Grundlagen der Programmkonstruktion
Übungsblatt 6

1 Object (4.5)

In dieser Aufgabe sollen Sie die Klassen der vorigen Übung um die Methoden `String toString()` und `boolean equals(Object)` erweitern. Diese Methoden werden inhaltlich in der nächsten Vorlesung besprochen. Probieren Sie trotzdem schon vorher, diese Aufgabe zu lösen.

1.1 Point (1.5)

1.1.1 `String toString()` (0.5)

Ergänzen Sie die Klasse `Point` aus der vorigen Übung um die Methode `String toString()`. Diese Methode soll eine String-Repräsentation des Punktes in der Form "(*x-Koordinate*, *y-Koordinate*)" zurückliefern.

1.1.2 `boolean equals(Object)` (1)

Ergänzen Sie die Klasse `Point` um die Methode `boolean equals(Object)`. Diese Methode soll `true` zurückliefern, wenn das Objekt im Parameter eine Instanz von `Point` mit den selben Koordinaten wie das aktuelle Objekt enthält.

Hinweise:

- Lesen Sie sorgfältig den Eintrag zu `equals` in der Java-API (`http://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#equals(java.lang.Object)`) und behandeln Sie die dort angeführten Sonderfälle (`null`!) wie dort angegeben.
- Der Parameter in der Methode `boolean equals(Object)` ist vom Typ `Object` und nicht `Point`.
- Die Methode darf *keine* NullPointerException oder ClassCastException werfen, unabhängig davon, welcher Parameter übergeben wird.
- Die Methode `boolean equals(Object)` darf *keine* Objektvariable ändern (auch dann nicht, wenn deren Inhalt am Ende wieder hergestellt wird).
- Das Verwenden von `toString()` um die Objekte zu vergleichen ist *nicht* (!) erlaubt.

1.2 Triangle (3)

Sie dürfen im Folgenden annehmen, dass die Klasse `Point` die beiden Methoden `String toString()` und `boolean equals(Object)` enthält.

1.2.1 `String toString()` (1)

Ergänzen Sie die Klasse `Triangle` aus der vorigen Übung um die Methode `String toString()`. Wählen Sie selbst ein geeignetes Format für `toString()`.

1.2.2 `boolean equals(Object)` (2)

Ergänzen Sie die Klasse `Triangle` um die Methode `boolean equals(Object)`. Beachten Sie dabei die Hinweise oben zur `equals`-Methode. Sie können eine der folgenden Varianten auswählen um festzustellen, ob zwei Dreiecke gleich sind:

Leichte Variante: Zwei Instanzen von `Triangle` sind gleich wenn der erste Eckpunkt des ersten Dreiecks gleich dem ersten Eckpunkt des zweiten Dreiecks ist, der zweite Eckpunkt gleich ist und auch der dritte übereinstimmt.

Schwere Variante - Bonusaufgabe: Zwei Instanzen von `Triangle` sind gleich wenn die Dreiecke deckungsgleich sind.

2 Interfaces (3)

2.1 Verschiebbar: Point(0.5)

Implementieren Sie das Interface `Verschiebbar` aus der Vorlesung in `Point`.

2.2 Verschiebbar: Triangle (1)

Implementieren Sie das Interface `Verschiebbar` in der Klasse `Triangle`. Sie dürfen dafür annehmen, dass `Point` das Interface `Verschiebbar` korrekt implementiert.

2.3 Spiegelbar (1.5)

2.3.1 Interface erstellen (0.5)

Erstellen Sie ein Interface `Spiegelbar` mit zwei neuen Methoden:

- Eine Methode soll das Objekt horizontal spiegeln (d.h. an der y-Achse).
- Eine Methode soll das Objekt vertikal spiegeln (d.h. an der x-Achse).

Sie können selbst entscheiden, ob die Methoden das bestehende Objekt manipuliert oder ein neues gespiegeltes Objekt erstellen und zurückliefert. Wählen Sie daher selbst einen geeigneten Namen, Parameter und Rückgabetyp der Methoden.

2.3.2 Spiegelbar: Point (0.5)

Implementieren Sie das Interface `Spiegelbar` in der Klasse `Point`.

2.3.3 Spiegelbar: Triangle (0.5)

Implementieren Sie das Interface `Spiegelbar` in der Klasse `Triangle`. Sie dürfen dafür annehmen, dass `Point` das Interface `Spiegelbar` korrekt implementiert.