

Opgaver til stoffet gennemgået ved forelæsningen 9/11-2004

Opgave 1

Betragt et binært søgetræ som starter med at være tomt. Tegn træet på et stykke papir eller en tavle, som der ser efterhånden som følgende operationer udføres.

```
indsæt(20);  
indsæt(30);  
indsæt(35);  
indsæt(10);  
indsæt(33);  
indsæt(32);  
indsæt(17);  
indsæt(13);  
indsæt(8);
```

```
slet(35);  
indsæt(35);  
slet(10);  
slet(20);
```

Opgave 2

Du skal programmere et antal metoder, som undersøger egenskaber ved et binært træ. Der er givet en skabelon af en kildetekst, BinTree.java, som du kan finde på kursets hjemmeside, og som du kan tage udgangspunkt i.

Den indeholder en klasse for binære søgetræer med konstruktorer, og »tomme« metoder som du så skal gøre færdige, så de virker.

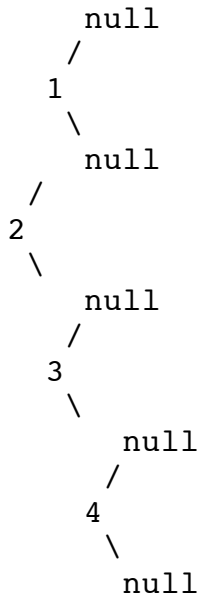
Spørgsmål 2.1

Du skal skrive en metode `public static void printTree(BinTree t)`, som skriver et træ ud, således at dybden af den enkelte deltræer er angivet ved indrykninger.

Hvis du f.eks. har oprettet et træ således:

```
BinTree t = new BinTree  
            ( new BinTree(1),  
              2,  
              new BinTree(3, new BinTree(4)) );
```

... skal du gerne få genereret følgende udskrift (eller noget der ligner):



NB: Det kan være en fordel at skrive en hjælpemetode
`public static void printTree(BinTree t, String indent)`
 hvor argument nr. 2 angiver den stribe blanktegn, som skal skrives ud til venstre for træet.

Spørgsmål 2.2

Skriv en metode

`public static boolean isSearchTree(BinTree t),`
 som returnerer `true` hvis et træ opfylder betingelsen for at være en binært SØGETræ; ellers `false`.

Spørgsmål 2.3

Skriv en metode

`public static boolean isCompleteAndBalanced(BinTree t),`
 som returnerer `true` for et træ som er lige dybt ver det hele; ellers `false`.

Bemærk: En nødvendig, men ikke tilstrækkelig betingelse for at det skal returneres `true`, er at antallet af knuder i træet er 1, 3, 7, 15, ... $(2^i) - 1$, (Du må selv om, om du accepterer et træ med 0 knuder eller ej).

Spørgsmål 2.4

Skriv en metode

`public static boolean isAVL(BinTree t),`
 som returnerer `true` hvis et træ opfylder betingelsen for at være et AVL-træ; ellers `false`.