

Grundlagen der Programmkonstruktion

Übungsblatt 1 (zu lösen bis 6./7./8. Mai 2013)

Bevor Sie die Aufgaben lösen, lesen Sie bitte die folgenden Hinweise. Sollten Sie diese Hinweise in ihrer Lösung nicht beachten, so wird ihre Aufgabe als falsch gewertet und Ihnen entsprechend Punkte abgezogen.

Hinweise

- Beachten Sie, dass Sie die richtigen Methoden mit den richtigen Parametertypen und den richtigen Rückgabetypen verwenden.
- Sollten Sie eine der Methoden mit dem Schlüsselwort `static` implementieren, so ist diese falsch.
- Beachten Sie, dass die Methoden `toString()` und `equals(Object)` in der Klasse `Object` vordefiniert sind. Lesen Sie daher zu diesen Methoden die Dokumentation. Sie sollen in der Lage sein, die wesentlichen Aspekte dieser Methoden zu erklären.
- Sie dürfen in den Klassen weitere Methoden erstellen und verwenden.
- Achten Sie darauf, dass Sie lokale Variablen nicht als Objektvariablen anlegen.
- Achten Sie auf den Programmierstil. Schreiben Sie nicht zu kompakt (80 Zeichen pro Zeile sind genug, Leerzeilen machen ein Programm nicht langsamer), benutzen Sie aber auch nicht unnötig viele lokale Variablen.
- Achten Sie darauf, insbesondere wenn Sie mit Kollegen zusammenarbeiten, dass Sie sich nicht aussuchen können, welche Aufgabe Sie präsentieren.

1 Klasse Punkt (1.3 Punkte)

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

1. Erstellen Sie die Klasse mit allen nötigen Objektvariablen und einem *geeigneten* Konstruktor.
2. 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`.

3. 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.
4. Erstellen Sie die Methode `toString()`. Sie können dabei selbst das Ausgabeformat wählen.
5. Erstellen Sie eine Methode `equals(Object o)`, mit der Sie `Point` mit einem anderen Punkt vergleichen. Sie können sich aussuchen, ob Sie `equals` mit `instanceof` oder `getClass()` implementieren, achten Sie aber darauf, dass `equals(null)` immer `false` zurückliefern muss.

2 Klasse Dreieck (2.3 Punkte)

Erstellen Sie mit Hilfe der Klasse `Point` eine Klasse `Triangle`. Diese repräsentiert Dreiecke.

1. Finden Sie selbstständig geeignete Objektvariablen und erstellen Sie einen entsprechenden Konstruktor. Beachten Sie, dass Zwischenergebnisse von Berechnungen nicht in Objektvariablen gespeichert werden dürfen.
2. Erstellen Sie eine Methode `perimeter()`. Diese soll den Umfang des Dreiecks zurückliefern.
3. Erstellen Sie eine Methode `midTriangle()`. Diese soll das Dreieck zurückliefern, das sie erhalten, wenn sie die Mittelpunkte der Seiten als Eckpunkte verbinden.
4. Erstellen Sie eine Methode `area()` die den Flächeninhalt des Dreiecks zurückliefert.
5. Erstellen Sie die Methode `toString()`.
6. Erstellen Sie eine Methode `equals(Object o)`, mit der Sie `Triangle` mit einem anderen Dreieck vergleichen.

3 Testprogramm (0.5 Punkte)

Erstellen Sie ein kleines Testprogramm. Sie sollen darin jede der oben erstellten Methoden aufrufen (das darf auch indirekt geschehen) und dadurch diese testen.