Les 4 lesprogramma

# Opgave Slider deel 2

Individueel, begrip

Hieronder is de draw lus van de slider gegeven. Verander de functies uit de oude code naar methoden van de klasse slider en test de nieuwe klasse slider in de draw-lus van processing.

void draw() {

int s1HuidigePositie = bepaalSliderPositie(s1X, s1Breedte, s1NPosities);

tekenSlider(s1X, s1Y, s1Breedte, s1Hoogte, s1HuidigePositie, s1NPosities);

}

void tekenSlider(float x, float y, float breedte, float hoogte,

int positie, int nPosities) {

float blokjeBreedte = breedte / nPosities;

noStroke();

fill(255);

rect(x, y, breedte, hoogte);

fill(#2257F0);

rect(x + positie \* blokjeBreedte, y, blokjeBreedte, hoogte);

}

int bepaalSliderPositie(float x, float breedte, int nPosities) {

float blokjeBreedte = breedte / nPosities;

if (mouseX < x) {

return 0;

}

else if (mouseX >= breedte + x) {

return nPosities - 1;

}

else {

return floor((mouseX - x) / blokjeBreedte);

}

}

Tags: methoden, functies, instantievariabelen, parameters, eigenschappen

# Opgave toString()

Klassikaal en individueel, string-representatie

## Persoon

Klassikaal

Voeg aan de klasse Persoon de methode toString die een relevante string retourneert. Test de methoden in het hoofdprogramma.

## Damsteen

Individueel

Voeg aan de klasse Damsteen de methode toString die een relevante string retourneert. Test de methoden in het hoofdprogramma.

Tags: toString

# Opgave Kaartautomaat

Tweetallen, programmeervaardigheid

De klasse KaartjesAutomaat simuleert een kaartautomaat bij een bioscoop die kaartjes kan leveren voor één film met één prijs. De film en de prijs van een kaartje wordt ingesteld via de constructor. Instanties zorgen ervoor dat een gebruiker alleen bedragen inwerpen die groter zijn dan 0 en dat de automaat alleen een kaartje afdrukt als er voldoende geld 'ingeworpen' is. Daarnaast kan het wisselgeld bepaald en teruggegeven worden. Ook wordt het totaal ingeworpen bedrag bijgehouden en kan de automaat geleegd worden (waardoor het totaal weer op 0 komt te staan). De afdruk van de kaartjes wordt gesimuleerd door de naam van het kaartje en de prijs naar de console te schrijven (eventueel samen met artistieke asciiart).

## Onderdeel A Maak het klassenontwerp

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

## Opgave B Implementeren en testen

#### Implementatieteam

Maak deze klasse en zorg ervoor dat de code zo goed mogelijk omgaat met foutieve invoer.

#### Testteam

Maak het hoofdprogramma waarin je één instantie van deze klasse test. Test door alle uitvoer met println naar het uitvoerscherm te schrijven. Verzin invoer en bedenk welke uitvoer je wilt hebben. Probeer zo grondig mogelijk te testen door naast geldige invoer ook foutieve invoer aan de ticketmachine te voeren.

Oefeningen

# Opgave Meer klassendefinities

## Onderdeel A Student

Maak de klasse student. Een student heeft een naam (voornaam en achternaam) en een nummer. Maak een bijpassende constructor en test de klasse in het hoofdprogramma

## Onderdeel B Dobbelsteen

Maak de klasse Dobbelsteen. Verzin zelf welke eigenschap(pen) een dobbelsteen moet hebben. Maak een bijpassende constructor en test de klasse in het hoofdprogramma.

# Opgave Slider toString()

Voeg aan de klasse Slider de methode toString die een relevante string retourneert. Test de methoden in het hoofdprogramma.

# Opgave Bestaande constructor gebruiken

Maak een tweede constructor voor klok waarmee je ook de tijd kunt initialiseren. Voorkom herhalende code door in de nieuwe constructor gebruik te maken van de constructor die al bestaat en de methode setTijd.

Hieronder is alvast de definitie:

Klok(float x, float y, float breedte, int uren, int minuten) {

}

# Uitdaging: Wat is een String (& hoe verhoudt zich dat tot objecten)

Draai onderstaande code en bekijk de uitvoer in de console.

class Persoon {

String naam;

Persoon(String naam) {

this.naam = naam;

}

}

void setup() {

String t = "hallo";

verander(t);

println(t);

int g = 10;

verander(g);

println(g);

Persoon p = new Persoon("han");

verander(p);

println(p.naam);

veranderAnders(p);

println(p.naam);

}

void verander(String tekst) {

tekst = "veranderd";

}

void verander(int getal) {

getal = 0;

}

void verander(Persoon persoon) {

persoon.naam = "kareltje";

}

void veranderAnders(Persoon persoon) {

persoon = new Persoon("pietje puk");

}

## Opgave A

Is een String op basis van de uitvoer een referentietype, of een primitief type.

## Opgave B

Hoewel een String zich niet lijkt te gedragen als een referentietype, blijkt een String wel een referentietype te zijn. Zoek uit wat er aan de hand is met Strings en teken het geheugenmodel van de aanroep van verander(t).

## Opgave C

Leg aan de hand van het geheugenmodel uit waardoor println(t) nog steeds de waarde t afdrukt.