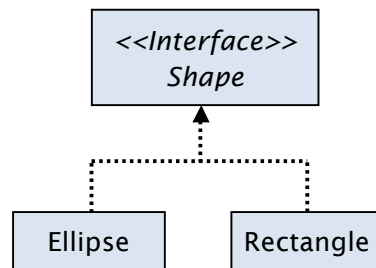


# Aufgabenblatt 1

## Teil 1)

Um das Arbeiten mit geometrischen Formen (Shapes) zu unterstützen, soll das Interface *Shape* und die beiden Klassen *Ellipse* und *Rectangle* realisiert werden.



### Interface Shape:

Das Interface *Shape* definiert folgende Methoden:

- `setCenter(Point)`                      setzt den Flächenschwerpunkt
- `Point getCenter()`                      liefert den Flächenschwerpunkt zurück
- `setWidth(double)`                      setzt die Breite
- `double getWidth()`                      liefert die Breite zurück
- `setHeight(double)`                      setzt die Höhe
- `double getHeight()`                      liefert die Höhe zurück
- `double getArea()`                      liefert den Flächeninhalt zurück

### Klasse Point:

Definieren Sie eine mutable Klasse *Point* für Punkte in der Ebene. Implementieren Sie die Methoden `setX(double)`, `getX()`, `setY(double)`, `getY()` und `toString()`. Sehen Sie geeignete Konstruktoren vor.

### Klassen Ellipse und Rectangle:

Die Klassen *Ellipse* und *Rectangle* sollen beide das Interface *Shape* implementieren. Beachten Sie, dass Höhe und Breite unabhängig geändert werden können. Man hat daher Rechtecke (keine Quadrate) und Ellipsen (keine Kreise). Die Flächen sind mit Hilfe geeigneter Formeln zu berechnen. Sehen Sie Konstruktoren und `toString()`-Methoden vor.

### Klasse ShapeApplication:

Schreiben Sie eine Klasse *ShapeApplication* mit einer Main-Methode, welche die Klassen ausgiebig testet. Achten Sie auf richtige Kapselung Ihrer Klassen. Wie verhält sich Ihr Programm wenn folgende Anweisungsfolge abläuft:

```
Point p = new Point(1,2);  
Shape s1 = new Ellipse(p);  
Shape s2 = new Rect(p);  
System.out.println(s1 + ", " + s2);  
p.setX(10);  
// s1 und s2 sollten unverändert bleiben:  
System.out.println(s1 + ", " + s2);
```

Schreiben Sie außerdem eine statische Methode

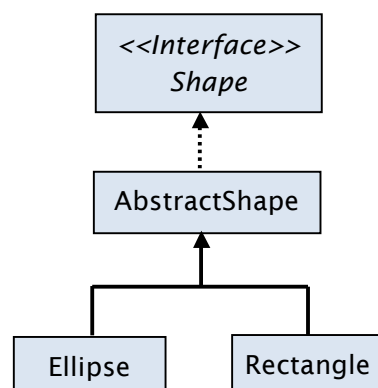
```
static double totalArea(Shape[])
```

die für ein Feld von *Shapes* die Summe der Flächen zurückliefert.

## Teil 2)

### Abstrakte Klasse *AbstractShape*:

Einige Methoden der Klasse *Rectangle* und der Klasse *Ellipse* besitzen die gleiche Implementierung. Um den Aufwand der mehrfachen Implementierung zu vermeiden, soll eine abstrakte Klasse realisiert werden, wo gemeinsame Teile, wie Flächenschwerpunkt, Höhe und Breite und die dazugehörigen Methoden implementiert werden. Die Methode *getArea()* ist selbstverständlich abstrakt.



### Abgabe:

- Es sind geeignete Testaufrufe vorzusehen, die die Funktionsfähigkeit Ihres Programms zeigen.
- Bei der Abgabe soll der erstellte Code genau erklärt werden können. Außerdem sollte ein Programmablauf mit dem Debugger durchlaufen werden können.