Les 11 voorbereiding

# Theorie

**Screencast over onderwerp 8: statische en dynamische types en abstract**

<http://www.youtube.com/playlist?list=PLpd9jJvk1PjmbJnRN4kOfpYgKvJ2PCDsP>

## Boek

### Hoofdstuk 8

8.7 t/m 8.11 pagina 319 t/m 328

### Hoofdstuk 9

9.1 t/m 9.4, pagina 332 t/m 341

9.6 t/m 9.12 pagina 343 t/m 356

### Hoofdstuk 10

10.3 en 10.4 , pagina 376 t/m 385 details van de code zijn niet belangrijk, zorg dat je de definities snapt

# Opgave Compile-time vs Runtime

Hieronder is een lijst van begrippen te vinden. Geef per begrip aan of ze gedurende compile-time of gedurende runtime een rol spelen.

* Abstract
* Klasse
* Instantie
* Dynamische type
* Statische type
* Geheugenmodel
* Methode look-up
* Uitvoeren van een programma
* Controleren van een programma

# Opgave Geheugenmodel Knop

Hieronder is een gedeelte van een hoofdprogramma gegeven.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | @SuppressWarnings("serial") public class KnopApp extends PApplet {  ...  private Licht l;  private Knop k;    public void setup() {  l = new Licht(this);  **k = new LichtKnop(this, l, 20, 20, 50, 50);**  ...  }  ... } |

Maak een geheugenmodel van het programma op het moment dat het statement op regel 9 (vetgedrukt) net is uitgevoerd. Je hoeft de velden uit LichtKnop niet weer te geven en voor de klassen Knop, LichtKnop en Licht wordt de code uit de screencasts gebruikt.

# Opgave ABC 2

Gegeven onderstaande klassen:

public class A {  
   
 @Override  
 public String toString() {  
 return "A";  
 }   
}

public class B extends A {  
   
 @Override  
 public String toString() {  
 return "B";  
 }   
}

public class C extends B {  
   
 @Override  
 public String toString() {  
 return "C";  
 }  
}

## Opgave A

In de klasse A staat de annotatie @Override bij de methode toString waarmee gesuggereerd wordt dat deze methode in een superklasse van A bestaat en overschreven wordt. Erft de klasse A inderdaad van een andere klasse die de methode toString definieert, of kan de annotatie beter weggelaten worden?

## Opgave B

Gegeven onderstaande hoofdprogramma:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | public class ABC2 {  public static void main(String[] args) {  A a1 = new A();  A a2 = new B();  A a3 = new C();  B b1 = new A();  B b2 = new B();  B b3 = new C();  C c1 = new A();  C c2 = new B();  C c3 = new C();    System.out.println(a1.toString());  System.out.println(a2.toString());  System.out.println(a3.toString());  System.out.println(b1.toString());  System.out.println(b2.toString());  System.out.println(b3.toString());  System.out.println(c1.toString());  System.out.println(c2.toString());  System.out.println(c3.toString());   } } |

Geef voor elk statement op regel 3 t/m 11 aan of ze door de compiler heenkomen, of dat er een foutmelding optreedt. Welk patroon zie je?

## Opgave C

Geef voor elk statement op regel 13 t/m 21 aan uit welke klasse de methode toString komt die tijdens het uitvoeren van het programma gebruikt wordt. Sla uiteraard de regels over die vanwege eerdere compileerfouten niet kunnen worden uitgevoerd.