

Opgave 1

Voor deze opgave wordt een propeller als volgt gekarakteriseerd:

- 1) Een propeller staat of aan, of uit
- 2) Een propeller heeft een bepaald toerental.
Dit toerental is 0 als de propeller uit staat.

Na creatie moet een propeller geïnitieerd zijn als uitstaand.

- a) De specificatie van een object bestaat uit
 - 1- een abstracte beschrijving van dit object en
 - 2- een specificatie van de mogelijke operaties op dit object
 Geef aan hoe de punten 1- en 2- in een specificatie uitgewerkt worden.
- b) Geef een specificatie voor een object Propeller en realiseer deze specificatie vervolgens in een class Propeller.

 Buiten de constructor moeten er in een Propeller-object methodes zijn om de propeller aan te zetten, uit te zetten, het toerental in te stellen en het toerental op te vragen. Na het aanzetten van de propeller draait de propeller stationair, d.w.z. met een toerental van 10 omwentelingen per seconde. Het toerental kan nooit op minder dan het stationaire toerental worden ingesteld. Het toerental kan alleen worden ingesteld als de propeller aanstaat.
- c)
 - Wat is een abstracte methode?
 - Wat is een final methode?
 - Wat is een private methode?
 - Wat is een static methode?
 - Wat is een protected methode?

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

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

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

    public Knoop (int data, Knoop p, Knoop n) {
        this.data = 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 (int data);
/* PRE -
   POST - lijst POST = lijst PRE + knoop met inhoud data
*/

public boolean aanwezig (int 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 (int 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 de copy-constructor aanwezig zijn in de class Lijst.