Programmierung 1
Prof. Dr. Jörg Kreiker
Wintersemester 2017/18

Hochschule Fulda
University of Applied Sciences

9. November 2017

Übungsblatt 4

Abgabe: 22. November 2017, 23:59 Uhr

Bearbeitung dieses Blattes

- Ladet Euch die Datei `Bruch.java` aus der Übungsaktivität in Moodle herunter.
- Diese Datei enthält eine Klassendefinition `Bruch`.
- Die `main` Methode dieser Klasse ist bereits definiert. Lasst diese *unverändert*! Sie dient später dem Testen der Lösungen.
- Die Klasse enthält ebenfalls bereits die Methode `ggt` aus der Vorlesung.
- In jeder Aufgabe wird eine oder mehrere neue Methoden zur Klasse `Bruch` hinzugefügt.
- Für jede Methode ist bereits ein Header und ein rudimentärer Rumpf angelegt, so dass das gesamte Programm kompiliert. Ändert keinesfalls die Header der Methoden sondern nur die Rümpfe.
- Abgabe am Ende: die bearbeitete Datei `Bruch.java` (sonst nix).

Aufgabe 1: Kleinstes gemeinsames Vielfaches (2 Punkte)

Schreibt eine Methode `kgv` zur Berechnung des kleinsten gemeinsamen Vielfachen zweier natürlicher Zahlen. *Hinweis*: Es bietet sich an, hierbei auf die ggt Methode zurückzugreifen.

Aufgabe 2: Kürzen (2 Punkte)

Schreibt eine Methode `kuerzen`, die Zähler und Nenner eines Bruches als Argumente bekommt und den *gekürzten* Bruch wie folgt auf die Konsole schreibt:

- Aufruf `kuerzen(30,18)` liefert: `5/3`
- Aufruf `kuerzen(30,5)` liefert: `6`

Aufgabe 3: Brüche addieren (2 Punkte)

Schreibt eine Methode `bruchSumme`, die jeweils Zähler und Nenner zweier Brüche erhält (Header beachten!). Die Methode soll die gekürzte Summe der Brüche auf der Konsole ausgeben. Beispiel: `bruchSumme(1,2,1,3)` liefert `1/2 + 1/3 = 5/6`.

Aufgabe 4: Ziffer 7 (2 Punkte)

Gegeben sei eine positive ganze Zahl n. Schreibt eine Methode `hat7`, die genau dann `true` zurückgibt, wenn n in der Dezimaldarstellung mindestens eine Ziffer 7 enthält. Beispiele: `hat7(4711)` liefert `true` und `hat7(42)` liefert `false`.

Aufgabe 5: Wallissches Produkt (2 Punkte)

Implementiert in der vorgegebenen Klasse `Bruch` die Methode `wallis`, die π mit Hilfe des *wallisschen Produktes* approximiert.

$$\frac{\pi}{2} = \prod_{k=1}^{\infty} \left(\frac{2k}{2k-1} \cdot \frac{2k}{2k+1}\right) = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdots$$