

25 punten

OPGAVE 1

De onderdelen a en b van deze opgave gaan over de klassen Leerling en Klas. De definitie van deze klassen vindt u hieronder.

```
public class Leerling {
    private String naam;

    public Leerling(String naam) {
        this.naam = naam;
    }

    public String getNaam() {
        return naam;
    }
}

public class Klas {
    private String naam;
    private int aantal;
    private Leerling[] leerlingen;

    public Klas(String naam, int maxAantal) {
        this.naam = naam;
        this.leerlingen = new Leerling[maxAantal];
    }

    public void voegToe(Leerling l) {
        if (aantal < leerlingen.length) {
            leerlingen[aantal] = l;
            aantal++;
        }
    }

    public void voegToe(String naam) {
        Leerling l = new Leerling(naam);
        voegToe(l);
    }

    public Leerling getLeerling(int n) {
        if (n >= 0 && n < leerlingen.length) {
            return leerlingen[n];
        } else {
            return null;
        }
    }
}
```

Het onderstaande fragment wordt uitgevoerd. Gemakshalve zijn de opdrachten in het fragment genummerd.

```
(1) Klas klas = new Klas("1a",3);  
(2) Leerling leerlingA = new Leerling("Jan");  
(3) klas.voegToe(leerlingA);  
(4) String piet = "Piet";  
(5) klas.voegToe(piet);
```

- 1a Teken het geheugenmodel na uitvoering van regel 1 t/m regel 5. Strings moeten als aparte objecten worden opgenomen in het diagram.

Vervolgens worden de volgende opdrachten uitgevoerd:

```
(6) Leerling leerlingB = new Leerling("Jan");  
(7) boolean gelijk1 = leerlingB == leerlingA;  
(8) Leerling leerlingC = klas.getLeerling(1);  
(9) boolean gelijk2 = leerlingC.getNaam() == piet;  
(10) Leerling leerlingD = klas.getLeerling(2);
```

- 1b Wat zijn de waarden van de variabelen `gelijk1` en `gelijk2` en `leerlingD` na uitvoering van dit fragment. U hoeft uw antwoord niet te motiveren.

Gegeven is het onderstaande fragment:

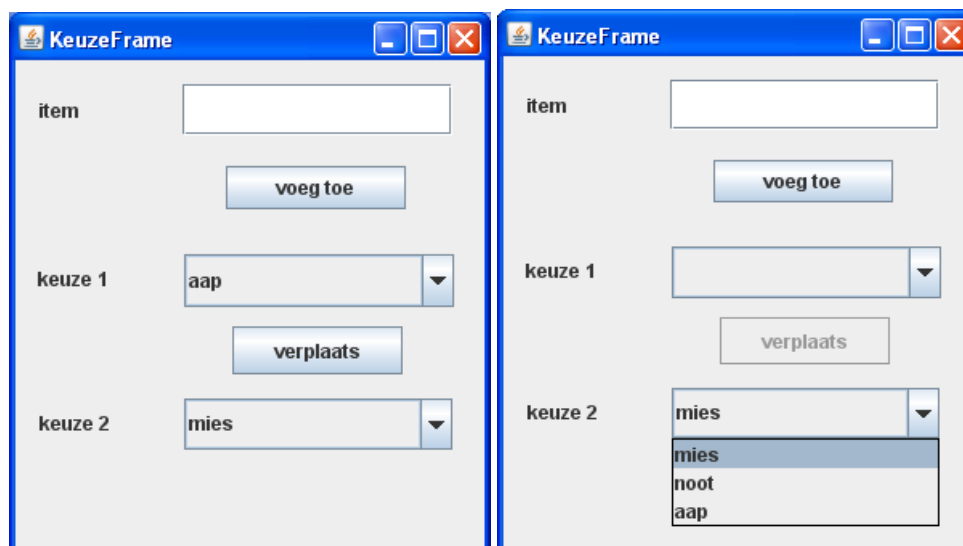
```
int i = 1;  
int j = 0;  
while (i <= 4) {  
    i++;  
    if (i%2 == 1)  
        i++;  
    j = j + i;  
}
```

- 1c Wat is de waarde van `i` en `j` na verwerking van dit fragment? U hoeft uw antwoord niet te motiveren.

25 punten

OPGAVE 2

In deze opgave ontwerpen we een applicatie `KeuzeFrame` om de werking van de Swing-klasse `JComboBox` (voor keuzelijsten) te testen. U zult een aantal methoden uit deze klasse en uit de klasse `JButton` nodig hebben. Een deel van de interface van deze twee klassen vindt u in de bijlage na de tentamenopgaven. Hieronder ziet u de grafische gebruikersinterface van de applicatie.



Klikt de gebruiker op de knop `voegToeKnop` met opschrift "voeg toe" dan wordt de inhoud van het tekstveld `itemVeld` (mits dit niet leeg is) onderaan in de keuzelijst `keuze1` toegevoegd. Het toegevoegde item wordt tevens het geselecteerde item van `keuze1`. Het tekstveld `itemVeld` wordt weer leeg gemaakt en de knop met opschrift "verplaats" wordt klikbaar (enabled).

Klikt de gebruiker op de knop `verplaatsKnop` met opschrift "verplaats" dan wordt het geselecteerde item van de keuzelijst `keuze1` verplaatst naar onderaan in de keuzelijst `keuze2`. Was het item het laatste item van `keuze1` dan wordt het item in `keuze1` dat er aan voorafging het nieuwe geselecteerde item. Was het niet het laatste, dan wordt het item dat erop volgde het nieuwe geselecteerde item. Als `keuze1` na het verplaatsen van een item leeg is, dan wordt de knop `verplaatsKnop` weer onklikbaar (disabled), zoals in de rechter figuur.

In bovenstaand voorbeeld heeft de gebruiker de teksten "aap", "noot" en "mies" achtereenvolgens toegevoegd aan `keuze1` en vervolgens tweemaal geklikt op de `verplaatsKnop`. De toestand is dan zoals getoond in de linker figuur. Na nog éénmaal klikken op deze knop en het openklappen van `keuze2` is de toestand zoals in rechter figuur.

- 2a Geef de code die nodig is om de `verplaatsKnop` te declareren, te initialiseren, te voorzien van een passende tekst en onklikbaar te maken.
- 2b Geef een implementatie (kop en romp) van de eventhandler `voegToeKnopAction` voor `voegToeKnop`.



Let op:

Een toegevoegd item is niet automatisch het geselecteerde item. U moet programmacode opnemen om het toegevoegde item expliciet te selecteren. Het eerste item in een keuzelijst heeft index 0.

- 2c Geef een implementatie (kop en romp) van de eventhandler verplaatsKnopAction voor verplaatsKnop.

Let op:

U moet wederom programmacode opnemen om het gewenste item expliciet te selecteren.

25 punten

OPGAVE 3

Een groepje collega's wil samen een voetbalpool opzetten voor het voorspellen van de uitslag van het wereldkampioenschap voetbal. Elke deelnemer moet voorspellen:

- welke twee landen finalist zijn: dit levert 1 punt op per correct voorspelde finalist
- welke finalist de winnaar wordt: dit levert 2 punten extra op.

Voor de voetbalpool moet een klasse VoetbalPool worden ontworpen waarin de voorspellingen van maximaal 100 deelnemers kunnen worden vastgelegd (er kunnen ook minder dan 100 deelnemers meedoen).

Van iedere deelnemer wordt zijn/haar eigen naam en de namen van de voorspelde winnaar en voorspelde verliezer vastgelegd. Daarvoor wordt een klasse Deelnemer ontworpen. Wanneer de finale is gespeeld moet de score van een deelnemer kunnen worden bepaald op basis van de namen van de daadwerkelijke winnaar en verliezer.

Om de voetbalpool te kunnen organiseren bevat de klasse VoetbalPool de volgende gebruiksmogelijkheden:

- Het toevoegen van deelnemers. Daarbij wordt de naam van de deelnemer en zijn/haar voorspellingen gespecificeerd. Wanneer het maximum aantal deelnemers al is bereikt, dan wordt de deelnemer niet meer toegevoegd.
- Het bepalen van de uitslag van de voetbalpool nadat de finale is gespeeld. Op basis van de daadwerkelijke winnaar en verliezer van de finale wordt een String teruggegeven die de uitslag bevat. Wanneer deze String uitgeprint zou worden ziet deze er als volgt uit:

```
Josje 4 punten
Peter 0 punten
Nike 3 punten
Harold 1 punten
```

- 3a Maak een klassendiagram voor de voetbalpool. Vergeet daarbij de typen van attributen, parameters en terugkeerwaarden niet. Specificeer in het klassendiagram ook de constructors van de gebruikte klassen.

Aanwijzing:

Get-methoden die enkel de waarde van een attribuut teruggeven hoeft u niet op te nemen.

- 3b Geef een volledige implementatie van klasse VoetbalPool (dus inclusief attributen, constructor, en methoden).
- 3c Geef een implementatie van de methode in klasse Deelnemer waarmee het aantal behaalde punten van de deelnemer kan worden bepaald.



25 punten

OPGAVE 4

In deze opgave ontwerpen we een aantal methoden in de klasse `Temperatuur`. Deze klasse bevat een attribuut `temp` waarin voor elke dag van een bepaald jaar de maximum temperatuur is opgenomen. Het element `temp[0]` bevat de maximum temperatuur op 1 januari, `temp[1]` die op 2 januari enzovoorts tot aan `temp[364]` de maximum temperatuur op 31 december (als het jaar geen schrikkeljaar is). Hieronder vindt u een deel van de definitie van deze klasse. De klasse wordt onder andere gebruikt voor het bepalen van de lengte van de langste zomerse periode, en om te onderzoeken of er een hittegolf heeft plaatsgevonden.

Een zomerse periode is een aaneengesloten rij dagen waarop iedere dag de maximale temperatuur minstens gelijk is aan 25° Celsius.

Een hittegolf is een aaneengesloten periode van minimaal 5 dagen waarin op elk van de dagen de temperatuur minstens gelijk is aan 25° Celsius en bovendien op minstens 3 dagen van deze periode de temperatuur minstens gelijk is aan 30° Celsius.

```
public class Temperatuur {
    private double[] temp;
    private int jaar;
    private static final double TROPISCH = 30.0;
    private static final double ZOMERS = 25.0;

    /**
     * Geeft het dagnummer van de warmste dag van het jaar,
     * waarbij 1 jan dagnummer 1 heeft en 31 december dagnummer 365
     * (of 366 als het jaar een schrikkeljaar is).
     * Als er meer dan een dag is, waarop de warmste temperatuur wordt
     * aangenomen, retourneert de methode het nummer van de eerste van
     * deze dagen.
     * @return het dagnummer van de warmste dag
     */
    public int warmsteDag() {
        // opgave a
    }

    /**
     * Bepaalt de lengte van de langste zomerse periode in het jaar.
     * @return de lengte van de langste zomerse periode
     */
    public int langsteZomerPeriode() {
        // opgave b
    }

    /**
     * Onderzoekt of in er in het jaar van een hittegolf plaatsvond.
     * @return true als er een hittegolf plaatsvond
     */
    public boolean hittegolf() {
        // opgave c
    }
}
```



- 4a Geef een implementatie van de methode warmsteDag.
- 4b Geef een implementatie van de methode langsteZomerPeriode.
- 4c Geef een implementatie van de methode hittegolf.

Bijlage bij opgave 2

Klasse JButton

Constructor Summary

[JButton\(\)](#)
Creates a button with no set text or icon.

Method Summary

<u>String</u>	<u>getText()</u> Returns the button's text.
void	<u>setEnabled(boolean b)</u> Enables (or disables) the button.
void	<u>setText(String text)</u> Sets the button's text.

Klasse JComboBox

Constructor Summary

[JComboBox\(\)](#)
Creates a JComboBox with a default data model.

Method Summary

void	<u>addItem(Object anObject)</u> Adds an item to end of the item list.
<u>Object</u>	<u>getItemAt(int index)</u> Returns the list item at the specified index.
int	<u>getItemCount()</u> Returns the number of items in the list.
int	<u>getSelectedIndex()</u> Returns the first item in the list that matches the given item.
void	<u>removeItemAt(int anIndex)</u> Removes the item at anIndex.
void	<u>setSelectedIndex(int anIndex)</u> Selects the item at index anIndex.