

## Examenvragen Programmeren I, januari 2009-2010

(naar het geheugen van Lisa Bouckaert en Wietse Heremans)

### 1. (2 punten)

Schrijf een programmafragment dat telkens drie getallen (x,y,z) waarvan de som 2010 is afprint met  $0 < x < y < z$ . Gebruik zo weinig mogelijk if-else (er is een oplossing zonder if-else).

Bvb. 1, 2, 2007  
1, 3, 2006  
....  
669, 670, 671

Dit mag eventueel in een andere volgorde afgeprint worden.

### 2. (4 punten)

Als je een klasse long wil toevoegen aan een package admin moet je long.class opslaan in de map admin. Maar dit is niet voldoende. Wat moet er nog gebeuren?

Schrijf een stukje code waarin je weergeeft of de eerste letter van een stringobject str een hoofdletter is.

return

Stel matrix een 2-dimensionale tabel met minstens één element, vul dan de code aan zodat totaal de som is van het aantal rijen en aantal kolommen van matrix:

Int totaal =

Primitieve types zijn geen subclasses van de klasse Object. Toch zal de volgende javacode compileren. Waarom?

Object object = 13;

### 3. (10 punten)

Een programma voor een sociale netwerksite heeft een klasse Groep die een groep bijhoudt. Deze klasse heeft een methode getLeden() die een Array van de leden van die groep weergeeft. De leden van zo'n groep zijn personen voorgesteld door de klasse Persoon. Deze klasse heeft onder andere de methodes getFamilienaam () en getWoonplaats() die de familienaam en woonplaats van een persoon in String teruggeven.

Nu willen we kunnen herkennen of een groep een bepaalde eigenschappen voldoet. Dit kan een methode van de vorm

Boolean checkEigenschap(Groep groep)

die controleert of een groep aan een bepaalde eigenschap voldoet. Zo kunnen we de volgende eigenschappen willen controleren:

- ZelfdeWoonplaats: controleert of alle leden van de groep dezelfde opgegeven woonplaats hebben.
- BevatFamilienaam: controleert of de groep een lid met de opgegeven familienaam heeft.
- MinimumGrootte: controleert of een groep minstens een bepaald aantal leden bevat.

- A. Schrijf de interface/abstracte/klassen Herkenner, ZelfdeWoonplaats, BevatFamilienaam en ????. Hou er rekening mee dat er altijd nieuwe eigenschappen bij kunnen komen en dat deze van een geheel andere aard kunnen zijn dan de voorbeelden hier gegeven.
- B. Schrijf een methode filter die met als parameters een lijst van groepen en een herkenner. Deze methode moet een nieuwe lijst van groepen retourneren bestaande uit de groepen van de opgegeven lijst die aan de eigenschap van de opgegeven herkenner voldoen.
- C. Schrijf nu kort (5 lijntjes) neer hoe je met filter volgende groepen uit een lijst met groepen zou halen:
  - Alle groepen waarvan de leden dezelfde woonplaats hebben.
  - Alle groepen met een lid met familienaam 'Obama'.
  - Alle groepen met een lid met familienaam 'Van Rompuy'.
  - Alle groepen met minimum 1000 leden.
  - Alle groepen met minimum 2 leden.

4. (3 punten)

Een bag is een manier van ints in een tabel noteren. Elke int in de tabel wordt gevolgd door een andere int, die aanduidt hoeveel keer deze voorkomt. Als we de volgende elementen

-1 4 4 4 4 4 7 7

als bag noteren, krijgen we een tabel met de volgende inhoud:

-1 1 4 5 7 2

Met andere woorden, -1 komt één keer voor, 4 vijf keer en 7 twee keer, in die volgorde.

Schrijf een methode *public int[] flatten (int[] bag)* die zo'n bag als parameter neemt, en de uitgeschreven variant ervan returned. Als we dus de array -1 1 4 5 7 2 ingeven, moet de array -1 4 4 4 4 4 7 7 teruggegeven worden.

5. (4 punten)

Beschouwen we opnieuw een bag zoals in de vorige opgave. Schrijf een methode *public void add (int[] bag, int c)* die een int c toevoegd aan een ingegeven bag. Dit element moet op de juiste positie komen, dit wil zeggen, alle elementen moeten gerangschikt blijven van klein naar groot. De lengte van de bag mag hierbij niet veranderen (het is een array en de lengte van een array is constant). Als de ingegeven bag de volgende is:

1 2 4 1 6 3 9 2

en we voegen het getal 6 toe, moet de bag veranderen naar:

1 2 4 1 6 4 9 2

Voegen we hierbij het getal 3 toe, dan wordt de bag:

1 2 3 1 4 1 6 4

Als we het getal 10 toevoegen, gebeurt er niets.

6. (3 punten)

In de bibliotheek van Java vinden we een interface Icon met onder andere volgende methodes:

```
Public interface Icon {  
    public int getWidth(); // geeft de breedte.  
    public int getHeigth(); //geeft de hoogte.  
    ...  
    public void paintIcon(Component c, Graphics g, int x, int y); // tekent Icon op de ingegeven  
    coördinaten x en y.  
}
```

(niet panikeren: deze interface is niet gezien in de les... dat is de bedoeling)

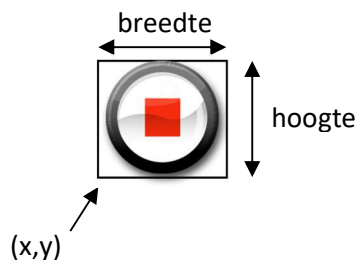
De methode paintIcon() maakt gebruik van een Component c waarop het icoon wordt afgebeeld, en Graphics g. De eigenlijke werking van deze variabelen is niet belangrijk voor de opgave.

Een Icon stelt een afbeelding voor. Een nadeel is echter dat deze afbeelding telkens dezelfde hoogte en breedte heeft. Daarom maken we gebruik van de klasse ResizeableIcon die Icon implementeerd en een object van het type Icon als parameter opneemt in zijn constructor.

Tevens bevat deze klasse de methoden setIconWidth en setIconHeight waarmee een nieuwe breedte en hoogte kunnen ingesteld worden.

Schrijf nu de volledige broncode van deze klasse ResizeableIcon neer. Een object van deze klasse moet gebruikt kunnen worden overal waar Icon gebruikt kan worden.

Het middelpunt moet hetzelfde blijven als de nieuwe lengte en breedte ingesteld worden. Als de basisafbeeling van ResizeableIcon als volgt werd afgebeeld (met methode paintIcon):



Dan beeldt ResizeableIcon de afbeelding met nieuwe hoogte en breedte als volgt af:

