



Aufgabenblatt 2

Aufgabe 1 (10 Punkte)

Wir wollen dreidimensionale Körper im Raum (\mathbb{R}^3) verwalten. Hierzu sind die folgenden Anforderungen gegeben:

- Als Körper kommen Kugeln und Quader in Frage.
 - Jeder Körper befindet sich an einer bestimmten Position $p \in \mathbb{R}^3$ im Raum. Für eine Kugel ist p der Mittelpunkt, für ein Quader der linke, vordere und untere Eckpunkt.
 - Jeder Körper hat eine farbige Oberfläche. Die Farbe der Oberfläche kann geändert werden.
 - Die Größe der dreidimensionalen Körper soll mit einem Faktor α skaliert werden können.
 - Die Körper sollen verschoben werden können.
 - Für jeden Körper soll es möglich sein, das Volumen und die Oberfläche zu berechnen.
 - Es soll bestimmbar sein, ob sich ein Punkt $q \in \mathbb{R}^3$ in einem Körper befindet.
 - Informationen zu einem Körper (Art, Lage, Parameter, Volumen, Oberfläche, Farbe) sollen auf einfache Weise ausgegeben werden können.
- (a) Implementieren Sie in Java eine entsprechende Klassenhierarchie. Verwenden Sie dabei in angemessener Weise Vererbung und abstrakte Klassen. Falls Sie die Formeln für Oberfläche und Volumen der Körper nicht kennen, so recherchieren Sie im WWW.
- (b) Schreiben Sie ein Testprogramm für Ihre Klassenhierarchie.

Aufgabe 2 (Nutzung einer implementierten Schnittstelle, 10 Punkte)

Informieren Sie sich zunächst über die Schnittstelle `CharSequence` im Paket `java.lang`. Diese Schnittstelle stellt ein abstraktes Konzept für den Zugriff auf Zeichenfolgen bereit. Einen Link zur Dokumentation des Java SDK finden Sie auf der Homepage der Veranstaltung.

Erstellen Sie anschließend eine Klasse mit einer Klassenmethode, die folgende Signatur aufweist:

```
public static int zaehleVokale(CharSequence sequence)
```

Die Methode `zaehleVokale()` soll die Anzahl der Vokale innerhalb der Zeichenfolge `sequence` ermitteln.

Schreiben Sie anschließend ein kleines Testprogramm, in dem die Methode `zaehleVokale()` zweimal aufgerufen wird und geben Sie jeweils das Ergebnis aus. Bei den Aufrufen soll die Zeichenfolge (Parameter `sequence`) einmal als `String` und einmal als `StringBuffer` übergeben werden.

Aufgabe 3 (Konstruktoren in einer Klassenhierarchie, 5 Punkte)

Gegeben sei der folgende Quelltext eines Java-Programms:

```
public class RediMix extends Concrete {
    RediMix() { System.out.print("r "); }
    public static void main(String[] args) {
        new RediMix();
    }
}
class Concrete extends Sand {
    Concrete() { System.out.print("c "); }
    private Concrete(String s) { }
}
abstract class Sand {
    Sand() { System.out.print("s "); }
}
```

Welche Ausgabe erzeugt dieses Programm? Erläutern Sie, wie es zu dieser Ausgabe kommt.

Hinweis: Der Quelltext des Programms ist auf der Homepage der Veranstaltung verfügbar.

Aufgabe 4 (Vergleich bei Referenztypen, 5 Punkte)

Gegeben sei der folgende Quelltext eines Java-Programms:

```
public class Twine {
    public static void main(String[] args) {
        String s = "";
        StringBuffer sb1 = new StringBuffer("hi");
        StringBuffer sb2 = new StringBuffer("hi");
        StringBuffer sb3 = new StringBuffer(sb2);
        StringBuffer sb4 = sb3;
        if (sb1.equals(sb2)) s += "1 ";
        if (sb2.equals(sb3)) s += "2 ";
        if (sb3.equals(sb4)) s += "3 ";
        String s2 = "hi";
        String s3 = "hi";
        String s4 = s3;
        if (s2.equals(s3)) s += "4 ";
        if (s3.equals(s4)) s += "5 ";
        System.out.println(s);
    }
}
```

Welche Ausgabe erzeugt dieses Programm? Erläutern Sie, wie es zu dieser Ausgabe kommt.

Hinweis: Der Quelltext des Programms ist auf der Homepage der Veranstaltung verfügbar.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KW 15. Abgabe der Lösungen bis **Sonntag, 14. April 2012, 24 Uhr** via E-Mail an die Adresse `dat-alg13@mail.inf.h-brs.de`.