

## Opgaver fredag den 27. august

### Opgave 1

Lav en rekursiv metode der kan finde antallet af lige tal i en liste. Metoden skal have følgende signatur:

```
public static int ligeTal(ArrayList<Integer> list)
```

Lav både en version der anvender en hjælpemetode og en der ikke gør.

### Opgave 2

Et palindrom er en tekststreng der læses ens forfra og bagfra som f.eks "Anna" og "radar". Skriv en rekursiv metode `public static boolean palindrom(String tekst)` der returnerer om teksten er et palindrom. Anvend hjælpemetoder, så der ikke skal laves substrings.

### Opgave 3

Betragt nedenstående rekursive definition af en talfølge

$$\begin{aligned} \text{tal}_0 &= 2 \\ \text{tal}_1 &= 1 \\ \text{tal}_2 &= 5 \\ \text{tal}_n &= 2 \cdot \text{tal}_{n-3} - \text{tal}_{n-1} && \text{når } n > 2 \text{ og } n \text{ er lige} \\ \text{tal}_n &= \text{tal}_{n-1} + \text{tal}_{n-2} + 3 \cdot \text{tal}_{n-3} && \text{når } n > 2 \text{ og } n \text{ er ulige} \end{aligned}$$

#### Opgave 3.1

Find værdien af det tredje, fjerde, femte og sjette tal i talfølgen; mere præcist ønskes værdien af  $\text{tal}_3$ ,  $\text{tal}_4$ ,  $\text{tal}_5$  og  $\text{tal}_6$  beregnet.

#### Opgave 3.2

Lav en **rekursiv** metode,  $\text{talN}$ , der beregner det  $n$ 'te tal i ovenstående talfølge. Mere præcist ønskes nedenstående metode realiseret

```
public int talN (int n)
```

```
/**
```

```
 * Krav:       $n \geq 0$ 
```

```
 * Post:       $\text{talN}' = n$ 'te tal i talfølgen, givet ved ovenstående definition af  $\text{tal}_n$ 
```

```
 */
```

## Opgave 4 Gennemløb af mappestruktur

Følgende metoder gennemløber en mappestruktur rekursivt

```
public static void ScanDir(String path) {
    System.out.println("[DIR]  " + path);

    // skab et File-objekt svarende til mappen path
    File file = new File(path);

    // få listen over alle filer og undermapper
    String[] names = file.list();
    for (String name: names) {
        File file2=new File(path + "/" +name);
        if (file2.isDirectory())
            ScanDir(path + "/" +name);
    }
}
```

Et gennemløb startes f.eks. med kaldet

```
String path = "c:/programmer";
ScanDir(path);
```

a) Læs metoden igennem og forstå princippet i løsningen.

**NB:** Scan ikke med start-mappen "c: / ", dels tager det meget lang tid og nogle af dem er beskyttede af styresystemet.

b) Lav en ny rekursiv løsning der gennemløber en mappe og tæller antal directories i stedet for at udskrive dem, Signaturen for metoden skal være

```
public static int ScanDirCount(String path)
```

Kaldet skal f.eks. være

```
int n = ScanDirCount("c:/programmer");
System.out.println("Antal mapper: " + n);
```

c) Udskriften i a) er ikke overskuelig, den kan se ud som

```
[DIR]  c:/programmer
[DIR]  c:/programmer/Adobe
[DIR]  c:/programmer/Adobe/Acrobat 6.0
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Es1
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Help
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Help/DAN
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Help/ENU
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Reader
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Reader/ActiveX
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Reader/Browser
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Reader/HowTo
[DIR]  c:/programmer/Adobe/Acrobat 6.0/Reader/HowTo/DAN
```

Udvid signaturen for ScanDir, så du udskrive niveauet i strukturen foran mappestien:

```
1: [DIR] c:/programmer
2: [DIR] c:/programmer/Adobe
3: [DIR] c:/programmer/Adobe/Acrobat 6.0
4: [DIR] c:/programmer/Adobe/Acrobat 6.0/Esl
4: [DIR] c:/programmer/Adobe/Acrobat 6.0/Help
5: [DIR] c:/programmer/Adobe/Acrobat 6.0/Help/DAN
5: [DIR] c:/programmer/Adobe/Acrobat 6.0/Help/ENU
4: [DIR] c:/programmer/Adobe/Acrobat 6.0/Reader
5: [DIR] c:/programmer/Adobe/Acrobat 6.0/Reader/ActiveX
5: [DIR] c:/programmer/Adobe/Acrobat 6.0/Reader/Browser
5: [DIR] c:/programmer/Adobe/Acrobat 6.0/Reader/HowTo
6: [DIR] c:/programmer/Adobe/Acrobat 6.0/Reader/HowTo/DAN
```