

Onsdag den 5. maj

Opgave 1

Lav en udgave af bubbleSort, der kan sortere et array af String-objekter, og test metoden på et array af Strings. For eksempel:

```
String[] s = {"Erna", "Elly", "Laurits", "Bertha", "Christian",  
             "August", "Marius", "John", "Tove", "Poul", "Torkild"};
```

Opgave 2

Lav to udgaver af selectionSort, der kan sortere henholdsvis et String[]-array og en ArrayList af Customer, som implementerer Comparable<Customer> (se Customer.jar). Test metoderne.

Opgave 3

Lav to udgaver af insertionSort, der kan sortere henholdsvis et String[]-array og en ArrayList af Customer. Test metoderne.

Opgave 4

I klassen java.util.Collections findes en static metode *sort*, der kan anvendes til at sortere en ArrayList af Comparable objekter. En anvendelse af metoden ser ud som følger:

```
import java.util.ArrayList;  
import java.util.Collections;  
  
public class TestSort {  
  
    public static void main(String[] args) {  
        ArrayList<String> list = new ArrayList<String>();  
        list.add("Jan");  
        list.add("Bent");  
        list.add("Thomas");  
        list.add("Karsten");  
        list.add("Dan");  
        System.out.println(list);  
        Collections.sort(list);  
        System.out.println(list);  
    }  
}
```

Prøv at udføre ovenstående program - bliver listen sorteret? Hvad er det der gør det muligt for sort-metoden at sortere listen?

Opret og indsæt et antal Customers i en ArrayList, sortér listen med Collections.sort(), og udskriv den sorterede liste.

Opgave 5

Gennemfør en systematisk måling af de tre algoritmers tidsforbrug (Bubble, Selection og Insertion) ved hjælp af metoden `System.nanoTime()`. Brug `MeasureSort.jar`, som indeholder kode til tidsmåling og kode til at generere lange lister af Strings, som kan bruges som testdata. Lav dernæst en tabel, der for hver af algoritmerne, viser sammenhængen mellem listens størrelse og tidsforbrug. Hvilken algoritme er hurtigst?

Opgave 6

Suppler tabellen fra opgave 5 med målinger af `Collections.sort()` metodens tidsforbrug. Hvilken algoritme er nu hurtigst?