

Grundlagen der Programmkonstruktion

Übungsblatt 5

1 Rekursion nachvollziehen (3)

Versuchen Sie folgende rekursive Methode nachzuvollziehen. Ermitteln Sie alle Aufrufe von `hanoi(3, 1, 2, 3)`; und die Ausgabe der Methode in der richtigen Reihenfolge (siehe Beispiel unten). Sie sollen in der Lage sein, genau zu erklären, warum wann welcher Aufruf erfolgt!

```
public static void hanoi(int n, int a, int b, int c) {
    if(n == 0) {
        return;
    } else {
        hanoi(n - 1, a, c, b);
        System.out.println("Von " + a + " nach " + c);
        hanoi(n - 1, b, a, c);
    }
}
```

Beispiel

Zur besseren Übersichtlichkeit ist es hilfreich, entsprechend der Rekursionstiefe die Aufrufe und Ausgaben einzurücken.

Die Lösung für `hanoi(1, 1, 2, 3)`; sieht folgendermaßen aus:

```
hanoi(1, 1, 2, 3);
    hanoi(0, 1, 3, 2);
        "Von 1 nach 3"
    hanoi(0, 2, 1, 3);
```

Die ersten drei Zeilen Ihrer Lösung sollten folgendermaßen aussehen:

```
hanoi(3, 1, 2, 3);
    hanoi(2, 1, 3, 2);
        hanoi(1, 1, 2, 3);
```

Bonusaufgabe

Wieviele Aufrufe der Methode `hanoi` gibt es bei einem Aufruf von `hanoi(10, 1, 2, 3)`, `hanoi(20, 1, 2, 3)` und `hanoi(n, 1, 2, 3)` wobei n ein beliebiger Integer ist? Begründen Sie ihre Entscheidung!

2 Objekte: Punkte und Linien (4)

In dieser Aufgabe müssen Sie mehrere Objekte erstellen. Beachten Sie, dass Objektvariablen immer *private* sein müssen, und benutzen Sie *keine* Getter und Setter. Vermeiden Sie bitte langen und unnötigen Programmcode.

2.1 Klasse Punkt (2)

Erstellen Sie eine Klasse `Point`. Diese repräsentiert Punkte auf einer Ebene. Ein Punkt hat somit zwei Objektvariablen `x` und `y`, diese sollen in unserem Fall vom Typ `double` sein.

- Erstellen Sie die Klasse mit allen nötigen Objektvariablen und *einem geeigneten* Konstruktor.
- Ergänzen Sie die Klasse um eine Methode `dist(Point p)`. Diese soll den Abstand zwischen dem aktuellen Objekt und dem Objekt im Parameter zurückliefern. Mathematische Berechnungen wie die Quadratwurzel finden Sie in der Java-Klasse `Math`.
- Ergänzen Sie die Klasse um eine Methode `middle(Point p)`. Diese soll einen neuen Punkt zurückliefern, der genau in der Mitte zwischen dem aktuellen Objekt und dem Objekt im Parameter liegt.

2.2 Klasse Dreieck (2)

Erstellen Sie eine Klasse `Triangle`. Diese repräsentiert Dreiecke.

- Finden Sie selbstständig geeignete Objektvariablen und erstellen Sie einen entsprechenden Konstruktor. Beachten Sie, dass Zwischenergebnisse von Berechnungen nicht in Objektvariablen gespeichert werden dürfen.
- Erstellen Sie eine Methode `perimeter()`. Diese soll den Umfang des Dreiecks zurückliefern.
- Erstellen Sie eine Methode `midTriangle()`. Diese soll das Dreieck zurückliefern, dass sie erhalten, wenn sie die Mittelpunkte der Seiten als Eckpunkte verbinden.
- (Bonusaufgabe) Erstellen Sie eine Methode `area()` die den Flächeninhalt des Dreiecks zurückliefert.