

Opgave 1.

- a) Datum (Datum d) {
 dag = d.dag;
 maand = d.maand;
 jaar = d.jaar;
 }

Er hoeft geen copy-constructor gebruikt hoeft te worden om 's' toe te kennen aan 'achternaam' omdat Strings immutable zijn. Het probleem dat als twee referenties hetzelfde object aanwijzen, waardoor beide een andere waarde aanwijzen als via 1 van beide deze waarde veranderd wordt, kan dus niet optreden.

- b) in de class Datum

```
public boolean equals (Object rhs) {
    if (! rhs instanceof Datum) {
        return false;
    }

    Datum d = (Datum) rhs;
    return dag == d.dag && maand == d.maand && jaar == d.jaar;
}
```

in de class Mens

```
public boolean equals (Object rhs) {
    if (! rhs instanceof Mens) {
        return false;
    }

    Mens m = (Mens) rhs;
    return achternaam.equals(m.achternaam) &&
        geboortedatum.equals(m.geboortedatum) ;
}
```

- c) marker interface

Marker interfaces worden gebruikt om verschillende classes een gemeenschappelijk type te geven met het doel een gemeenschappelijke eigenschap te delen.

- d) PropedeuseStudent (Datum datum, String string,
 Tentamenresultaten tamenresultaten) {
 super(datum, string);

```
this.tentamenresultaten=new Tentamenresultaten(tentamenresultaten);
}
```

- e) de bedoeling was om alleen studerenden in het array op te kunnen slaan, m.a.w.: Studeert[]

Gezien de vage formulering van de vraag zal Mens[] ook worden goedgekeurd.

♀

Opgave 2

- a) interface QueueInterface {
- /* Elementen: getallen van het type int
 Structuur: lineair
 Domein : alle rijen van gehele getallen
 */
- /* constructors
- Queue () ;
 PRE -
 POST - Een nieuwe queue is aangemaakt.
 Deze nieuwe queue is leeg.
- Queue (Queue q);
 PRE -
 POST - Een nieuwe queue is aangemaakt.
 Deze nieuwe queue bevat een kopie van q.
- */
- void enqueue (int x);
 /* PRE -
 POST - queue-POST is queue-PRE verkregen door achter het
 laatste element van queue-PRE x toe te voegen.
- */
- int serve ();
 /* PRE - queue-PRE is niet leeg.
 POST - queue-POST is queue-PRE verkregen door het
 eerste element uit queue-PRE te verwijderen.
 De waarde van het verwijderde eerste element van
 queue-PRE is geretourneerd.
- */
- int size ();
 /* PRE -
 POST - Het aantal elementen in de queue is geretourneerd.
- */

```
}
```

b) static final int START_AANTAL_ELEMENTEN = 128;

```
private int[] queue;
private int aantalElementen;
```

```
Queue () {
    queue = new int[START_AANTAL_ELEMENTEN];
    aantalElementen = 0;
}
```

```
private void verdubbelArray () {
    int[] resultaat = new int[queue.length * 2];
    for (int i = 0; i < q.aantalElementen; i++) {
        resultaat[i] = queue[i];
    }
    queue = resultaat;
}
```

```
void enqueue (int x) {
    if (aantalElementen == q.length) {
        verdubbelArray();
    }

    queue[aantalElementen] = x;
    aantalElementen += 1;
}
```

c) Queue (Queue q) {

```
        queue = new int[q.queue.length];
        for (int i = 0; i < q.aantalElementen; i++) {
            queue[i] = q.queue[i];
        }
        aantalElementen = q.aantalElementen;
    }
```