Les 6 lesprogramma

Op Scholar staan enkele extra oefeningen voor het geheugenmodel.

# Opgave Voorbeeldtentamen

Voorbeeldtentamen op papier.

# Opgaven Geheugenmodel van tekenDamsteen

## Onderdeel A

Gegeven onderstaande code om een damsteen te maken.

class Damsteen {

float x, y, d;

int kleur;

boolean isGeselecteerd;

Damsteen(float x, float y, float d, int kleur) {

this.x = x;

this.y = y;

this.d = d;

this.kleur = kleur;

isGeselecteerd = false;

}

void tekenDamsteen() {

ellipseMode(CORNER);

fill(kleur);

if (isGeselecteerd) {

stroke(255, 0, 0);

}

else {

noStroke();

}

ellipse(x, y, d, d);

}

}

Damsteen d1;

void setup() {

d1 = new Damsteen(0.0, 0.0, 20.0, 255);

}

void draw() {

d1.tekenDamsteen();

}

Teken in het meegeleverde diagram het stack frame van tekenDamsteen met de juiste lokale variabele(n). op het moment dat deze methode wordt aangeroepen op de een na laatste regel

## Onderdeel B

De methode tekenDamsteen wordt nu uit de klassedefinitie van Damsteen gehaald en opnieuw gedefinieerd als globale functie zoals hieronder te zien is:

class Damsteen {

float x, y, d;

int kleur;

boolean isGeselecteerd;

Damsteen(float x, float y, float d, int kleur) {

this.x = x;

this.y = y;

this.d = d;

this.kleur = kleur;

isGeselecteerd = false;

}

}

void tekenDamsteen(Damsteen steen) {

ellipseMode(CORNER);

fill(steen.kleur);

if (steen.isGeselecteerd) {

stroke(255, 0, 0);

}

else {

noStroke();

}

ellipse(steen.x, steen.y, steen.d, steen.d);

}

Damsteen d1;

void setup() {

d1 = new Damsteen(0.0, 0.0, 20.0, 255);

}

void draw() {

tekenDamsteen(d1);

}

Teken opnieuw in het meegeleverde diagram het stack frame van tekenDamsteen met de juiste lokale variabele(n). op het moment dat deze methode wordt aangeroepen op de een na laatste regel

## Onderdeel C

Versie A is beter dan versie B. Leg uit waarom.

Reflectieopgaven

# Opgave Testen van een klasse

Bij sommige opgaven is er gevraagd om een test te schrijven voor een klasse. Geef een schets van de code die je maakt, als je een test voor een klasse schrijft.

# Opgave Functie <> Methode

Wat is de overeenkomst tussen een functie en een methode?

Wat is het verschil tussen een functie en een methode?

# Opgave Klassendiagrammen <> Geheugenmodel

Hoewel klassendiagrammen en geheugenmodellen beide informatie geven over het programma, is het soort informatie

Leg uit voor beide diagrammen uit welke informatie eruit te halen is en op welk moment in het ontstaan van het programma je het diagram kunt gebruiken. Gebruik daarbij zoveel mogelijk van onderstaande begrippen:

*Toestand van een programma, overzicht, klassen, objecten, relatie, variabelen, ontwerp, runtime*

# Opgave Opgaven en tags

## Onderdeel A

Bekijk nogmaals elke opgave en bekijk welke opgaven je hebt gemaakt, welke opgaven je denkt te beheersen en van welke opgave je onderdelen nog niet begrijpt.

## Onderdeel B

Bij elke opgave staan tags genoemd. Benoem de rol van elke tag in de opgave, of vraag dit na. Ga ook na of er een tag bij de opgave mist.