

Opgave 1.

Gegeven zijn de volgende classes

```
class P extends Object {
    int i;
```

```
    P () {
        i = 2;
    }
```

```
    public String toString () {
        return "P: " + i;
    }
}
```

```
class Q extends P {
    int i;
```

```
    Q () {
        i = 3;
    }
```

```
    public String toString () {
        return "Q: " + i;
    }
}
```

```
class R extends Q {
    int i;
```

```
    R () {
        i = 5;
    }
```

```
    public String toString () {
        return "R: " + i;
    }
}
```

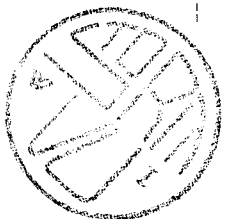
- a) Verandert er iets voor objecten die een instantie van de class P zijn indien "class P extends Object {" vervangen wordt door "class P {" ?
Zo ja, wat verandert er dan; zo nee, waarom niet?

- b) Eveneens gegeven is het onderstaande programma:

```
class Opgave1b {
    Output out;
```

```
    P p;
    Q q;
    R r;
```

```
    Opgave1b () {
        out = new Output();
        p = new P();
        q = new Q();
        r = new R();
    }
}
```



```
void start () {
    // nog in te vullen
}
```

```
    public static void main (String[] argv) {
        new Opgave1b().start();
    }
}
```

Wat drukt het bovenstaande programma af indien de body van start () de volgende statements bevat?

```
out.println(p.i);
out.println(q.i);
out.println(r.i);
out.println(((Q)r).i);
out.println(((P)r).i);
out.println(((P)q).i);
p = q;
q = r;
out.println(p.i);
out.println(q.i);
```

- c) Wat drukt het programma uit onderdeel b) van deze vraag af indien de body van start() de volgende statements bevat?

```
out.println(p.toString());
out.println(q.toString());
out.println(r.toString());
out.println(((Q)r).toString());
out.println(((P)r).toString());
out.println(((P)q).toString());
p = q;
q = r;
out.println(p.toString());
out.println(q.toString());
```

- d) Gegeven is de onderstaande class

```
class A {
    private int x;
    private String s;
```

```
    A () {
        x = 0;
        s = "";
    }
```

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

Schrijf de methode equals() voor de class A.

Opgave 2.

Gegeven is de onderstaande class voor de implementatie van knopen in een boom:

```
class Knoop {
    int data;
    Knoop links, rechts;

    Knoop (int data, Knoop links, Knoop rechts) {
        this.data = data;
        this.links = links;
        this.rechts = rechts;
    }

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

- a) Een binaire boom is een hoop (heap) indien deze binaire boom voldoet aan de hoopvoorwaarden (heap storage rules) en een complete binaire boom (complete binary tree) is.

Welke zijn de hoopvoorwaarden?

Wanneer is een binaire boom een complete binaire boom?

Geef een recursieve implementatie van de onderstaande methode

```
int hoopafwijking (Knoop k);
/* PRE - k is een referentie naar de wortel van een
complete binaire boom
POST - Het aantal knopen dat niet voldoet aan de
hoopvoorwaarden is getourneerd.
*/
```

- b) Wanneer is een binaire boom een binaire zoekboom?

Geef een recursieve implementatie van de onderstaande methode

```
int aantal (Knoop k, int x);
/* PRE - k is de wortel van de binaire zoekboom b-PRE.
POST - Het aantal maal dat x in b-PRE voorkomt is
getourneerd.
*/
```

- c) Geef een recursieve implementatie van de onderstaande methode

```
Knoop voegToe (Knoop k, int x);
/* PRE - k is de wortel van de binaire zoekboom b-PRE.
POST - b-POST is de binaire zoekboom die verkregen is uit de
binaire zoekboom b-PRE door daaraan een knoop met
inhoud x toe te voegen. De wortel van b-POST is
getourneerd.
*/
```

Opgave 3.

- a) Geef een specificatie voor een ADT van een Stack van Strings.

De specificatie voor de Stack van Strings moet buiten een default constructor operaties bevatten om een kopie van een string toe te voegen en het totale aantal elementen in de stack op te vragen. Verder moeten er een methode zijn waarmee een programma kan opvragen wat de top van de stack is en een methode waarmee een element van de stack afgehaald kan worden.

Geef bovenaan in de specificatie een abstracte beschrijving van de stack van Strings via "elementen", "structuur" en "domain".

Vermeld bij iedere methode de PRE- en POST-conditie.

- b) Implementeer de specificatie uit onderdeel a) van deze opgave.

Realiseer de stack van Strings met behulp van een eenvoudige enkelvoudig gelinkte lijst.

Beoordelingstabel	opgave	a	b	c	d	e	f	totaal
1	5	9	9	7				30
2	6	9	10					25
3	15	20						35
TOTAAL								90

Het eindcijfer E van het gehele tentamen wordt als volgt uit het totaal T verkregen : $E = T/10 + 1$.