BP2 - Oefentoets 1

Array van cijfers

Een student kan voor een bepaald vak vijf cijfers halen die in een array van floats worden opgeslagen. Het eindcijfer is het gemiddelde van deze vijf cijfers, tenzij een of meerdere van de cijfers onvoldoende is (lager dan 5,5). In dat geval is het eindcijfer een 0,0.

Voorbeelden:

- Student A haalde: 6,0 – 6,0 – 7,0 – 8,0 – 8,0. Het eindcijfer is een 7,0 (het gemiddelde).
- Student B haalde: 6,0 – 5,0 – 4,0 – 7,0 – 8,0. Het eindcijfer is een 0,0 (want er zit een onvoldoende bij).

De array van floats is te vinden in de klasse `Student`. Hieronder staat de klassedefinitie.

```
public class Student {
    private String naam;
    private float[] cijferLijst;

    public Student(String naam, float[] cijfers) {
        this.naam = naam;
        cijferLijst = cijfers;
    }

    public String getNaam() {
        return naam;
    }

    public float[] getCijferLijst() {
        return cijferLijst;
    }
}
```

Opdracht

In een andere klasse dat het hoofdprogramma bevat, moet de methode `bepaalEindcijfer` geïmplementeerd worden. Deze methode krijgt een object van het type `Student` en retourneert het eindcijfer.
Schrijf de header en body van deze methode in Java of Processing (je mag dus zelf weten of je gebruik maakt van access modifiers).

Tweede methode

Gegeven is de volgende klasse:

```
public class Cirkel {
  public float straal;
  public int kleur;

  public void zetEigenschappen(float straal,int kleur) {
    this.straal=straal;
    this.kleur=kleur;
  }

  public float getOppervlakte() {
    return 3.1415926 * (straal * straal);
  }

}
```

De maker van deze code bedenkt dat het eigenlijk wel handig is als er naast de straal-kleur-zetter ook een methode komt waarbij je de diameter (dat is het dubbele van de straal) en de kleur moet meegeven. Hij voegt de volgende code toe binnen de klasse:

```
  public void zetEigenschappen(float diameter,int kleur) {
    zetEigenschappen(diameter/2.0,kleur);
  }
```

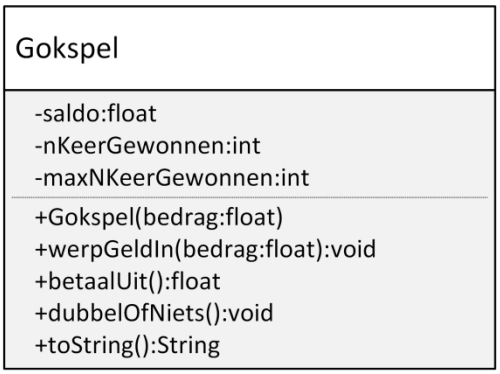
Opgave

Deze code compileert niet. Wat is de reden van deze compilatiefout?

Gokspel

We gaan een gokspelletje simuleren. Je hebt een bedrag en elke keer dat je een gok neemt kan dit bedrag verdubbeld worden of je raakt al je geld kwijt. De kans op een verdubbeling is 75%, de kans op alles verliezen is 25%.

Gegeven is onderstaand klassendiagram:



Specificatie

- Het `saldo` bevat de huidige hoeveelheid geld. Het `saldo` mag nooit kleiner dan 0 worden. De variabele `nKeerGewonnen` houdt bij hoe vaak er achter elkaar is gewonnen en de variabele `maxNKeerGewonnen` houdt bij hoe vaak er maximaal achter elkaar is gewonnen.
- De `constructor` maakt het saldo gelijk aan het meegegeven bedrag.
- De methode `werpGeldIn` verhoogt het saldo met het meegegeven bedrag.
- De methode `betaalUit` moet de huidige hoeveelheid geld retourneren en `saldo` op 0 zetten.
- Er kan daadwerkelijk gegokt worden met de methode `dubbelOfNiets`. Deze methode zorgt ervoor dat het saldo verdubbeld wordt, of op 0 gezet wordt. De kans op verdubbeling is 75% en de kans op alles verliezen is 25%. Er kan alleen gegokt worden als het saldo groter is dan 0.
- De methode `toString` retourneert een String-representatie van de klasse.

Opgave

Implementeer de hele klasse in Java of Processing (je mag dus zelf weten of je gebruik maakt van access modifiers).

Hint

Je kunt voor de `dubbelOfNiets`-methode gebruikmaken van de volgende code die een willekeurig getal genereert dat 0, 1, 2 of 3 is.

```
Random r = new Random();
int willekeurig = r.nextInt(4);
```

Een hoop boeken

Onderdeel A

Gegeven onderstaande code:

```
01  class Boek {
02    String titel;
03    String auteursnaam;
04
05    Boek(String titel,String auteursnaam) {
06     this.titel = titel;
07     this.auteursnaam = auteursnaam;
08    }
09  }
10
11  void setup() {
12    Boek boekA = new Boek("De aanslag","Mulisch");
13    Boek boekB = new Boek("De procedure","Mulisch");
14
15    boekA.titel = boekB.titel;
16    boekB.titel = "Siegfried";
17
18    println(boekA.titel);
19  }
```

Teken het geheugenmodel op het moment dat println(boekA.titel) op regel 18 wordt uitgevoerd en verklaar de uitvoer in de console (je hoeft het stack frame van println niet te tekenen).

Onderdeel B

Het programma wordt een beetje aangepast (zie vetgedrukte code) :

```
01  class Boek {
02    String titel;
03    String auteursnaam;
04
05    Boek(String titel,String auteursnaam) {
06     this.titel = titel;
07     this.auteursnaam = auteursnaam;
08    }
09  }
10
11  void setup() {
12    Boek boekA = new Boek("De aanslag","Mulisch");
13    Boek boekB = new Boek("De procedure","Mulisch");
14
15    boekA = boekB;
16    boekB.titel = "Siegfried";
17
18    println(boekA.titel);
19  }
```

Teken het geheugenmodel op het moment dat println(boekA.titel) op regel 18 wordt uitgevoerd (zonder stack frame van println) en verklaar het verschil van de uitvoer in de console met onderdeel A.