Programmieren 1
Lernteam Semesterwoche 11

Aufgabe 1: Implementation "direktes Auswählen" (siehe ALG SW11 Seite 25ff.)

Implementieren Sie eine Klasse Sort mit einer Methode directSelect(). Testen Sie Ihre Methode. Vielleicht geht's hier schneller ohne JUnit, z.B. mit einer Ausgabe auf die Konsole.

```
//Lösungsbeispiel
public class DirectSelect
{
    int[] a;

    public DirectSelect(int[] values)
    {
        directSelect(values);
        print(values);
    }

    private void directSelect(int[] values)
    {
        for(int i=0; i<values.length-1; i++)
        {
            for(int j=i+1; j<values.length; j++)
            {
                if(values[i]>values[j])
                {
                    int temp = values[j];
                    values[j] = values[i];
                    values[i] = temp;
                }
            }
        }
    }

    private void print(int[] values)
    {
        for(int x=0; x<values.length; x++)
        {
            System.out.println(values[x]);
        }
    }
}
```

Der Pseudocode zu diesem Beispiel wurde gefunden auf:
http://public.tfh-berlin.de/~siegel/wiinfii/Sortieren_2.html

Aufgabe 2: Analyse "direktes Einfügen"

Analysieren Sie den Sortieralgorithmus "direktes Einfügen". Konsultieren Sie dazu nochmals die Unterlagen. Gehen Sie analog wie beim "direkten Auswählen" vor.

```
//Lösungsbeispiel
public class DirectInsert
{
    int[] a;

    public DirectInsert(int[] values)
    {
        directInsert(values);
        print(values);
    }

    private void directInsert(int[] values)
    {
        for(int i=1; i<values.length; i++)
        {
            int key = values[i];
            int j = i-1;

            while(j >= 0 && values[j] > key)
            {
                values[j+1] = values[j];
                j--;
            }
            values[j+1] = key;
        }
    }

    private void print(int[] values)
    {
        for(int x=0; x<values.length; x++)
        {
            System.out.println(values[x]);
        }
    }
}
```

Der (fehlerhafte) Pseudocode zu diesem Beispiel wurde gefunden auf:
http://de.wikipedia.org/wiki/Insertionsort

Aufgabe 3: Iteration vs. Rekursion, JUnit

Implementieren Sie eine Klasse Multiplication mit den beiden Methoden multIt() und multRe() (siehe Vorlesung). Testen Sie die Methoden mit Hilfe von JUnit.

Aufgabe 4: Objektdiagramm

Lösen Sie die Aufgabe 7.12 aus dem Lehrbuch.
Optionale Zusatzaufgabe 5: Refactoring
Lösen Sie die Aufgabe 7.18 aus dem Lehrbuch.

Zusammenfassung

Verschiedene Sortieralgorithmen und deren Zeit- bzw. Speicherkomplexität:

	best case	average case	worst case	zus. Speicher
Auswahl	n	n^2	n^2	1
Einfügen	n	n^2	n^2	1
Bubblesort	n	n^2	n^2	1
Quicksort	n log n	n log n	n^2	log n
Heapsort	n log n	n log n	n log n	1
Bucketsort	n	n	n log n, n^2	n
Mergesort	n log n	n log n	n log n	n