

Universität Potsdam
Institut für Informatik
Wintersemester 2010
Programmierung
Lehrblatt Woche 4

Name: _____

Matr.-Nr.: _____ Login-Name: _____

Die mit einem * versehenen Aufgaben werden korrigiert.

1. Schreiben Sie eine Klasse `Rational` rationaler Zahlen, die durch gemeine Brüche repräsentiert werden. Sie hat zwei gekapselte Datenelemente `num` (*numerator*, *Zähler*) und `denom` (*denominator*, *Nenner*) vom Typ `int`. Ferner soll die Klasse über folgende öffentliche Konstruktoren verfügen:

- `Rational()` zum Erzeugen der 0,
- `Rational(int)` zum Erzeugen der als Parameter angegebenen ganzen Zahl,
- `Rational(int, int)` zum Initialisieren beider Datenelemente.

Die Methoden der Klasse, alle öffentlich, sind:

- `int gcd()` zum Ermitteln des g.g.T. von Zähler und Nenner
(*Copy-and-Paste von Teilen aus Euklid.java?!*),
- `void reduce()` zum Kürzen des Bruches, so dass Zähler und Nenner von nun an teilerfremd sind (Überschreiben der Werte der Datenelemente!),
- `void extend(int)` zum Erweitern des Bruches mit dem als Parameter angegebenen Faktor (Überschreiben der Werte der Datenelemente!),
- `String toString()`, die den Bruch in der Form a/b zurückgibt,
- `String toStringReduced()`, die den gekürzten Bruch zurückgibt.

Rufen Sie, wenn immer es möglich ist, bereits definierte Methoden auf.

Testen Sie Ihre Klasse mit Hilfe von `UseRational.java` (Kopie aus `/home/rlehre/W4!`)

2.* Kopieren Sie die Datei `/home/rlehre/W4/Static.java`. Analysieren Sie die Applikation `Static` zur Demonstration von Effekten mit Klassenvariablen und beantworten Sie folgende Fragen.

(a) Die Klasse `Static` besitzt ein statisches Datenelement `snum`. Welche Aussage können Sie über den Wert von `snum` treffen, wenn ein Exemplar von `Static` mit dem impliziten Konstruktor erzeugt wird?

(b) Weshalb ist es sinnvoll, dass `increase()` und `output()` nicht statisch definiert sind?

(c) Weshalb kann der Aufruf von `set(5)` in der `main`-Methode ohne Qualifikation mit einem Objekt erfolgen?

(d) Weshalb kann der Aufruf von `set(5)` in der `main`-Methode ohne Qualifikation mit dem Klassennamen erfolgen?

3. Programmieren Sie eine Klasse `Gcd`, die eine statische Methode `gcd(int, int)` bereitstellt, mit der der größte gemeinsame Teiler der beiden `int`-Parameter ermittelt und zurückgegeben wird. Es soll der Euklid'sche Algorithmus verwendet werden.

4. Ergänzen Sie in `Rational` eine Methode `Rational mult(Rational)`, die das `Rational`-Objekt mit der gekürzten Form des Parameters multipliziert (Verwendung von `reduce()`). Das Produkt soll als neues Exemplar zurückgegeben werden. Testen Sie mit einer geeigneten Ergänzung in `UseRational`!

5.* Beantworten Sie folgende Fragen!

(a) Geben Sie zwei Möglichkeiten an, eine Variable `squares` vom Typ Verweis auf ein Array ganzer Zahlen (vom Typ `Integer`) zu definieren!

(b) Mit welcher Anweisung initialisieren Sie dieses Array mit den ersten drei Quadratzahlen (direkte Initialisierung)?

(c) Mit welchen Anweisungen wird das gleiche Array erzeugt, wobei der `new`-Operator verwendet wird?

(d) Analysieren Sie die Datei `SimpleOutputDemo.java`! Weshalb beginnt der Zähler der `for`-Schleife mit dem Wert von `args.length-1` und nicht von `args.length`?

- 6.* Geben Sie den Programmcode an, mit dem ein Array durch direkte Initialisierung erzeugt wird, das die ersten vier Zeilen des PASCALSchen Dreiecks speichert:

```
      1
     1 1
    1 2 1
   1 3 3 1
```

Das Array sollte einfach geschachtelt, d.h. ein Array von Arrays ganzer Zahlen sein.

7. Entwerfen Sie eine Applikation `FibArray`, die ein Array ganzer Zahlen erzeugt, dessen Elemente die ersten k Fibonacci-Zahlen sind. Die Größe k des Arrays soll beim Programmstart als Parameter übergeben werden. Ist dieser Parameter kleiner als 1, so gibt das Programm „Zu kleiner Parameter!“, sonst die Elemente des Arrays auf der Konsole aus. (Andere ungültige Parameter brauchen nicht behandelt zu werden.)
8. Weitere Notizen: