
Opgave 1.

- a) De subclass B erft uit A: s, d, m1() (2p)
- i wordt niet geerfd omdat deze variabele private is. (2p)
- De constructor wordt niet geerfd omdat deze alleen in de class A zinvol is. (2p)
- b) super() in de constructor van de class B roept de constructor van de class A aan ZONDER DAT ER EEN NIEUW OBJECT AANGEMAAKT WORDT.
2 + 2 p
- c) 0.0
5.0
0.0
5.0
0.0
5.0
-1 punt voor iedere fout
- d) ja (2p)

ja (2p)

ja (2p)
- e) 0.0
35.0
0.0
35.0
- f) in class B
in class B
in class B
in class B

Opgave 2.

- a) /* De class Knoop hoeft alleen de default constructor te bevatten.
- ```
class Knoop {

 char data;
```

```
Knoop links,
 rechts;
```

```
Knoop () {
 char = '\u0000'; // N.B. ieder char is goed.
 links = rechts = null;
}
}
```

```
class Boom {
 private Knoop wortel;

 Boom () {
 wortel = null;
 }
}
```

```
private String toString (Knoop k) {
 if (k == null) {
 return "";
 }

 return toString(k.links) + toString(k.rechts) + k.data;
}
```

```
public String toString () {
 return toString(wortel);
}
}
```

b) voeg toe aan de class Knoop

```
Knoop (char c) {
 this.c = c;
 links = rechts = null;;
}
```

In de class Boom

```
public void insert (char c) {
 wortel = insert(wortel, c);
}
```

```
private Knoop insert (Knoop k, char c) {
 if (k == null) {
 return new Knoop(c);
 }
}
```

```

 }

 if (c <= k.data) {
 k.links = insert(k.links, c);
 } else { // c > k.data
 k.rechts = insert(k.rechts, c);
 }

 return k;
}

```

c) De drie gevallen zijn:

- 1- een knoop met 0 kinderen  
laat de links/rechts referentie van de ouder naar null wijzen
- 2- een knoop met 1 kind  
laat de links/rechts referentie van de ouder naar dit kind wijzen
- 3- een knoop met 2 kinderen  
zoek de grootste/kleinste nakomeling in de linker/rechter subboom. stop een kopie van deze nakomeling in de te verwijderen knoop. Verwijder de eerder gevonden nakomeling.

### Opgave 3.

a) Bij een B-boom van de orde  $n$  is het minimale aantal elementen dat een knoop, die niet de wortel is, kan bevatten:  $n$

Bij een B-boom van de orde  $n$  is het maximale aantal elementen dat een knoop kan bevatten:  $2^n$

Het minimale aantal elementen dat de wortel van een B-boom van de orde  $n$  kan bevatten is: 0 (1 wordt ook goed gerekend)

Het maximale aantal elementen dat de wortel van een B-boom van de orde  $n$  kan bevatten is:  $2^n$

De elementen in een knoop van een B-boom zijn stijgend gesorteerd.

Het aantal subbomen van een knoop van een B-boom met  $x$  elementen is:  $x+1$

Voor de waarden in de  $i$ -de subboom geldt dat deze alle  $< x$  zijn.

Voor de waarden in de  $(i+1)$ -ste subboom geldt dat deze alle  $> x$  zijn.

6

|       |       |     |   |     |       |    |
|-------|-------|-----|---|-----|-------|----|
|       |       | 0   | 3 |     | 9     | 12 |
|       | -2 -1 | 1 2 | 4 | 7 8 | 10 11 |    |
| 13 14 |       |     |   |     |       |    |

b) Een binaire boom is compleet als alle lagen behalve de onderste vol

2001-12-20 Antwoorden D2.txt

zijn en de onderste laag van links naar rechts gevuld is.

Een complete boom is een hoop indien voor iedere knoop geldt dat de inhoud van deze knoop  $\geq$  is dan de inhoud van de twee kinderen.