Opgaveløsninger
(sæt 2)

Opgave 6: 4.21 (1 point)

```
public class MinTest {
    public static Comparable min(Comparable[] a) {
        int minIndex = 0;
        for (int i = 1; i < a.length; i++)
            if (a[i].compareTo(a[minIndex]) < 0)
                minIndex = i;
        return a[minIndex];
    }

    public static void main (String[] args) {
        String[] st = { "Bent", "Carl", "Alf", "Frederik" };
        System.out.println(min(st));
    }
}
```

Opgave 7: 4.22 (2 point)

```
public class Max2Test {
    public static Comparable[] max2(Comparable[] a) {
        int maxIndex0 = 0, maxIndex1 = 0;
        if (a[1].compareTo(a[0]) > 0)
            maxIndex0 = 1;
        else
            maxIndex1 = 1;
        for (int i = 1; i < a.length; i++) {
            if (a[i].compareTo(a[maxIndex0]) > 0) {
                maxIndex1 = maxIndex0;
                maxIndex0 = i;
            } else if (a[i].compareTo(a[maxIndex1]) > 0)
                maxIndex1 = i;
        }
        return new Comparable[] { a[maxIndex0], a[maxIndex1] };
    }

    public static void main(String[] args) {
        String[] st = { "Bent", "Carl", "Alf", "Frederik" };
        Comparable[] result =  max2(st);
        System.out.println(result[0] + " " + result[1]);
    }
}
```

Opgave 8: 4.27 (2 point, ikke-obligatorisk)

```
import java.util.*;

public class SortedArrayList {
    /**
     * Constructs an empty ArrayList.
     */
    public SortedArrayList() {
        clear( );
    }

    /**
     * Adds an item to this collection, at the end.
     * @param x any Comparable.
     * @return true.
     */
    public boolean add(Comparable x) {
        if (theItems.length == size()) {
            Comparable[] old = theItems;
            theItems = new Comparable[theItems.length * 2 + 1];
            for (int i = 0; i < size(); i++)
                theItems[i] = old[i];
        }
        int i;
        for (i = size() - 1;
             i >= 0 && theItems[i].compareTo(x) > 0;
             i--)
            theItems[i + 1] = theItems[i];
        theItems[i + 1] = x;
        theSize++;
        return true;
     }

    /**
     * Removes an item from this collection.
     * @param idx the index of the Comparable.
     * @return the item was removed from the collection.
     */
    public Comparable remove(int idx) {
        Comparable removedItem = theItems[idx];

        for (int i = idx; i < size() - 1; i++)
            theItems[i] = theItems[i + 1];
        theSize--;
        return removedItem;
    }

    /**
     * Change the size of this collection to zero.
     */
    public void clear() {
        theSize = 0;
        theItems = new Comparable[DEFAULT_CAPACITY];
    }
    /**
     * Returns the size of this collection.
     */
    public int size() {
        return theSize;
    }

    private static final int DEFAULT_CAPACITY = 10;

    private int theSize;
    private Comparable[] theItems;
}
```

Det er lidt “snyd” - men i orden - at implementere klassen ved brug af `ArrayList`:

```
public class SortedArrayList {
    public void add(Comparable x) {
        int i = list.size() - 1;
        while (i >= 0 && ((Comparable) list.get(i)).compareTo(x) > 0)
            i--;
        list.add(i + 1, x);
    }

    public Comparable remove(int i) { return (Comparable) list.remove(i); }

    public int size() { return list.size(); }

    private ArrayList list = new ArrayList();
}
```