Grundlagen der Programmkonstruktion
Übungsblatt 2 (zu lösen bis 13./14./15. Mai 2013)

1 Interfaces (0.9 Punkt)

1. Erstellen Sie ein Interface `Shape`. Dieses Interface soll eine Methode `double perimeter()` und eine Methode `double area()` implementieren.

2. Erstellen Sie ein Interface `Scaleable`. Dieses Interface soll eine Methode `void scale(double sx, double sy)` enthalten. Objekte, die dieses Interface implementieren, sollen relativ zum Ursprung um den Faktor `sx` (für die X-Koordinate) bzw. `sy` (für die Y-Koordinate) skaliert werden (d.h. die X-Koordinate wird mit `sx` multipliziert und die Y-Koordinate mit `sy`).

3. Erstellen Sie ein Interface `Moveable`. Dieses Interface soll eine Methode `void move(double tx, double ty)` enthalten. Objekte, die dieses Interface implementieren, sollen beim Aufruf der Methode um den Vektor (tx, ty) verschoben werden.

4. Erstellen Sie ein Interface `Rotate`. Dieses Interface soll eine Methode `void rotate(double alpha)` enthalten. Objekte, die dieses Interface implementieren, sollen um den Winkel *alpha* (sie knnen das Winkelmaß selbst wählen) rotiert werden (relativ zum Ursprung).

2 Implementieren des Interfaces (2 Punkte)

1. Implementieren Sie die Interfaces in der Klasse `Triangle` vom vorigen Übungsblatt. Sie dürfen (und sollten) beliebige Methoden den Klassen `Point` und `Triangle` hinzufügen. Das Verwenden von Klassen, die bereits in Java vordefiniert sind (wie z.B. Klassen aus `java.awt`) ist dabei mit Ausnahme von der Klasse `Math` nicht erlaubt.

2. Erstellen Sie eine Klasse `Circle` die alle Interfaces aus Punkt 1 implementiert.

3 Interface Transformable (1.3 Punkte)

1. Erstellen Sie ein Interface `Transformable` mit den Methoden `Transformable rotate2(double alpha)`, `Transformable move2(double tx, double ty)`, `Transformable scale2(double sx, sy)` enthalten. Die Definition ist dabei fast identisch zu Punkt 1, allerdings sollen diese Methoden das Objekt nicht manipulieren, sondern müssen ein neues Objekt zurückliefern, das das originale Objekt entsprechend transformiert darstellt.

2. Implementieren Sie dieses Interface in den Klassen `Triangle` und `Circle`. Sie dürfen beliebige Methoden in `Point` erstellen und auch auf die in den vorigen Aufgaben definierten Methoden zugreifen.

3. Schreiben Sie ein kleines Testprogramm, aus dem ersichtlich wird, dass `Transformable` nicht das Objekt ändert im Gegensatz zu den anderen Interfaces.