Les 8 lesprogramma

# Opgave Lottomachine

## Functionele specificatie

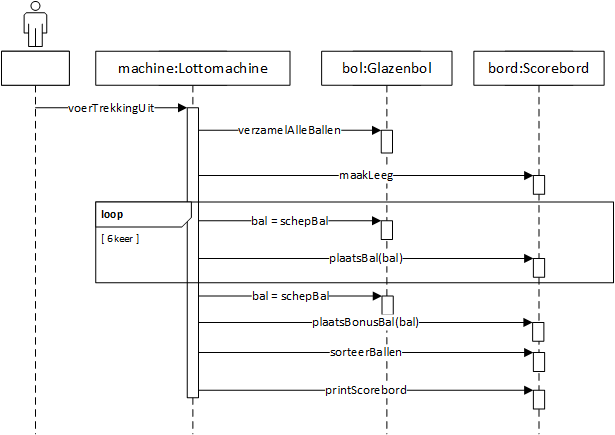
Een lottomachine bestaat uit een glazen bol die in de beginsituatie 45 balletjes, genummerd van 1 tot en met 45, bevat. Verder heeft de lottomachine een 'scorebord' dat uit zeven glazen bestaat. Het besturingsgedeelte van de lottomachine brengt de bol in beweging en zorgt er voor dat er een balletje uit de glazen bol wordt geschept dat in het eerste glas valt. Vervolgens wordt er een tweede balletje uit de bol geschept dat in het tweede glas verdwijnt. Zo worden nog vier balletjes gekozen die in het derde tot en met zesde glas vallen. Het zevende en laatste balletje dat wordt getrokken komt in het zevende glas terecht; het nummer op dit balletje wordt het bonusgetal genoemd. Bij het presenteren van de uitslag worden de eerste zes balletjes die getrokken zijn van klein naar groot getoond.

De uitslag van een lottotrekking wordt bijvoorbeeld als volgt gepresenteerd: 4 11 15 27 31 40 bonusgetal: 18

## Technische specificatie

De lottoapplicatie gaat uit vier klassen bestaan: Lottomachine, Glazenbol, Scorebord en Lottobal. Een vijfde klasse TestLottoApp bevat de main methode en wordt gebruikt om de machine te testen.

Hieronder is een sequentiediagram te zien waarin de communicatie te zien is tussen de klassen gedurende een trekking. Let op: de methode verzamelAlleBallen is o.a. verantwoordelijk voor het aanmaken van de ballen .



Het poppetje links in het diagram representeert de klasse TestLottoApp.

De implementatie van Lottobal staat hieronder weergegeven:

public class Lottobal {

private int balNummer;

public Lottobal(int nummer) {

balNummer = nummer;

}

public boolean isNummerGroterDan(Lottobal andereBal) {

return balNummer > andereBal.balNummer;

}

public String toString() {

return "" + balNummer;

}

}

## Opgave A

Maak een klassendiagram met daarin vier klassen (Lottomachine, Glazenbol, Scorebord en Lottobal) waaruit de lottoapplicatie bestaat. Gebruik het sequentiediagram en definitie van Lottobal om te bepalen welke methoden en attributen de klassen moeten hebben.

## Opgave B

Maak op basis van het klassendiagram uit opgave A de klassendefinities en implementeer alvast de constructors in elke klasse.

## Opgave C

Implementeer de overige methoden in elke klasse door het sequentiediagram van boven naar beneden af te lopen en elke methode te implementeren die je tegen komt.

Ga pas verder met een volgende methode als je de net geïmplementeerde methode getest hebt.

#### Hint 1

In de methode schepbal moet je een willekeurige bal selecteren uit een verzameling ballen. Hiervoor heb je de klasse Random nodig. Zie boek 5.4.1 en 5.4.2 (op pagina 182 tot en met pagina 184) voor een beschrijving van deze klasse, of bekijk de Java-API-documentatie.

#### Hint 2

Om de ballen op het scorebord te sorteren kun je de onderstaande code gebruiken:

public void sorteerBallen() {  
 for (int i = ballen.size(); i > 0; i--) {  
 for (int j = 0; j < i-1; j++) {   
 if (ballen.get(j).getNummer() > ballen.get(j+1).getNummer()) {  
 Lottobal bal = ballen.get(j);  
 ballen.set(j, ballen.get(j + 1));  
 ballen.set(j + 1, bal);  
 }  
 }  
 }  
 }

## Opgave D

In het sorteeralgoritme uit opgave D staat dit statement.

if (ballen.get(j).getNummer() > ballen.get(j+1).getNummer())

Deze regel zondigt tegen de (zeer zuivere) regel dat een klasse niets mag weten van de interne werking van een andere klasse. In dit geval weet Scorebord dat de klasse Bal ints gebruikt om het nummer bij te houden.

Pas de code in dit if-statement zo aan, dat er weer wordt voldaan aan de bovenstaande regel.