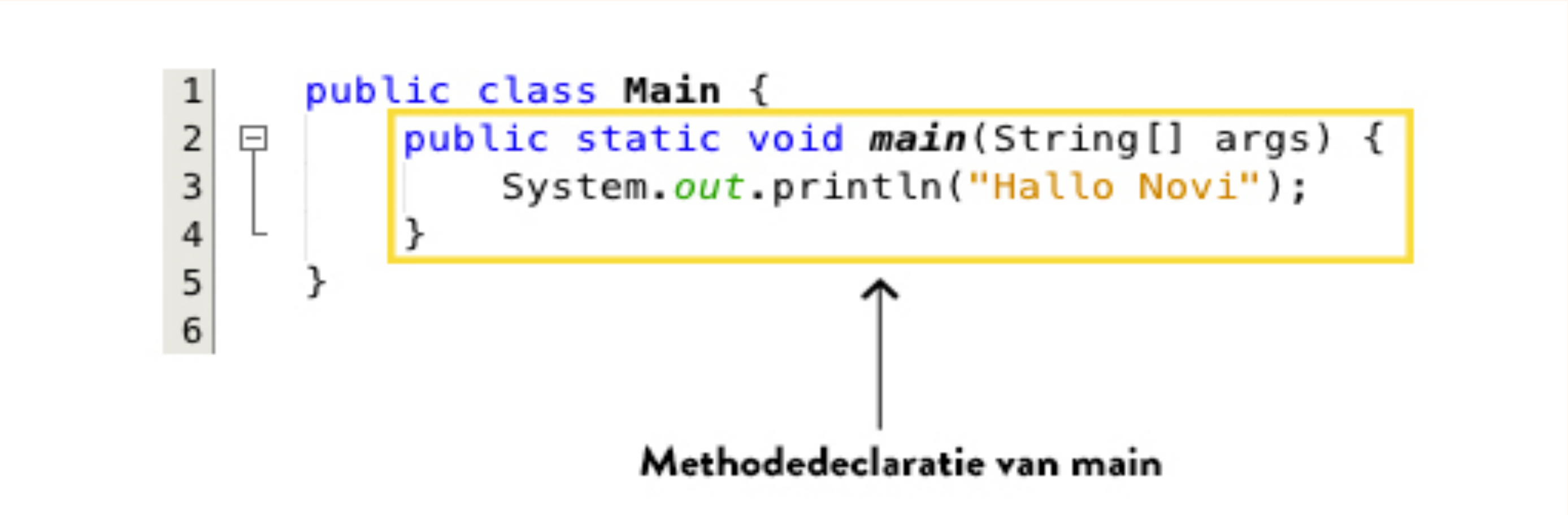
→ **Modifier**: public static

→ **Return type**: void
(void = geen return type)

→ **Methode naam**: main

→ **Parameters**: String[] args

→ **Methode Body**: System.out.println();



```
1 public class Main {  
2     public static void main(String[] args) {  
3         System.out.println("Hallo Novi");  
4     }  
5 }  
6
```

Methodedeclaratie van main

Methoden

Voorbeeld-code (<https://www.geeksforgeeks.org/methods-in-java/>)



```
public class Main {  
    public static void main (String[] args) {  
        System.out.println("Grootste getal: " + max(10, 3) );  
    }  
    public static int max(int x, int y) {  
        if(x > y) {  
            return x;  
        }  
        return y;  
    }  
}
```

// **modifier:** public (static)
// **return type:** int
// **methodenaam:** max
// **parameters:** int x, int y
// **return waarde:** x of y

Java demonstratie

- Verzamelingen van informatie (arrays)
- Methoden
- Stel gerust vragen via chat

Opdracht

Euro/Dollar converter

60 min

- Ga verder met de Java-code van de opdracht van vorige lessen.
- Verdeel code over diverse methoden. Bijvoorbeeld een methode om de **berekening** uit te voeren waarbij 2 variabelen (koers, euro-bedrag) meegeeft en het dollar-bedrag als waarde teruggeeft.
- Lukt het je om gebruikersinvoer (bijvoorbeeld de euro-bedragen) toe te voegen aan een double-array?
- *Bonus:* Creëer een nieuwe klasse met 1 of meerdere methoden. Probeer de methoden vanuit de main-methode aan te roepen.

Bijvoorbeeld klasse "**CurrenyConvertor**" met een methode **doMath**(double rate, double amount).

Quiz

Vragen over Java-code

→ Arrays:

https://www.w3schools.com/java/exercise.asp?filename=exercise_arrays1

→ Methoden:

https://www.w3schools.com/java/exercise.asp?filename=exercise_methods1

A large, solid yellow shape in the bottom right corner of the slide, resembling a stylized wave or a large comma.

Huiswerk

Edhub & opdrachten

- Neem **alle hoofdstukken** door van de cursus Programming Basics
- Rond de opdracht af die tijdens de les is besproken