

Opgave 1.

- a) Kunnen bij de declaratie van een class meerdere classes achter het reserved word 'extend' staan?  
Kunnen zelfgemaakte classes 0 superclasses hebben?  
Kan een class meerdere subclasses hebben?  
Kan een class meerdere interfaces implementeren?
- b) Indien een object het type Iterable heeft kunnen de elementen van dit object alle langsgelopen worden met een for-each loop.

Gegeven is de declaratie

```
ArrayList<StringBuffer> barcodeList
```

en dat de barcodeList gevuld is.

Tenslotte is gegeven dat objecten die instanties zijn van de class ArrayList het type Iterable hebben.

Schrijf nu een for-each loop waarmee de barcodes in de barcodeList ieder op een aparte regel worden afgedrukt.

- c) Wanneer is er bij een methode sprake van overloading?  
wanneer is er bij een methode sprake van polymorfisme?
- d) Gegevens is een

```
interface Brandbaar {  
    ..... // inhoud van de interface doer er niet toe  
}
```

Aangezien Brandbaar geen class maar een interface is, is het niet mogelijk om objecten te maken die een instantie zijn van Brandbaar. Tegelijkertijd definieert een interface een type, waardoor het mogelijk is om een variabele van het type Brandbaar te declareren als volgt:

Brandbaar hout;

Leg uit hoe aan de referentie hout een Object met het type Brandbaar gehangen zou kunnen worden.

- e) Kan aan een final variabele een waarde worden toegekend?  
Kan een final class een superclass hebben?  
Kan een final methode een override doen?
- f) - Wat is een abstracte methode?  
- Wat is een abstracte class?  
- Wanneer is het verplicht om een class abstract te maken?  
- Wat is het nut om in een abstracte class een constructor te hebben?  
- Is het mogelijk dat een abstracte class geen abstracte methodes bevat?  
- Abstracte methodes kunnen in abstracte classes staan. Waarin kunnen ze nog meer staan?

Opgave 2.

Gegeven is de interface

```
interface Kopieerbaar extends Cloneable {  
    public Object clone();  
}
```

Het bedrijf Dutch Import & Export importeert en exporteert stoelen, tafels en banken. Voor deze artikelen zijn classes Stoel, Tafel en Bank gemaakt. Alledrie deze classes hebben het type Kopieerbaar. Er moet nog een class gemaakt worden die gebruikt kan worden om de drie verschillende voorraden aan stoelen, tafels en banken in op te slaan.

- a) Maak hiertoe een specificatie van een generiek Voorraad-ADT met een default constructor, een methode inslag() die een kopie van een gegeven artikel toevoegt aan de voorraad, een methode size die de grootte van de voorraad retourneert en een methode uitslag() die een willekeurig artikel uit de voorraad verwijdert en dit verwijderde artikel als functieresultaat retourneert.
- De type-parameters van het ADT moeten beperkt (bound) worden tot objecten die het type Kopieerbaar hebben.
- b) Maak voor het Voorraad-ADT een implementatie op basis van de bij onderdeel a) gemaakte specificatie. Doe dit op zo'n manier dat het mogelijk is om een willekeurig aantal artikelen in de voorraad op te slaan. Voor de te gebruiken datastructuur mag geen gebruik gemaakt worden van al in de API geïmplementeerde datastructuren zoals ArrayList.

Beoordelingstabel

| opgave | a | b  | c | d | e | f | totaal |
|--------|---|----|---|---|---|---|--------|
| 1      | 3 | 2  | 2 | 2 | 3 | 4 | 16     |
| 2      | 8 | 12 |   |   |   |   | 20     |
| TOTAAL |   |    |   |   |   |   | 36     |

Het cijfer E van het tentamen wordt als volgt uit het totaal T verkregen :  
 $E = T/4 + 1$