

Opgave 1.

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

Gegeven zijn de onderstaande classes

```
class Coördinaat {
    private double x, y;

    Coördinaat (double x, double y) {
        this.x = x;
        this.y = y;
    }
}
```

```
class Cirkel {
    private Coördinaat middelpunt;
    private double straal;
    ...
}
```

- a) Schrijf een constructor
Cirkel (double x, double y, double straal)
voor de class Cirkel.
- b) Override in de class Cirkel de methode equals() uit de class Object met een public methode equals().
- c) Schrijf een methode "public Object clone()" voor de class Cirkel.
----- einde onderdeel c), vervolg gegeven ----

Gegeven zijn de onderstaande classes

```
class Kleur {
    ....

    public Object clone () {
        ....
    }

    public boolean equals (Object obj) {
        ....
    }
}
```

De twee gegeven methodes uit de class Kleur kunnen, indien nodig, gebruikt worden zonder deze te implementeren.

```
class GekleurdeCirkel extends Cirkel {
    private Kleur kleur;
    ....
```

- d) Schrijf de body van de gegeven constructor in de class GekleurdeCirkel. Deze constructor moet een kopie van het Kleur-object in het GekleurdeCirkel-object stoppen.
- e) Schrijf de body van de methode equals() uit de class GekleurdeCirkel.

Opgave 2.

Gegeven is de onderstaande class voor knopen in een dubbelgeketende circulaire lijst (doubly linked circular list).

```
class Knoop {
    String data;
    Knoop prior, next;

    public Knoop (String data) {
        this(data, null, null);
    }

    public Knoop (String data, Knoop p, Knoop n) {
        this.data = new String(data);
        prior = p;
        next = n;
    }
}
```

Programmeer een class lijst die m.b.v. de class Knoop een circulaire lijst implementeert. Deze class lijst moet er voor zorgen dat ieder lijst-object een referentie naar het eerste en naar het laatste element van de circulaire lijst heeft en het aantal knopen bijhoudt.

In de class lijst moeten de volgende methodes aanwezig zijn:

```
public void init ();
/* PRE -
    POST - de lijst is leeg.
*/

public void insert (String data);
/* PRE -
    POST - lijst-POST = lijst-PRE +
           een knoop met als inhoud een kopie van data
*/

public boolean aanwezig (String data);
/* PRE -
    POST - TRUE: er is een knoop in lijst met inhoud data
        FALSE: er is geen knoop in lijst met inhoud data
*/

public void remove (String data);
/* PRE - het aantal knopen in lijst-PRE met inhoud data is > 0
    POST - het aantal knopen in lijst-POST met inhoud data is >= 0
*/

Verder moeten een parameterloze constructor en een methode
"public Object clone()" aanwezig zijn in de class lijst.
```

Opgave 3.

- Aan een lege B-boom van de orde 1 worden, in de gegeven volgorde, de getallen 3, 5, 7, 9, 11, 13, 15, 17, 19, 12, 14, 16, 18 toegevoegd. Teken de resulterende B-boom na het toevoegen van deze getallen. Vervolgens worden, in de gegeven volgorde, de getallen 16 en 3 verwijderd. Teken de B-boom na het verwijderen van ieder van deze getallen.
- Wanneer is een binaire boom een complete boom? Wanneer is een complete boom een hoop (heap)? Beschrijf het toevoegen van een element en het verwijderen van het element in de wortel van een complete boom die een hoop is.
- Collisions bij hashing worden opgelost met collision resolution strategies. Noem de naam en beschrijf de werking van twee van de in het boek besproken collision resolution strategies.

Beoordelingstabel :

Opgave	a	b	c	d	e	Totaal
1	4	8	8	4	4	28
2		28				28
3	12	12	10			34
TOTAAL						90

Het eindcijfer E voor het tentamen volgt uit het puntentotaal T als volgt: $E = T/9+1$