

Tentamen Programmeren in Java (IN1608WI), 11 april 2011, 9.00-11.00,
Technische Universiteit Delft, Faculteit EWI, Afdeling 2
Open boek tentamen. Je mag alleen gebruik maken van een Javaboek, niet van de slides of
practicumopdrachten.

Opgave 1 (7 + 7 + 6 punten)

Een wijnhandel koopt en verkoopt regelmatig partijen wijn, die bestaan uit een grote hoeveel flessen wijn van dezelfde soort. Voor de registratie van de wijnhandel is een klasse Partij ontwikkelt, met als attributen de naam van de wijn, het aantal flessen in de partij en de waarde van de partij.

Hieronder is een specificatie van de methoden van klasse Partij gegeven. **Je hoeft deze klasse niet te implementeren.**

Partij
Partij(nm: String, al: int, wd: double) pre: nm is niet leeg, 0 <= al, 0 <= wd post: indien pre, naam = nm, aantal = al en waarde = wd anders is er een InvalidParameterException geworpen
setNaam(nm : String) pre: nm is niet leeg post: indien pre, naam = nm, anders is er InvalidParameterException geworpen
setAantal(al: int) pre: 0 <= al post: indien pre, aantal = al, anders is er InvalidParameterException geworpen
setWaarde(wd: double) pre: 0 <= wd post: indien pre, waarde = wd, anders is er InvalidParameterException geworpen
getNaam(): String post: retourneert naam
getAantal() : int post: retourneert aantal
getWaarde(): double post: retourneert waarde

Na enige tijd blijkt dat klasse Partij moet worden uitgebreid. Er moeten methoden worden toegevoegd voor de aankoop van een nieuwe partij (van flessen met dezelfde naam) en voor de berekening van de verkoopprijs van een fles (inclusief BTW).

Bij het beantwoorden van de volgende vragen dien je een nieuwe klasse PartijPlus te schrijven die een uitbreiding is van klasse Partij. PartijPlus dient de volgende methoden te bevatten:

- a. **PartijPlus(nm: String, al: int, wd: double)**
 pre: nm is niet leeg, $0 \leq al$, $0 \leq wd$
 post: indien pre is naam = nm, aantal = al en waarde= wd,
 anders is er een `InvalidParameterException` geworpen
- b. **koop(nm : String, al : int, wd: double)**
 pre: nm gelijk is aan naam
 post: indien pre zijn er al flessen en waarde wd aan de partij
 toegevoegd, anders is er een `InvalidParameterException` geworpen

De prijs van een fles wordt berekend door de waarde van een partij te delen door het aantal flessen en aan de uitkomst 19% BTW toe te voegen. Het BTW-percentages dient als konstante aan de klasse te zijn toegevoegd.

- c. **prijsPerFles(): double**
 post: retourneert de prijs per fles, zoals hierboven
 gespecificeerd

Opgave 2(10+10+10 punten)

Gegeven klasse Voorraad waarin objecten van het type Partij kunnen worden opgeslagen:

```
public class Voorraad{
    private Partij[] partijen = new Partij[100];
    private int aantal;
    //het aantal Partij-objecten in het array

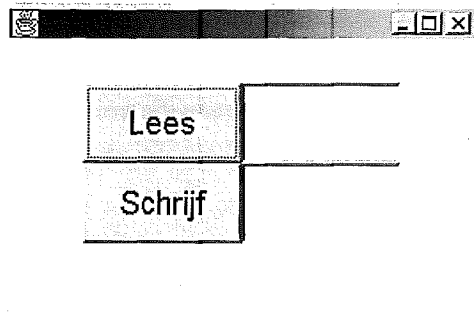
    public void voegtoe(Partij partij){
        //pre: er is nog ruimte in het array
        //post: partij is toegevoegd
        partijen[aantal] = partij;
        aantal++;
    }
}
```

Let op: in het array kunnen meerdere partijen voorkomen met dezelfde naam.

- a. Implementeer in de klasse Voorraad de methode:
public boolean komtVoor(String nm)
 post: retourneert true als in het array een partij voorkomt met naam=nm,
 retourneert anders false
- b. Implementeer in de klasse Voorraad de methode:
public int aantalFlessen(String nm)
 post: retourneert het aantal flessen van alle partijen met naam = nm
- c. Implementeer in de klasse Voorraad de methode:
public Partij duursteWijn()
 pre: aantal > 0
 post: indien pre, retourneert de partij met de wijn met de hoogste prijs
 per fles, indien er twee of meer partijen zijn met dezelfde hoogste
 prijs per fles wordt er een van de partijen geretourneert,
 retourneert anders null,

Opgave 3(10+10+15+5 punten).

In de Appendix is een implementatie gegeven van klasse ListFrame. De klasse genereert een grafische interface met twee knoppen en twee tekstvelden.



Als er op een van de knoppen wordt geklikt, gebeurt er niets.

We willen de klasse zo veranderen dat deze in staat is gegevens te lezen uit een bestand of om gegevens te schrijven naar een bestand. Het bestand bevat nul of meer String die elk op een aparte regel staan.

Het bovenste tekstveld (indexVeld) wordt gebruikt om een van de Strings te kiezen. Het onderste tekstveld (waardeVeld) zal de geselecteerde String tonen.

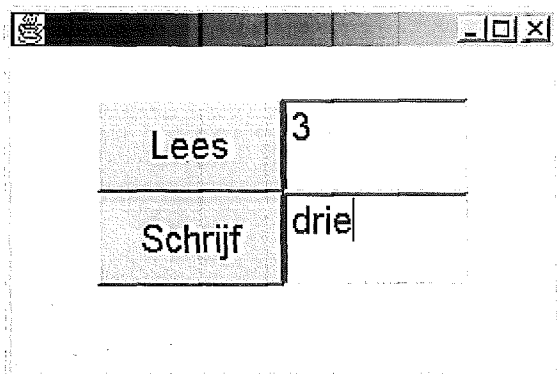
Na de verandering moeten de twee knoppen als volgt reageren:

Na het aanklikken van de leesknop, moet de inhoud van het bestand worden gelezen en opgeslagen in een ArrayList. De String die gekozen met het indexVeld wordt zichtbaar gemaakt in het waardeVeld.

Na het aanklikken van de schrijfnop wordt de inhoud van het waardeVeld in de ArrayList geplaatst op de positie aangewezen door indexVeld. Daarna wordt de inhoud van de ArrayList gekopieerd naar het bestand.

Toelichting: stel dat een bestand bevat de volgende 4 Strings bevat:

```
"nul"  
"een"  
"twee"  
"drie"
```



Als indexVeld het getal 3 bevat en de leesKnop wordt aangeklikt, wordt in het waardeVeld de String "drie" getoond.

Stel dat indexVeld het getal 2 bevat, en het waardeVeld de String "hallo".

Wordt nu de schrijfKnop aangeklikt, dan de String "hallo" in de ArrayList worden geplaatst op positie 3 en wordt daarna de inhoud van de ArrayList naar het bestand gekopieerd. De inhoud van het bestand is dan:

```
"nul"  
"een"  
"hallo"  
"drie"
```

Bestudeer de code in de Appendix goed voordat je de vragen beantwoord.

- a. Geef een implementatie van methode lees() die de Strings uit een bestand (met naam = bNaam) invoert en opslaat in een ArrayList.

```
+lees(bNaam : String): ArrayList<String>  
pre: bestand bNaam bevat 0 of meer Strings die op afzonderlijke regels  
zijn geplaatst  
post: retourneert een ArrayList die de Strings uit het bestand bevat
```

b. Geef een implementatie van methode `schrijf()` die de Strings in de `ArrayList` kopieert naar het bestand met de naam `bNaam`.

+schrijf(lijst: ArrayLijst<Streng>, bNaam : String)

pre: lijst bevat 1 of meer `String`

post: de Strings uit lijst zijn naar het bestand met de naam `bNaam` gekopieerd, elke `String` staat op een afzonderlijke regel

c. Geef een implementatie van methode `actionPerformed()`, zoals hieronder is gespecificeerd. Op pagina 3 is de werking van `actionPerformed()` nader omschreven.

+actionPerformed(e : ActionEvent)

pre: `indexVeld` bevat een niet-negatief geheel getal

post:

1. de `leesKnop` is aangeklikt:

de inhoud van bestand "`data.txt`" is ingevoerd en opgeslagen in lijst, indien het getal uit `indexVeld` een geldige index is van de lijst, is de bijbehorende waarde getoond in het `waardeVeld`, anders is de tekst "onjuiste index" getoond in het `waardeVeld`,

2. de `schrijfKnop` is aangeklikt:

indien het getal `g` uit `indexVeld` een geldige index is van de lijst, is de waarde uit het `waardeVeld` in de lijst geplaatst op index `g`, en is de lijst gekopieerd naar bestand "`data.txt`", anders is de tekst "onjuiste index" getoond in het `waardeVeld`

d. Geef aan welke andere wijzigingen in het programma moeten worden aangebracht om ervoor te zorgen dat het werkt zoals op pagina 3 is omschreven.

Appendix

```
import java.util.*;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class ListFrame extends JFrame{

    private JPanel knoppenPaneel = new JPanel();
    private JButton leesKnop      = new JButton("Lees");
    private JButton schrijfKnop   = new JButton("Schrijf");

    private JPanel veldenPaneel  = new JPanel();
    private JTextField indexVeld = new JTextField();
    private JTextField waardeVeld = new JTextField();

    private ArrayList<String> lijst;

    public ListFrame(){

        setBounds(100,100,300,200);
        setLayout(null);
        Container c = getContentPane();

        knoppenPaneel.setBounds(50,50,100,100);
        knoppenPaneel.setLayout(new GridLayout(2,1));
        knoppenPaneel.add(leesKnop);
        knoppenPaneel.add(schrijfKnop);
        c.add(knoppenPaneel);

        veldenPaneel.setBounds(150,50,100,100);
        veldenPaneel.setLayout(new GridLayout(2,1));
        veldenPaneel.add(indexVeld);
        veldenPaneel.add(waardeVeld);
        c.add(veldenPaneel);

        setVisible(true);
    }

    public ArrayList<String> lees(String bestand){
        return null;
    }

    public void schrijf(ArrayList<String> lijst, String bestand){

    }

    public void actionPerformed(ActionEvent e){
    }
}
```