

Übungsblatt 11

Abgabe bis Montag, 11.07.2015, 23:59 Uhr

Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und evtl. benötigte Datendateien).

Aufgabe 11.1

Schreiben Sie eine Klasse `Sort`, die Methoden zum Sortieren von Arrays bereitstellt.

1. Implementieren Sie eine Methode

`static int[] union(int[] a1, int[] a2),`

der zwei **sortierte** Folgen von Integer-Zahlen übergeben werden. Diese Methode soll eine sortierte Folge zurückgeben, die alle Elemente aus `a1` und `a2` enthält.

2. Bestimmen Sie die Komplexität der Methode `union` in Abhängigkeit der Länge der Arrays `a1` und `a2`.

3. Implementieren Sie eine **rekursive** Methode

`static int[] sort(int[] array),`

die eine Folge von Integer-Zahlen in sortierter Reihenfolge zurückgibt. Gehen Sie dabei wie folgt vor:

- (a) Unterteilen Sie die Folge in zwei möglichst gleich große Teilfolgen.
 - (b) Sortieren Sie beide Teilfolgen mit Hilfe eines rekursiven Aufrufs der Methode `sort()`.
 - (c) Fügen Sie die beiden sortierten Teilfolgen zusammen mittels der zuvor implementierten Methode `union()`.
4. Bestimmen Sie die Komplexität der Methode `sort` in Abhängigkeit der Länge des Arrays `array`.

Aufgabe 11.2

1. Implementieren Sie ein Interface `GeometricObject`, das die Methoden `double computeArea()` und `double computeCircumference()` besitzt.

2. Vervollständigen Sie die Klasse `Rectangle` so, dass sie ebenfalls das Interface `GeometricObject` implementiert.

```
class Rectangle ... {
    public Rectangle(double length, double width) { ... }

    public double getArea() { ... }
    public double getCircumference() { ... }

    protected double length;
    protected double width;
}
```

3. Implementieren Sie eine Klasse `Square`, die von `Rectangle` abgeleitet ist. Die Seitenlänge des Quadrates soll im Konstruktor gesetzt werden.
4. Könnten die Instanzvariablen `length` und `width` auch als `private` deklariert werden?
5. Implementieren Sie die Klasse `GeometricObjectList`, die mehrere geometrische Objekte verwalten soll. Diese Klasse soll als Instanzvariable eine `ArrayList` `ArrayList<GeometricObject>` besitzen und folgende Methoden zur Verfügung stellen:
- (a) `void add(GeometricObject o)`
fügt ein Objekt in die `ArrayList` ein.
 - (b) `double areaSum()`
berechnet die Summe des Flächeninhalts aller Objekte in der `ArrayList`.
 - (c) `double circumferenceSum()`
berechnet die Summe des Umfangs aller Objekte in der `ArrayList`.