

Opgave 1.

Gegeven is het onderstaande gedeelte van een class X

```
class X {
    private Object inhoud;

    X () {
        ....
    }

    ....

    public Object clone () {
        // retourneert een deep copy van het X-object
        ....
    }
}
```

a) Geef een specificatie voor een ADT van een Bag van X-objekten.

De specificatie voor de de Bag van X-objekten moet buiten een default constructor operaties bevatten om een kopie van een X-object toe te voegen, een element te verwijderen en het totale aantal elementen in de bag op te vragen. Verder moet de specificatie een methode clone() bevatten om een deep copy van de bag van X-objekten te maken.

Geef bovenaan in de specificatie een abstracte beschrijving van de bag van X-objekten via "elementen", "structuur" en "domain".

Vermeld bij iedere methode de PRE- en POST-conditie.

b) Implementeer de specificatie uit onderdeel a) van deze opgave.

Realiseer de bag van X-objekten met behulp van een array dat initieel 20 elementen bevat en dat zich verdubbelt als er een element moet worden toegevoegd terwijl het vol is.

Maak hulpmethodes voor eventuele deelproblemen.

Opgave 2.

Bij deze opgave geldt dat deelproblemen in aparte methodes in de juiste class geprogrammeerd dienen te worden. Verder mogen gevraagde methodes in latere onderdelen, indien nodig, gebruikt worden alsof ze gegeven waren.

Gegeven zijn de onderstaande gedeelten van classes en interfaces

```
class IntVerzameling {
    private int[] intArray;
    private int aantalElementen, maxAantalElementen;

    IntRI (int maxAantalElementen) {
        intArray = new int[maxAantalElementen];
        aantalElementen = 0;
        this.maxAantalElementen = maxAantalElementen;
    }
}
```

```
boolean isElement (int getal) {
    // retourneert true als getal in intVerzameling zit
    ....
}
```



```
void voegToe (int getal) {
    // getal zit na aanroep van voegToe() in intVerzameling
    ....
}
```

```
public Object clone () {
    // retourneert een deep copy van intVerzameling
    ....
}
```

```
interface CardinalVerzamelingInterface {
    static final int MAX_AANTAL_ELEMENTEN = 250;
}
```

```
class CardinalVerzameling extends IntVerzameling
implements CardinalVerzamelingInterface {
    ....
}
```

a) Schrijf een defaultconstructor voor de class CardinalVerzameling. Schrijf een methode clone() voor de class CardinalVerzameling.

b) Aan objecten van het type CardinalVerzameling mogen geen negatieve getallen, maar alleen positieve getallen of het getal 0 worden toegevoegd. Maak in de class CardinalVerzameling een dienovereenkomstige override van de methode voegToe().

Bedenk zelf een manier hoe de methode voegToe() in de class CardinalVerzameling negatieve getallen afhandelt.

c) Als gegeven is dat twee intVerzamelingen A en B gelijk zijn indien

1. alle elementen van A in B zitten en
 2. alle elementen van B in A zitten,
- override dan in de class intVerzameling de methode equals() uit de class Object met een methode public boolean equals (Object rhs) welke test of twee intVerzamelingen gelijk zijn.

d) Het kan in programma's soms handig zijn om een lege interface te gebruiken.

Hoe heet zo'n lege interface?
 Wat is het doel van zo'n lege interface?

e) Bespreek wat wrapper classes zijn en waarvoor ze gebruikt worden.

f) Declareer een array met 75 elementen waarin zowel objecten van het type IntVerzameling als van het type CardinalVerzameling kunnen worden opgeslagen, maar waarin geen objecten van andere types kunnen worden opgeslagen.

Beoordelingstabel :

Opgave	a	b	c	d	e	f	Totaal
1	5	12					18
2	4	4	4	2	2		18

TOTAAL

36

Het eindcijfer B voor het eerste deeltentamen volgt uit het punten totaal n als volgt: $E = T/4 + 1$