

Opgave 1.

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 classes

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

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

```
class Kleur implements Cloneable {  
    ....  
  
    public Kleur clone () {  
        ....  
    }  
  
    public boolean equals (Object obj) {  
        ....  
    }  
}
```

De twee gegeven methodes uit de class Kleur hoeven niet geïmplementeerd

```
class GekleurdeCirkel extends Cirkel {  
    private Kleur kleur;  
  
    GekleurdeCirkel (double x, double y, double straal, Kleur kleur) {  
        ....  
    }  
  
    ....  
}
```

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().
N.B. Maak aparte methodes in de juiste classes voor deelproblemen!

c) Schrijf een methode clone() voor de class Cirkel die een deep copy maakt.
Maak ook hier aparte methodes in de juiste classes voor deelproblemen.

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) Met de aanroepen "this()" en "super()" worden constructors aangeroepen.

Geef voor ieder van deze twee aanroepen antwoord op de vragen:

- 1- welke constructor door de aanroep wordt aangeroepen en
- 2- waar de aanroep mag plaatsvinden.

Wat is een abstract class?

Wat is een abstracte methode?

te worden. Ze kunnen, indien nodig, zonder meer gebruikt worden.

Opgave 2

Gegeven zijn de onderstaande classes die een circulaire doubly linked list definieren.

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

    Knoop (int d) {
        data = d;
        prior = next = null;
    }
    ....
}

class Lijst {
    private Knoop first, last;
    private int aantalElementen;
    ....
}
```

- Maak een default constructor voor de class Lijst die een lege lijst aanmaakt.
- Implementeer de onderstaande methode in de class Lijst m.b.v. linear search.

```
Knoop zoek (int x);
/* PRE -
   POST - Indien de lijst een knoop met inhoud x bevat is er een
           referentie naar de eerste knoop vanaf de first die x bevat
           geretourneerd, anders is er null geretourneerd.
*/
```

- Bij zoeken in een lijst wordt meestal linear search gebruikt. Een andere zoekmethode is binary search. Leg uit hoe binary search werkt en waarom deze zoekmethode bij een lijst niet gebruikt kan worden.
- De orde van linear search in een lijst is $O(n)$, m.a.w. als het aantal elementen 1000 keer zo groot wordt, dan wordt ook de rekentijd 1000 keer zo groot.

orde	n	rekentijd
$O(\log n)$	X 1000	X 10
$O(n)$	X 1000	X 1000
$O(n \log n)$	X 1000	X 10000
$O(n^2)$	X 1000	X 1000000

Geef aan wat de orde is van de volgende operaties:

- zoeken in een binaire boom
- zoeken in een binaire zoekboom
- sorteren van een array met selection sort
- sorteren van een array met quick sort
- zoeken in een hoop (heap)
- een element toevoegen aan een hoop

- Implementeer de onderstaande methode in de class Lijst.

```
boolean insert (int x, int y);
/* PRE -
   POST - FALSE: De lijst bevat geen knoop met inhoud x en er is geen
           nieuwe knoop met inhoud y toegevoegd.
           TRUE: De lijst bevat wel een knoop met inhoud x en achter
                 deze knoop is een nieuwe knoop met inhoud y toegevoegd.
*/
```

Opgave 3.

- Wanneer is een boom een binaire boom?
wanneer is een boom een volle boom?
Wanneer is een boom een complete boom?

Als in een binarire zoekboom van char's 1 knoop K zit met inhoud 'A' en er wordt een tweede knoop met inhoud 'A' toegevoegd, komt deze tweede knoop dan in de linker subboom van K, in de rechter subboom van K of ergens anders?

Wanneer is een boom een hoop (heap)?

Beschrijf hoe heapsort werkt.

- Bespreek het bij hashing gebruikte tweetal open-address hashing methodes met de namen linear probing en double hashing.

Gebruik (en leg uit) in deze bespreking de termen: key, array index, collision en hash function.

Leg bij de bespreking van linear probing de term clustering uit.

Leg ook uit wanneer bij double hashing gegarandeerd is dat er (indien aanwezig) een "open address" gevonden zal worden.

Beoordelingstabel :

Opgave	a	b	c	d	e	Totaal
1	4	8	8	4	6	30
2	4	6	6	8	6	30
3	15	15				30
TOTAAL						90

Het eindcijfer E voor het tentamen volgt uit het puntentotaal T als volgt: $E = T/10 + 1$