

Aufgabenblatt 13

Bearbeitungsende: 15.01.2023

Aufgabe 1 [Programmierung – nicht bewertet]

Stellen Sie folgende Implementierungen der Schnittstelle **Funktion** (siehe Vorlesungsfolien und den dazu bereitgestellten Code) bereit:

- **Wurzel** beschreibt die Funktion $f(x) = \sqrt{x}$ (dargestellt als `"sqrt(x)"` – Verwendung von `Math.sqrt`).
- **Abs** beschreibt die Funktion $f(x) = |x|$ (dargestellt als `"|x|"`).
- **RezWurzel** beschreibt die Funktion $f(x) = \frac{1}{\sqrt{|x|}}$. Implementieren Sie diese mit Hilfe der vorhergehenden Funktionen sowie der bereitgestellten **Kettenfunktion**.

Aufgabe 2 [Programmierung – nicht bewertet]

Erweitern Sie das UML-Klassendiagramm aus entsprechenden früheren Aufgaben um die in dieser Aufgabe zu erstellenden Typen, bevor Sie **Java**-Code entwickeln.

Verwenden Sie im Folgenden die aus der Vorlesung bekannte Klasse **Punkt2D**, die einen 2D-Punkt modelliert.

Ordnen Sie Schnittstellen

- (a) **Figur**
- (b) **Viereck**
- (c) **Kreis**
- (d) **RechteckXY** (achsparalleles Rechteck)
- (e) **QuadratXY** (achsparalleles Quadrat)
- (f) **Dreieck**
- (g) **Polygon**
- (h) **Quadrat**
- (i) **Rechteck**

in einer sinnvollen Hierarchie an.

- (a) **Figur** bietet die Methoden der Klasse **Figur** aus Aufgabe 4 von Blatt 12 an.
- (b) **Kreis** bietet die Methode `radius` der Klasse **KreisMitMitteRadius** an.

- (c) `Polygon` bietet eine Methode `ecke` an, die einen ganzzahligen Wert annimmt und die entsprechende Ecke des Polygons zurückgibt, von einer beliebigen, aber stets gleichen (für einzelne Polygontypen ggf. genauer spezifizierten) Startecke aus gegen den Uhrzeigersinn gezählt, beginnend mit 0.
- (d) `Rechteck` bietet die Methoden `breite` und `hoehe` der Klasse `RechteckXYPerMitteLaengen` an.
- (e) `RechteckXY` bietet die Methode `ecke` der Klasse `RechteckXYPerMitteLaengen` an (die zwei `boolean`-Argumente annimmt). Außerdem gilt ab hier für die Methode `ecke` von `Polygon` (die nur ein `int`-Argument annimmt), dass die Zählung der Ecken bei der linken unteren Ecke beginnt.

Fügen Sie dann die Klassen `FigurPerMitte`, `RechteckXYPerMitteLaengen` und `KreisPerMitteRadius` aus Aufgabe 4 und 5 von Blatt 12 in die Hierarchie ein.

Aufgabe 3 [Programmierung – nicht bewertet]

`FigurCompFlaeche` implementiert `Figur` aus Aufgabe 4 von Blatt 12, indem jedes Objekt eine Referenz auf ein anderes `Figur`-Objekt enthält und dessen Eigenschaften wiedergibt. Alle Methoden sind also als entsprechende Aufrufe an diese Referenz zu implementieren. Die Referenz auf das andere `Figur`-Objekt wird dem (einzigen) Konstruktor übergeben.

Die Klasse soll zudem die Schnittstelle `Comparable` implementieren (siehe Vorlesung). Dazu sollen Instanzen der Klasse anhand ihrer Flächen verglichen werden.

Es soll damit z.B. möglich sein, die in der Vorlesung präsentierte Methode

```
ArrayUtil.min(Comparable[] a)
```

(Code in LEA verfügbar) auf ein Feld von solchen Figuren anzuwenden.

Aufgabe 4 [Programmierung – nicht bewertet]

Hinweis: Ergänzen Sie für diese Aufgabe zuerst das UML-Diagramm aus Aufgabe 4 von Blatt 12 entsprechend.

Integrieren Sie die Klasse `RechteckXYPerEcke` aus B08A3 mit allen Konstruktoren in die Struktur aus Aufgabe 2. Die Methoden `schnitt` und `istEnthalten(Rechteck r)` können, müssen aber nicht übernommen werden.

Ergänzen Sie dann die Schnittstelle `Figur` um die Methoden

- `huelle()`
- `huelle(Punkt p)`
- `huelle(Figur f)`

die eine Instanz vom Referenztyp `RechteckXY` zurückgeben. Diese stellt das kleinste (achsparallele) Rechteck dar, das die durch die Instanz gegebene Figur sowie den/die als Argument gegebene/n Punkt/Figur umschließt.

Hinweise:

- Verwenden Sie für deren Implementierung die `huelle`-Methoden von `RechteckXY`. Nur `huelle()` muss an den passenden Stellen (wo) explizit implementiert werden.
- Beachten Sie zu `equals` für `RechteckXYMitMitteLaengen` und `RechteckXYMitKoord`, dass Instanzen beider Klassen gleich sein können, wenn sie das mathematisch selbe Rechteck beschreiben.

Lösungen zu mit [**Programmierung**] markierten Aufgaben sind im **Praktomat** einzureichen.

Lösungen zu mit [**Programmierung – nicht bewertet**] markierten Aufgaben können ebenfalls im **Praktomat** eingereicht werden, werden jedoch nicht bewertet.

Allgemeine **Fragen** zu den Aufgaben können Sie im **LEA-Forum** „Übungsaufgaben“ stellen.

Hilfe bei der Lösung der Aufgaben erhalten Sie in den **Übungen** und in der **Studierwerkstatt** .