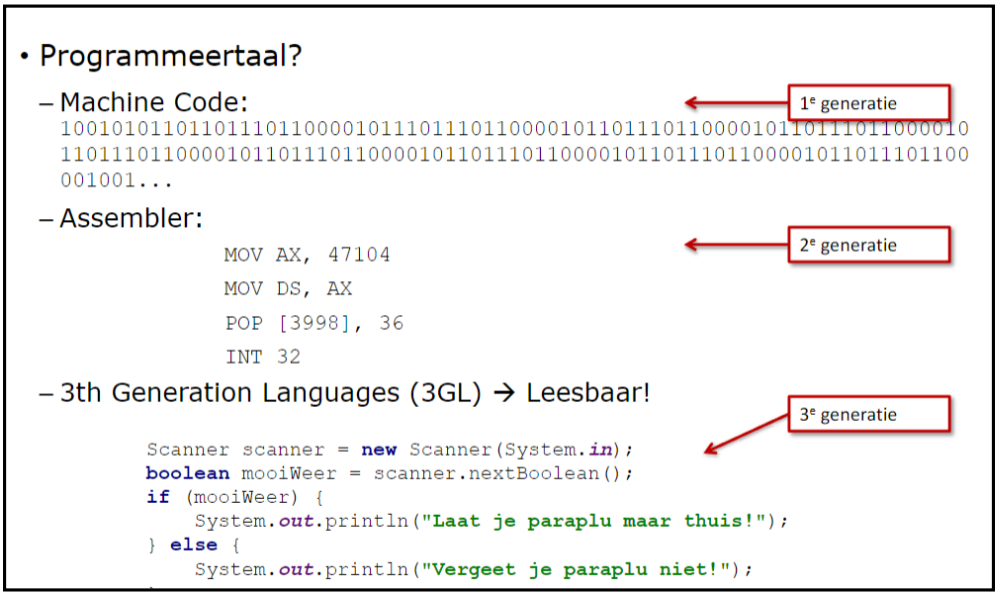
Object-oriented programming

Keanu Van Winckel - 104A

P1W1 - Inleiding - Algoritmes

Programmeertalen zijn geëvolueerd door drie generaties:

- Eerste generatie talen, zoals machinecode, zijn direct begrepen door computers.
- Tweede generatie talen, zoals 'Assembly', bieden symbolische representatie van machinecode.
- Derde generatie taal, zoals 'Java', is nog een hoger abstractieniveau. Java maakt gebruik van leesbare syntax en biedt platformonafhankelijkheid door het Java Virtual Machine (JVM) -concept.Hierdoor hoeven programmeurs zich minder bezig te houden met hardware-specifieke details.

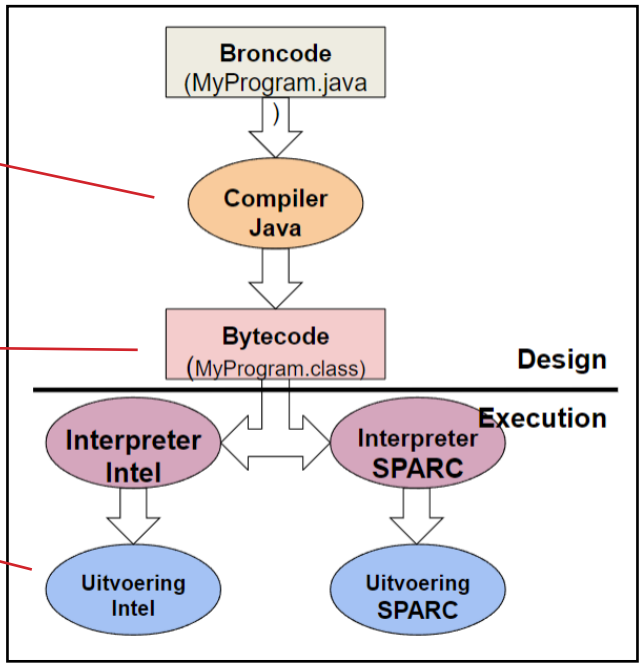


Van '.java' naar execution

Een Java-programma wordt eerst gecompileerd naar platformonafhankelijke bytecode met de javac-compiler.

Deze bytecode wordt vervolgens uitgevoerd door de Java Virtual Machine (JVM), die het op het doelsysteem vertaalt naar machinecode via Just-In-Time (JIT) compilatie.

Hierdoor kan Java op verschillende systemen draaien, waardoor het de "write once, run anywhere" (WORA) eigenschap heeft.



JAVA IS DUS: 'PLATFORMONAFHANKELIJK'

Java programma's uitvoeren

Java Runtime Environment (JRE):
De JRE is de runtime-omgeving die nodig is om Java-toepassingen uit te voeren.
Het bevat de Java Virtual Machine (JVM), die de bytecode van een Java-programma uitvoert, evenals de Java-bibliotheekklassen die nodig zijn voor de uitvoering.
Kortom, de JRE biedt alles wat nodig is om Java-applicaties te draaien, maar het bevat geen tools voor ontwikkeling.

Java programma's schrijven

Java Development Kit (JDK):
De JDK is een uitgebreidere kit dan de JRE, omdat het naast de runtime-componenten ook ontwikkelingstools bevat.
Het bevat de Java-compiler (javac) waarmee je Java-broncode kunt omzetten in bytecode.
Daarnaast bevat het hulpprogramma's voor debugging, monitoring en het bouwen van Java-toepassingen.
Kort gezegd, de JDK is nodig als je een Java-programma wilt ontwikkelen, compileren en bouwen.

Eerste Java programma

Commentaar: Commentaar: code tussen /* en */ of achter // wordt genegeerd door de compiler en is dus enkel ter info.

Pakketnaam: om sourcefiles te groeperen (creëert aparte submap)

class: wordt gedefinieerd. Een klasse is een essentieel begrip in objectgeorienteerd programmeren

"main methode": dit is het startpunt van elke java desktop applicatie

```
/* This Java application shows
the text 'Hello World!' on the screen. */

package hello;

public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello World!"); // Show the text
        }
}
```

Algoritme: reeks instructies die uitgevoerd moeten worden om een bepaald doel te bereiken.
→ Als je programmeert, doe je in feite niets anders dan algoritmes bedenken voor bepaalde problemen en ze dan in een programmeertaal aanbieden aan je computer...

Algoritme:
- Opeenvolging stappen
- Volgorde belangrijk
- Typische stappen zijn:
  - *Input vragen*
  - *Output geven*
  - *Iets testen*
  - *Een aantal stappen herhalen*
  - *Iets berekenen*
- Je moet de stappen verder uitsplitsen tot
- basisinstructies die de computer begrijpt

Kort samengevat:

Wat is een 3GL?
Een 3GL (Derde Generatie Taal) is een programmeertaal op een hoog abstractieniveau, zoals Java, C, of Python, die begrijpelijk is voor programmeurs en minder afhankelijk is van de onderliggende hardware.

Wat is compileren?
Compileren is het proces waarbij broncode van een programmeertaal wordt omgezet in machinecode of bytecode, wat uitvoerbaar is door een computer. Het resultaat is een uitvoerbaar bestand of bytecode dat later kan worden uitgevoerd.

Wat is bytecode?
Bytecode is een tussenliggende representatie van broncode die platformonafhankelijk is. Het wordt vaak gebruikt in talen zoals Java. De bytecode wordt vertaald naar machinecode door de Java Virtual Machine (JVM) tijdens de uitvoering.

Wat is de JVM?
De Java Virtual Machine (JVM) is een virtuele processor die Java-bytecode uitvoert. Het vertaalt bytecode naar machinecode op het moment van uitvoering, waardoor Java--programma's op verschillende systemen kunnen draaien zonder opnieuw te worden gecompileerd.

Wat is de JRE?
De Java Runtime Environment (JRE) is een bundel softwarecomponenten die nodig zijn om Java-toepassingen uit te voeren. Het bevat de JVM en de Java-bibliotheekklassen die nodig zijn voor de uitvoering van Java-programma's.

Wat is de JDK?
De Java Development Kit (JDK) is een uitgebreidere kit dan de JRE en bevat naast runtime-componenten ook ontwikkelingstools zoals de Java-compiler (javac) en hulpprogramma's voor het bouwen en debuggen van Java-programma's.

Waarvoor staat API?
API staat voor Application Programming Interface. Het is een set definities en protocollen waarmee softwarecomponenten met elkaar kunnen communiceren. Een API kan functies en procedures bevatten waarmee ontwikkelaars toegang hebben tot de functionaliteit van een bepaalde softwarecomponent.

Wat is een Algoritme?
Een algoritme is een reeks instructies die uitgevoerd moeten worden om een bepaald doel te bereiken.

Wat is commentaar?
Commentaar: code tussen /* en */ of achter // wordt genegeerd door de compiler en is dus enkel ter info.

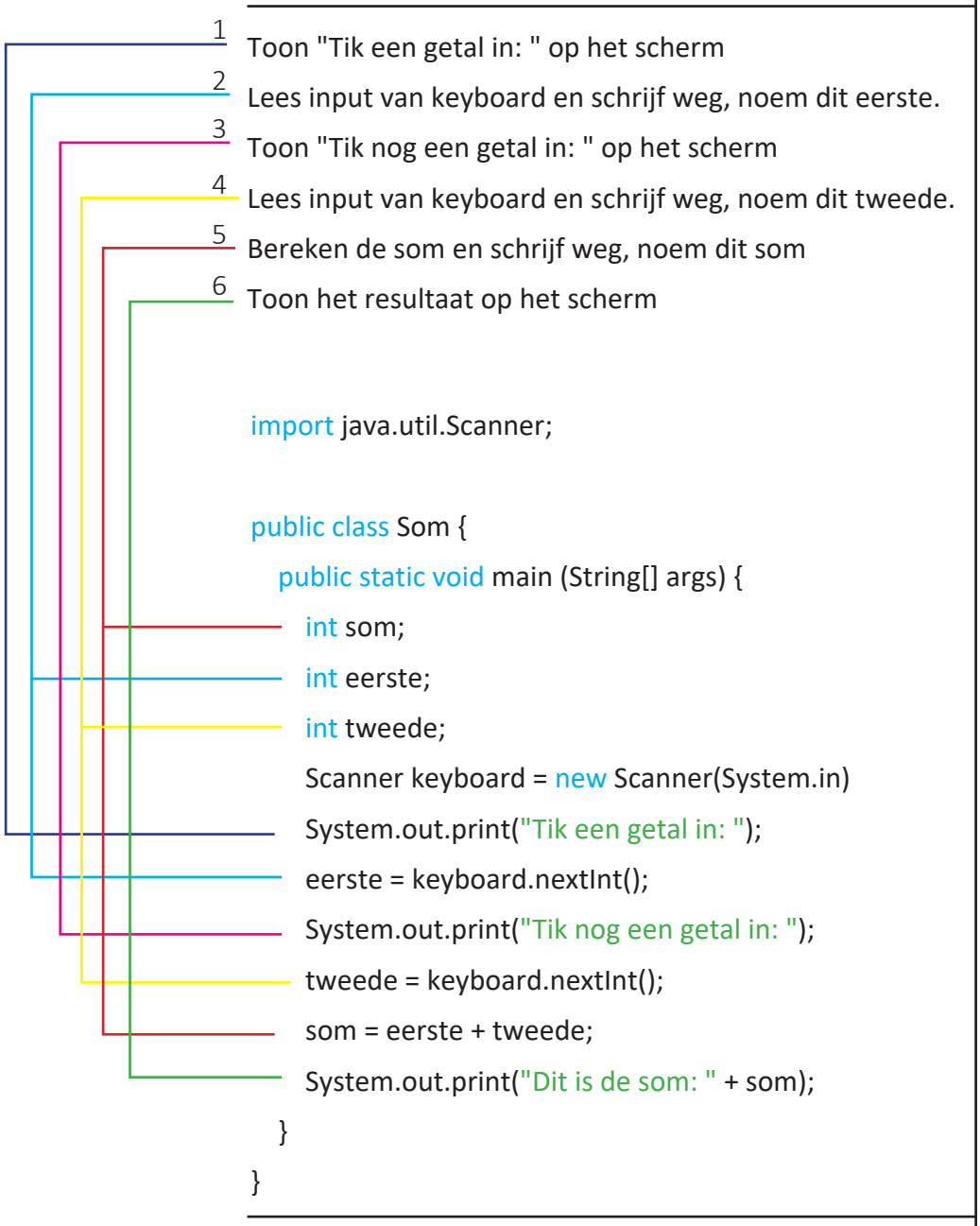
Wat is de Main Method?
De main method is hetstartpunt van elke java desktopapplicatie

Van algoritme naar Java code

NOTICE:
De import statement zorgt ervoor dat je een keyboard object kan maken om input the vragen aan de gebruiker

Uitvoering van de code start bij de main methode

Om een getal op te slagen met je een int initialiseren



Samengevat les 1

H1: Inleiding:
- 3GL, Compileren, Interpreteren, JVM

H2: Java Development Kit

H3: Het eerste Java programma
- HelloWorldApp, javac, java

H4: het algoritme
- Stappenplan
- Variabelen, rekenen, IO, testen, lussen
- Verfijnen tot Java instructies