

Pflichtaufgabe 1 - Kontaktliste (Array und Testen)

Dies ist Ihre erste Pflichtaufgabe, die in Moodle *fristgerecht* abgegeben werden muss und *bewertet* wird. Bitte laden Sie als Lösung nur java-Files (+ Ausnahme einer txt-Datei) hoch. Bitte beachten Sie, dass Abgaben, die **mehr als 10 Min.** zu spät hochgeladen werden, leider **nicht berücksichtigt** werden. Mit dem Anklicken der **Eigenständigkeitserklärung** bestätigen Sie, dass die Lösungen Ihre eigene (Gruppen-)Leistung sind. Bitte beachten Sie, dass (auch teilweise) identische Lösungen von Personen aus fremden Gruppen als **Betrugsversuch** ("Abschreiben") angesehen und mit **0 Punkten** bewertet werden. Bitte geben Sie mit @author in allen Klassen einen Autor an, bzw. beide Autoren.

Vererbung von Klassen

Lernziel: Eigene Klassen programmieren, die von einer Klasse abgeleitet werden und die auf Attribute und Methoden der Superklasse zugreifen.

Betrachten wir Klassen und deren Objekte aus dem Bereich der Geometrie (zweidimensional):

1.1 Shape (6 Punkte)

Implementieren Sie eine Klasse mit den Parametern und Methoden. Beachten Sie passende die passenden Zugriffsmodifikatoren. Gegeben sind nur folgende Kommentare aus der Dokumentation.

```
/**
 * Base class for geometrical shapes. Each shape has an explicit origin
 * represented by a point: (x,y).
 * @ version 1.x
 * @author X.Y.
 */
XXXXX

/** x-coordinate of the origin */
XXXXX

/** y-coordinate of the origin */
XXXXX

/**
 * calculate the area of a shape. the base shape is just point so it has no
 * area.
 * @return the area of the shape
 */
XXXXX

/**
 * calculate the circumference of a shape. the base shape is just point so
 * it has no circumference.
 * @return the circumference of the shape
 */
XXXXX

/**
 * constructs a textual representation of the origin in the form: "(x, y)".
```

```

    * @return the origin on textual representation
    */
XXXXX

/**
 * constructs a textual representation of the object.
 * @return the current object in a textual representation
 */
XXXXX

```

2.2 Rectangle, Square und Circle (3+3+3 Punkte)

Programmieren Sie drei weitere Klassen Rectangle, Square und Circle, die von der Klasse Shape direkt oder indirekt abgeleitet werden. Informationen finden Sie dazu auch in der Dokumentation.

Folgende Aufgaben sollen die Klassen erfüllen:

Rectangle.java:	<ul style="list-style-type: none"> • Subklasse von Shape. • erbt double-Werte xOrigin und yOrigin aus der Klasse Shape und speichert zusätzlich zwei double-Werte xDelta und yDelta, die die Hälfte einer Seitenlänge des Rechteckes in x- oder y-Richtung darstellen. • Initialisierung erfolgt im Konstruktor der Klasse Rectangle, der den Konstruktor der Superklasse aufruft. • die Klasse enthält eine circumference-Methode, die den Umfang des Rechteckes bestimmen soll. • die Klasse enthält eine area-Methode, welche die Fläche des Rechteckes berechnen soll. • die Klasse enthält eine toString-Methode, welche als Zeichenkette die Art des Objektes (hier Rectangle), den Ursprung und die Werte für xDelta und yDelta liefert.
Square.java:	<ul style="list-style-type: none"> • Subklasse von Rectangle. Ein Square ist ein Rectangle, dessen Seitenlängen in x- oder y-Richtung den gleichen Wert haben. • Initialisierung erfolgt im Konstruktor der Klasse Square, der den Konstruktor der Superklasse aufruft. • die Klasse erbt die beiden Methoden circumference und area, die Umfang und Fläche des Quadrates berechnen sollen. • die Klasse enthält eine toString-Methode, welche als Zeichenkette die Art des Objektes (hier Square), den Ursprung und den Wert für delta liefert.
Circle.java:	<ul style="list-style-type: none"> • Unterklasse von Shape. • erbt double-Werte xOrigin und yOrigin aus der Klasse Shape und speichert zusätzlich einen double-Wert radius. radius = Radius des Kreises. • Initialisierung erfolgt im Konstruktor der Klasse Circle, der den Konstruktor der Superklasse aufruft. • die Klasse enthält außerdem noch die beiden Methoden: circumference und area, der Umfang und Fläche des Kreises berechnen sollen. • die Klasse enthält eine toString-Methode, welche als Zeichenkette die Art des Objektes (hier Circle), den Ursprung und den Wert für radius liefert.

2.3 ShapeTester (3)

Programmieren Sie eine Testklasse ShapeTester, in der jeweils ein Objekt der oben beschriebenen Klassen Shape, Rectangle, Square und Circle erzeugt wird. In den einzelnen Klassen sollen die in Tabelle 1 hinterlegten Werte initialisiert werden:

Shape.java	x=10	y=40		
Rectangle.java	x=20	y=30	xDelta: 35	yDelta:15
Square.java	x=30	y=20	delta: 5	
Circle.java	x=40	y=10	radius: 25	

Der Umfang des Quadrates, die Fläche des Rechtecks sowie Umfang und Fläche des Kreises sollen berechnet und in der Eingabeaufforderung bzw. im Terminal ausgegeben werden.

Shape with origin: (10.0, 40.0) / Area: 0.0 / Circumference: 0.0

Rectangle with origin: (20.0, 30.0), xDelta: 35.0, yDelta: 15.0 / Area: 2100.0 / Circumference: 200.0

Square with origin: (30.0, 20.0) and delta: 5.0 / Area: 100.0 / Circumference: 40.0

Circle with origin: (40.0, 10.0) and radius: 25.0 / Area: 1963.5 / Circumference: 157.1

2.4 Reporter (2 Punkte)

Programmieren sie dazu eine Hilfsklasse Reporter, in der sich die Methode report mit der folgenden Signatur befindet: public static void report(Shape s). Führen Sie diese Methode mit einem oder mehrer ShapeObjekte aus.

Abgabe:

Quelldateien in Moodle hochladen:

- Rectangle.java
- Square.java
- Circle.java
- ShapeTester.java
- Reporter.java
- Ausgabe der Konsole als txt-Datei

Achten Sie bei der Implementierung aller Klassen auf einen guten Programmierstil (Effizienz bei der Vererbung, Kommentare, Formatierung, Namenskonventionen, eigener parameterloser Konstruktor usw.).

(Gesamt 20 Punkte)