Westfälische Hochschule

Fachbereich Informatik und Kommunikation
Prof. Dr. Marcel Luis

EPR: Praktische Aufgabe „Rationale Zahl“

Diese Aufgabe stellt im Praktikum den Einstieg in die *objektorientierte Programmierung* dar. Es geht um positive rationale Zahlen, die als Bruch mit natürlichem Zähler und Nenner darstellbar sind.

Das Hauptziel dieser Aufgabe ist es, die Realisierung und Anwendung von Konstruktoren und Instanzmethoden zu üben, nicht die Feinheiten der Bruchrechnung. Falls also die letzten beiden Testfälle (s. u.) bei Ihnen nicht das gewünschte Ergebnis zeigen und sie keine Idee mehr haben, woran es liegen könnte, ist das nicht weiter schlimm.

Realisieren Sie im Paket `rationalezahl` eine Klasse `RationaleZahl`, deren Objekte rationale Zahlen repräsentieren und die folgende Methoden enthält:

- Einen Konstruktor `RationaleZahl(long)`, durch den ein Objekt erzeugt wird, das die übergebene ganze Zahl repräsentiert (Auch ganze Zahlen sind rational!). Sie dürfen davon ausgehen, dass der Konstruktor nur mit aktuellem Parameter größer als 0 aufgerufen wird.
- Einen Konstruktor `RationaleZahl(long, long)`, durch den ein Objekt erzeugt wird, das den Quotienten aus Zähler (erster Parameter) und Nenner (zweiter Parameter) repräsentiert. Sie dürfen davon ausgehen, dass der Konstruktor nur mit aktuellen Parametern größer als 0 aufgerufen wird.
- Eine Instanzmethode `RationaleZahl addiere(RationaleZahl)` zur Addition dieser und der übergebenen Zahl. Die Summe beider Zahlen soll als Ergebnis der Methode zurückgegeben werden. Das Objekt, das die Methode ausführt, und das übergebene Objekt sollen unverändert bleiben.
- Eine Instanzmethode `String gibAlsText()`, die eine textuelle Darstellung dieser rationalen Zahl liefert. Die Darstellung soll im Format (Beispiel) `13/12` oder `4` erfolgen, abhängig davon, ob die rationale Zahl ganzzahlig oder ein echter Bruch ist. Die Zahl soll in ihrer maximal gekürzten Form dargestellt werden (s. Testablauf).

Realisieren Sie im gleichen Paket außerdem eine Klasse `RationaleZahlTest` zum Test der Methoden der Klasse `RationaleZahl`. Realisieren Sie darin eine Methode `main` mit folgendem Testablauf. Die Schreibweise $\frac{n}{m}$ meint dabei ein Objekt der Klasse `RationaleZahl` mit Zähler n und Nenner m.

Nr.	Operation	Sollergebnis
1	Objekt $\frac{2}{8}$ erzeugen, also ein Objekt der Klasse `RationaleZahl` mit Zähler 2 und Nenner 8	
2	textuelle Darstellung des Objekts aus Schritt 1 auf Bildschirm ausgeben	`1/4`
3	Objekt $\frac{5}{6}$ erzeugen	
4	`addiere`-Methode für Objekt aus Schritt 1 mit Objekt aus Schritt 3 als Parameter aufrufen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	`13/12`
5	`addiere`-Methode für Objekt aus Schritt 3 mit Objekt aus Schritt 1 als Parameter aufrufen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	`13/12`
6	`addiere`-Methode für $\frac{17}{4}$ mit $\frac{7}{4}$ als Parameter ausführen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	6
7	`addiere`-Methode für $\frac{100002}{4}$ mit rationaler Zahl 5 als Parameter ausführen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	`50011/2`
8	`addiere`-Methode für $\frac{11}{2108303}$ mit $\frac{31}{2117009}$ als Parameter ausführen, zum Ergebnis $\frac{47}{2134421}$ addieren und dieses Ergebnis textuell auf dem Bildschirm ausgeben	`189502901/4524816707267`
9	`addiere`-Methode für $\frac{3}{3037000510}$ mit rationaler Zahl $\frac{7}{3037000500}$ als Parameter ausführen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	`3037000507/922337206737025500`
10	`addiere`-Methode für $\frac{10957693037}{10967535067}$ mit rationaler Zahl $\frac{10961461793}{10967639796}$ als Parameter ausführen und Ergebnis der Methode textuell auf dem Bildschirm ausgeben	`2295450608955127/1148564142356508`

Hinweise

- Verwenden Sie nur den Vorlesungsstoff bis einschließlich Kapitel „Klassen und Objekte".
- Es ist ratsam, zuerst die Klasse `RationaleZahlTest` zu schreiben, dann die Klasse `RationaleZahl`. Da Sie in der Testklasse Anwendungsfälle für die Konstruktoren und Methoden der Klasse `RationaleZahl` schreiben, wird Ihnen deren Bedeutung und Zusammenspiel besser bewusst und ihre spätere Realisierung erleichtert. Dass NetBeans die Verwendung noch nicht implementierter Methoden als Fehler ansieht, können Sie zwischenzeitlich ignorieren.
- Sie dürfen weitere statische und Instanzmethoden implementieren.
- Da es in dieser Aufgabe u. a. um das Kürzen von Brüchen geht, ist eine Methode zur Berechnung des größten gemeinsamen Teilers (ggT) zweier Zahlen sinnvoll. Der ggT zweier Zahlen m und n lässt sich rekursiv wie folgt berechnen:

$$ggT(m,n) = \begin{cases} m & \text{falls } n = 0 \\ ggT(n, m \; Rest \; n) & \text{sonst} \end{cases}$$

- Ein Aufruf der Methode `addiere` ist ein Ausdruck des Typs `RationaleZahl`, dessen Wert zur Laufzeit ein Objekt dieser Klasse ist. Deshalb sind auch geschachtelte Ausdrücke möglich, z. B.

```
new RationaleZahl(4, 3)
        .addiere(new RationaleZahl(7, 3)
                 .addiere(new RationaleZahl(5, 6)))
        .addiere(new RationaleZahl(1))
```

  zur Berechnung von $(\frac{4}{3} + (\frac{7}{3} + \frac{5}{6})) + 1$. Dies sollten Sie bei Ihren Tests berücksichtigen.
- Die Methode `gibAlsText` soll die rationale Zahl so weit wie möglich gekürzt *darstellen*. Dies bedeutet nicht, dass das Kürzen in dieser Methode erfolgen muss.
- Denken Sie an die ausreichende Dokumentation und Kommentierung Ihrer Lösung. Beachten Sie die unterschiedliche Bedeutung der *externen Dokumentation* `/** ... */` *vor* einer Klasse oder Methode und des *Implementierungskommentars* `/* ... */` *innerhalb* einer Methode. Die externe Dokumentation sagt, *was* eine Klasse oder Methode leistet, der Implementierungskommentar hilft zu verstehen, *wie* es gemacht wird. Verwenden Sie Implementierungskommentare vor allem, um den Berechnungsablauf verständlich zu machen.
- Erzeugen Sie die HTML-Dokumentation Ihrer Klasse und überzeugen Sie sich, ob Ihre externe Dokumentation sinnvoll und ohne Kenntnis des Quellcodes der Klasse hilfreich ist.
- Laden Sie die Dateien `RationaleZahl.java` und `RationaleZahlTest.java` zu Moodle hoch.