

Übungsaufgaben 5

5.1 Vererbung und Polymorphie

- a) Ausgehend von der *korrigierten* Klassenhierarchie aus Abb. 1 implementieren Sie in der Klasse `Quadrat` eine neue Methode (z.B. `berechneDiagonale()`), die in `Rechteck` nicht vorhanden ist (d.h. im `Quadrat` nicht überschrieben wird).
- b) Erzeugen Sie in der Testklasse ein `Quadrat` (Referenz `q`). Definieren Sie eine `Rechteck`-Referenz `r` und weisen Sie ihr eine `Quadrat`-Referenz `q` zu. Was passiert an dieser Stelle? Welche Ausgaben erwarten Sie bei folgenden Aufrufen in der Testklasse:
 - (1) `System.out.println(r);`
 - (2) `System.out.println(r.berechneFlaeche());`
 - (3) `System.out.println(r.berechneDiagonale());`
 Überprüfen Sie die Richtigkeit Ihrer Vermutung durch Ausführung des Programms.
- c) Definieren Sie in der Testklasse eine weitere `Quadrat`-Referenz `q1` und weisen ihr die `Rechteck`-Referenz `r` zu. Was passiert an dieser Stelle? Welche Ausgaben erwarten Sie bei folgenden Aufrufen in der Testklasse:
 - (1) `System.out.println(q1);`
 - (2) `System.out.println(q1.berechneFlaeche());`
 - (3) `System.out.println(q1.berechneDiagonale());`

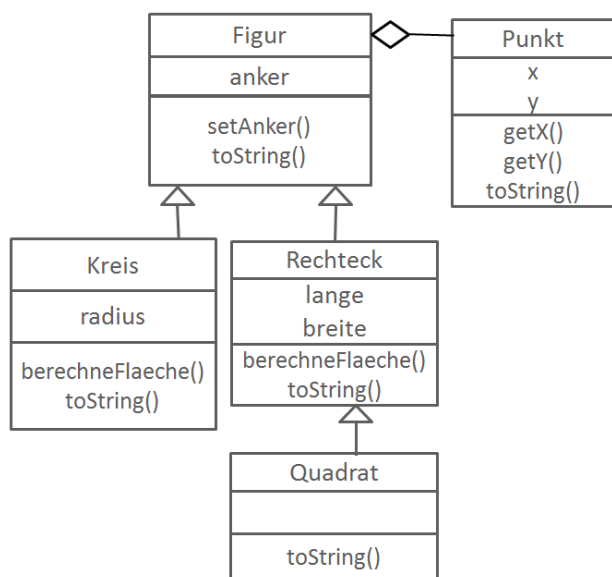


Abb. 1: Klassenhierarchie

5.2 Polymorphe Argumente

Implementieren Sie in der Klasse `Figur` eine Instanz-Methode

```
public int berechneXAbstand(Figur f),
```

welche einen X-Abstand (d.h. den Abstand zwischen den X-Koordinaten des Ankers) zwischen der Figuren `this` und `f` berechnet.

Berechnen Sie in der Testklasse mithilfe dieser Methode den X-Abstand zwischen zwei *Kreisen* und zwischen einem *Quadrat* und einem *Rechteck*, die Sie zuvor erzeugen.

5.3 Casting

Betrachten Sie das folgende Programm. Welches in der Klasse `Cast` durchgeführte Casting ist zulässig? Handelt es sich dabei um ein implizites oder explizites Casting?

```
// Datei: Cast.java
class A {
    private int x = 1;
}

class B extends A {
    private int y = 2;
}

class C extends B {
    private int z = 3;
}

class D {
    private float f = 2.0f;
}

public class Cast {
    public static void main (String[] args) {
        B b = new C();
        A a = b;
        Object o = b;
        C c = (C) b;
        C c2 = (C) o;
        D d = (D) b;
    }
}
```