Les 5 lesprogramma

# Opgave Geheelgetal

Individueel, begrip

In onderstaand programma is er een klasse gemaakt voor een geheel getal. De klasse heeft één eigenschap met de waarde van het getal. Hoewel het misschien raar is om een klasse te maken voor een ding dat al bestaat, gaat het in deze opgave om het verschil tussen een klasse en primitieve te onderzoeken.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | class GeheelGetal {  int waarde;    GeheelGetal(int startwaarde) {  waarde = startwaarde;  }  }  void setup() {  int getal1 = 10;  GeheelGetal getal2 = new GeheelGetal(10);  println("getal1: " + getal1);  println("getal2: " + getal2.waarde);  println("-----------------------");    telErTienBijOp(getal1);  telErTienBijOp(getal2);    println("getal1: " + getal1);  println("getal2: " + getal2.waarde);  }  void telErTienBijOp(int getal) {  getal += 10;  }  void telErTienBijOp(GeheelGetal getal) {  getal.waarde += 10;  } |

Draai het programma in processing en bekijk de uitvoer: regels 20 en 21 leveren niet hetzelfde getal.

## Onderdeel A

Teken het geheugenmodel op het moment dat telErTienBijOp(getal1) op regel 17 bezig is en hierbinnen bij regel 27 is.

## Onderdeel B

Teken het geheugenmodel op het moment dat telErTienBijOp(getal2) op regel 18 bezig is en hierbinnen bij regel 31 is.

## Onderdeel C

Verklaar op basis van de twee geheugenmodellen het verschil in de uitvoer.

## Onderdeel D

Klassikaal

Op regels 14 en 21 kun je niet println(getal2) typen, maar moet je println(getal2.waarde) gebruiken. Met een kleine toevoeging aan de klasse GeheelGetal is dit wel mogelijk. Welke? Waarom zouden we dit willen?

Tags: Geheugenmodel, overloading, primitieve typen, referentietypen, pass by reference, toString

# Opgave KaartjesAutomaat 2: Meerdere soorten kaartjes

Tweetallen, programmeervaardigheid

## Functionele informatie

De klasse KaartjesAutomaat moet nu kaartjes kunnen bijhouden van verschillende soorten films. De gebruiker van de automaat kan dan één kaartje selecteren uit het aanbod van de automaat. Natuurlijk moet het gekozen kaartje ook weer ontkozen kunnen worden. De overige functionaliteit is hetzelfde.

## Technische informatie

Alle informatie en al het gedrag van een kaartje moet uit de klasse KaartjesAutomaat gehaald worden en in een nieuwe klasse met de naam Kaartje worden gestopt. De klasse KaartjesAutomaat moet een array van Kaartje bijhouden. Daarnaast moet er in een instantievariabele bijgehouden worden welk Kaartje is geselecteerd. Het selecteren van het kaartje gaat doormiddel van de index in de array van kaartsoorten (dus niet gaan zoeken op kaartnaam o.i.d.). De lijst met kaartsoorten moet in de constructor van KaartjesAutomaat meegegeven worden en kan daarna niet worden gewijzigd.

## Opgave A Ontwerpen van de klassen

Teken het klassendiagram met de namen van de klassen, de eigenschappen van de velden (inclusief type) en de methoden (inclusief parameters met type en returntype).

Teken het geheugenmodel voor een KaartjesAutomaat met 3 kaartsoorten. Ga ervan uit dat de constructor van de KaartjesAutomaat net is uitgevoerd.

## Opgave B Ontwerpen van de testen

Ontwerp\* een programma waarmee je elke methode van de nieuwe klassen kunt testen. Let op dat je per methode bedenkt of er randvoorwaarden\*\* zijn om de methode met succes te kunnen uitvoeren. Bedenk welke foutmelding er geprint moet worden als er niet aan de randvoorwaarde is voldaan.

Schrijf de testen die je ontworpen uit in een processing programma.

#### \*Ontwerp

In dit geval wordt met ontwerp bedoeld dat je een programma maakt waarin je elke methode van de KaartjesAutomaat en Kaartje test. Je moet dus verzinnen in welke volgorde je deze methodes aanroept en hoe vaak je dit doet en wat je met die volgorde wilt bereiken.

#### \*\*Randvoorwaarden

Een randvoorwaarde van de methode printKaartje bijvoorbeeld is dat het saldo groter of gelijk aan de prijs van het kaartje dat geprint moet worden. Als er niet aan deze randvoorwaarde is voldaan, kan deze methode de taak waar hij voor bedoeld is niet uitvoeren (en daar moet je als programmeur iets mee doen).

## Opgave C Testen en implementeren

Los vervolgens error voor error op, door de klasse Kaartje te maken en de klasse KaartjesAutomaat aan te passen.

# Uitdaging KaartjesAutomaat 3

## Functionele informatie

De KaartjesAutomaat wordt verder aangepast. Er kunnen meerdere kaartjes tegelijkertijd worden gekocht (in één bestelling). Dit kunnen verschillende soorten kaartjes zijn, maar ook dezelfde.

## Technische informatie

De kaartjes die de gebruiker gaat kopen, moeten in een nieuwe klasse met de naam Bestelling komen. Bedenk zelf hoe de bestelling de gekozen kaartjes en de hoeveelheid kaartjes van één soort gaat bijhouden. Bedenk ook zelf welke methode in welke klasse ervoor zorgt dat er meerdere kaartjes van één kaartsoort geselecteerd kunnen worden.

## Opgave A Ontwerpen van de klassen

Teken het klassendiagram met de namen van de klassen, de eigenschappen van de velden (inclusief type) en de methoden (inclusief parameters met type en returntype).

Teken het geheugenmodel voor een KaartjesAutomaat met 3 kaarsoorten in één Bestelling.

## Opgave B Ontwerpen van de testen

Ontwerp een programma waarmee je elke methode van de nieuwe klassen kunt testen. Let op dat je per methode bedenkt of er randvoorwaarden zijn om de methode met succes te kunnen uitvoeren. Print een foutmelding als er niet aan de randvoorwaarde is voldaan.

## Opgave C Testen en implementeren

Schrijf de testen die je ontworpen hebt in opgave B uit in een processingprogramma.

## Opgave D

Los vervolgens error voor error op, door de nieuwe klasse te maken en de bestaande klassen aan te passen waar nodig.

Oefeningen

# Opgave Verhouding breedte/hoogte in Teller

## Opgave A

De verhouding tussen de breedte en de hoogte wordt nu afgevangen in klok door het statement

hoogte = breedte \* 0.4f;

Zorg ervoor dat de juiste verhouding tussen de breedte en de hoogte in de klasse Teller wordt afgevangen in plaats van in de klasse Klok. Test de wijzigingen goed.

## Opgave B

Welke implementatie geniet volgens jou de voorkeur:

1. Verhouding tussen de hoogte en de breedte in Klok
2. Verhouding tussen de hoogte en breedte in Teller
3. Maakt niet uit

Probeer zo goed mogelijk te formuleren waarom je voor jouw keuze hebt gekozen. Als je geen reden kunt verzinnen, mag je ook aangeven dat het om een vaag gevoel, of intuïtie gaat. Probeer dan dit zo helder mogelijk te beschrijven.

# Opgave toString in Klok

## Opgave A

Welke informatie zou je in de string representatie van Klok willen stoppen?

## Uitdaging Opgave B

Implementeer de toString in Klok op basis van het antwoord op opgave A.