Faculteit der Exacte Wetenschappen — toets Datastructuren

Vrije Universiteit — 17 mei 1999

tijdsduur : 1.5 uur

Opgave 1

a) gegeven is de onderstaande class

```
class X implements Cloneable {

String string;
int i;

....
....

}
```

schrijf de onderstaande methode voor deze class X

```
public Object clone (Object rhs);
/* PRE  -
```

```
                POST - een kopie (deep copy) van rhs is geretourneerd
            */
```

b) Wat betekenen de begrippen LIFO en FIFO?

Geef bij beide begrippen een voorbeeld van een datastructuur die dit begrip toepast.

c) gegeven zijn de onderstaande stukken van de classes Y en Z

```
class Y {
   String s;

   ....
   ....

   Y () {
         s = new String();
   }

   void put (String s) {
         this.s = new String(s);
   }

   String g () {
      return "Y" + s + "Y"
```

```
    }
}

class Z extends Y {
    ....
    ....

    Z () {
            super();
    }

    void put (String s) {
            super.put("Z" + s + "Z");
    }

    String g () {
            return "Z" + super.g() + "Z";
    }
}
```

als in een programma de volgende variabelen gedeclareerd zijn

```
Y y = new Y();
Z z = new Z();
Output out = new Output();
```

wat wordt er dan door de onderstaande statements afgedrukt?

```
y.put("1234");
z.put("test");
out.println(y.s);
out.println(y.g());
out.println(z.s);
out.println(z.g());
y = z;
out.println(y.s);
out.println(y.g());
out.println(z.s);
out.println(z.g());
```

d) Leg uit wat abstracte methodes zijn.

Zowel een interface als een abstracte class bevatten abstracte methodes.

Wat zijn de verschillen tussen een interface en een abstracte class?

Waarom zou het in JAVA niet mogelijk zijn om een instantie van een abstracte class te maken?

Z.O.Z. VOOR OPGAVE 2

Opgave 2.

Gegeven zijn de volgende interfaces

```
public interface Data {
    public Object clone();
    /* PRE  -
       POST - een kopie van this is geretourneerd
    */

    public int compareTo(Data d);
    /* PRE  - this is een instantie van dezelfde class als d
       POST - er is geretourneerd
                        een negatief getal, als this < d
                        nul,              als this = d
                        een positief getal, als this > d
    */
}


public interface BagInterface {
    /* Elementen: objecten van het type Data
       Structuur: geen
       Domein:          alle groepen (dubbelen toegestaan) van objecten van
                        het type Data
   */

    /* Constructors
```

```
Bag ();
PRE  -
POST - de nieuwe bag is leeg

Bag (Bag b);
PRE  -
POST - de nieuwe bag bevat een kopie van 'b'
*/

Bag include (Data x, int aantal);
/* PRE  -
   POST - 'aantal' kopieen van 'x' zijn toegevoegd aan bag-PRE
                bag-POST bevat het element 'x' 'aantal' maal meer dan
                bag-PRE
                bag-POST is geretourneerd
*/

Bag exclude (Data x, int aantal);
/* PRE  - bag-PRE bevat het element 'x' 'n' maal (n>=0)
   POST - bag-POST bevat het element 'x' 'n'-'aantal' maal
                als 'n' >= 'aantal' en 0 maal als 'n' < 'aantal'
                bag-POST is geretourneerd
*/

Bag init ();
```

```
    /* PRE  -
       POST - bag-POST is leeg
    */

    int size ();
    /* PRE  -
       POST - het aantal elementen in de bag is geretourneerd
    */

    int aantal (Data x);
    /* PRE  -
       POST - het aantal malen dat 'x' in de bag zit is geretourneerd
    */
}
```

a) Gegeven dat de eerste regel van de class Bag er als volgt uitziet

```
public class Bag implements BagInterface {
```

Het is de bedoeling om in de class Bag een bag te realiseren met een binaire zoekboom.

Voor de knopen van deze binaire zoekboom wordt de onderstaande class

Knoop gegeven:

```
class Knoop {

   Data data;
   int aantal;
   Knoop left, right;

   Knoop () {
         this(null, 0, null, null);
   }

   Knoop (Knoop k) {
         this(k.data, k.aantal, k.left, k.right);
   }

   Knoop (Data d, int aantal) {
         this(d, aantal, null, null);
   }

   Knoop (Data d, int a, Knoop l, Knoop r) {
         data = (d == null ? null : (Data) d.clone());
         aantal = a;
         left = l;
         right = r;
```

```
    }
}
```

Declareer de variabele(n) in de class Bag waarmee een binaire zoekboom gerealiseerd kan worden.
N.B. Denk er over na of deze variabelen public, protected, friendly of private moeten zijn.

b) Maak de default constructor Bag() voor de class Bag.

c) Maak ook een copy constructor voor de class Bag.

d) Implementeer de methodes include(), size() en aantal() voor de class Bag.

Beoordelingstabel

opgave	a	b	c	d	e	f	totaal
1	4	4	4	4			16
2	2	2	4	12			20
TOTAAL							36

Het eindcijfer E wordt als volgt uit het totaal T verkregen : E = T/4 + 1 .