

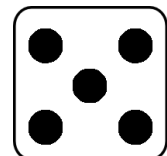
Aufgabe „exercise02“ (Abgabe vom 5.4. - 11.04.2016, bZv-relevant)

Im Verzeichnis Übungsaufgabe_2 finden Sie neben der Aufgabenstellung

- die Datei `Point.java`
Dies ist zum einen die Musterlösung der Aufgabe „exercise01“, zum anderen der Ausgangspunkt für die neue Aufgabe „exercise02“. Bitte vergleichen Sie diese Datei mit Ihrer Lösung der letzten Woche. Belassen Sie aber unbedingt eine **unveränderte Version**, andernfalls wird die Lösung Ihrer neuen Aufgabe schwierig.
- Die Datei `Polygon.java`
Dies ist die Spezifikation eines Abstrakten Datentyps in Form einer Java Interface Definition. Auch diese Datei lassen Sie bitte **unverändert**; sie stellt die Vorgabe für Ihre Implementierungsaufgabe dar.

Ein Polygon ist eine zweidimensionale geometrische Figur, die sich durch einen Linienzug durch mehrere Punkte ergibt. Bekannte Beispiele für Polygone sind Dreiecke, Vierecke, Fünfecke etc. Wichtig für die eindeutige Festlegung eines Polygons sind somit einerseits die Punkte, durch die der Linienzug führen soll, und andererseits die Reihenfolge, mit der die Punkte angefahren werden.

Dass die Reihenfolge wichtig für das Aussehen eines Polygons ist, wird deutlich, wenn Sie sich z.B. überlegen, welche unterschiedlichen Formen beim Verbinden einer Fünf auf einem Würfel möglich sind.

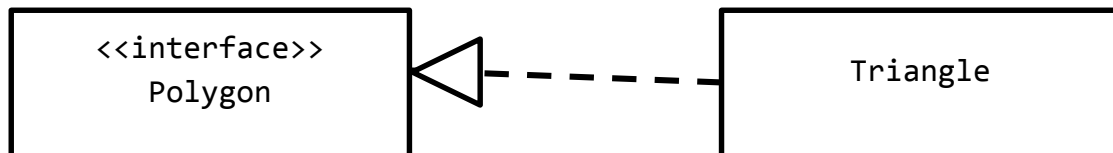


Der Abstrakte Datentyp Polygon soll nun folgende Methoden besitzen:

- `public Point[] allVertexes();`
Diese Methode soll alle Eckpunkte des Polygons zurückliefern
- `public double perimeter();`
Diese Methode soll den Umfang des Polygons zurückliefern
- `public double area();`
Diese Methode soll die Fläche des Polygons zurückliefern
- `public void move(double dx, double dy);`
Diese Methode soll das Polygon im Koordinatensystem verschieben

Ihre Aufgabe:

Implementieren Sie eine Klasse `Triangle`, die diesen Abstrakten Datentyp für den Spezialfall eines Dreiecks (engl. Triangle) erfüllt. Schreiben Sie hierzu eine neue Java-Klasse welche das Interface `Polygon` implementiert. Stellen Sie dabei sicher, dass folgende in der Vorlesung eingeführte Abhängigkeitsbeziehung erfüllt ist:



Neben den vier durch das Interface vorgegebenen Methoden soll die Klasse auch einen Konstruktor erhalten, dem die drei Eckpunkte des Dreiecks als Parameter übergeben werden:

```
Triangle(Point p1, Point p2, Point p3)
{
    ...
}
```

Hinweise:

1. Implementieren Sie in der Klasse `Triangle` eine `main`-Methode, die das Verhalten Ihrer Lösung überprüft. Orientieren Sie sich hierbei ggf. an der `main`-Methode der vorgegebenen `Point`-Klasse aus der Übung der letzten Woche. Beispielausgabe:

```
Punkt1: (0,00;0,00)
Punkt2: (1,00;0,00)
Punkt3: (0,00;1,00)
Umfang: 3.414213562373095 Erwartet: 3.414213562373095
Flaeche: 0.49999999999999983 Erwartet: 0.49999999999999983
Verschiebe den Punkt um die Distanz: +2.0, -2.0
Punkt1: (2,00;-2,00)
Punkt2: (3,00;-2,00)
Punkt3: (2,00;-1,00)
```

2. Für die Berechnung des Flächeninhalts eines Dreiecks bei gegebenen Eckpunkten ist der **Satz des Heron** hilfreich:

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

Dabei sind a , b und c die Längen der Dreiecksseiten und s der halbe Umfang, also:

$$s = \frac{1}{2}(a + b + c)$$