10. Übungsblatt: Programmierpraktikum I (WS 2003/04)

Abgabe: 19. Januar 2004

Bem.: Es dürfen nur solche Bibliotheken verwendet werden, die der Ein-/Ausgabe dienen oder die explizit angegeben sind.

Java: Klasse *Rational*

(`Rational.java`, 9 Punkte)

Schreibe eine Klasse **Rational**, die für die Arithmetik rationaler Zahlen a/b verwendet werden kann. Dazu gehören mindestens die folgenden Funktionen:

- Drei Konstruktoren:
  - ohne Parameter (setzt $a/b = 0/1$),
  - mit einem Parameter (long) x (setzt $a/b = x/1$),
  - mit zwei Parametern (long) x, y (setzt $a/b = x/y$).
- Funktionen `long zaehler()` und `long nenner()`, die Zähler bzw. Nenner zurückliefern.
- Eine Funktion `String toString()`, die a/b als String zurückgibt.
- Eine interne Hilfsfunktion `long ggT()`, die den größten gemeinsamen Teiler $ggT(a, b)$ berechnet.[1]
- Eine interne Hilfsfunktion `void kuerzen()`, die Zähler und Nenner durch den $ggT(a, b)$ teilt.
- Eine interne Hilfsfunktion `void normieren()`, die den Nenner positiv macht, indem sie ggf. Zähler und Nenner mit (-1) multipliziert.
- Eine Funktion `Rational betrag()`, die den Betrag der Zahl zurückgibt.
- Eine Funktion `Rational inverses()`, die das Inverse der Zahl (also bei a/b den Wert b/a) zurückgibt.
- Eine Funktion `boolean gleich(Rational)`, die vergleicht, ob die rationale Zahl gleich dem Parameter ist.

[1] Für einen effizienten Algorithmus zur Berechnung des ggT empfiehlt es sich, in Bibliothek oder Internet unter dem Schlagwort "Euklidischer Algorithmus" nachzuschlagen.

- Eine Funktion `boolean kleiner(Rational)`, die feststellt, ob die rationale Zahl kleiner als der Parameter ist.
- Funktionen
  - `Rational plus(Rational)`,
  - `Rational minus(Rational)`,
  - `Rational mal(Rational)` und
  - `Rational durch(Rational)`,

  die die vier Grundrechenarten zur Verfügung stellen.

Beachte bei allen Funktionen, dass keine ungültigen Zahlen wie $a/0$ entstehen dürfen. Tritt dieser Fall ein, so soll eine `ArithmeticException` ausgeworfen werden.
Außerdem sollen korrekte rationale Zahlen am Ende jedes Funktionsaufrufes (insbesondere nach dem Neuanlegen) in gekürzter und normierter Form sein. Umgekehrt darf man dann davon ausgehen, dass rationale Zahlen als Funktionsinput in gekürzter und normierter Form sind.

Java: Klasse *Complex* ableiten

(`Complex.java`, 6 Punkte)

In der Vorlesung wurde die Klasse *Complex* vorgestellt. Modifiziere diese Klasse so, dass sie von der abstrakten Klasse *Number* (aus *java.lang*) abgeleitet ist und die entsprechenden Methoden zur Typumwandlung bereitstellt.

Bem.: Zur Umwandlung von komplexen in nicht-komplexe Zahlen genügt es, den Real-Teil zu betrachten und entsprechend zu konvertieren.

Java: Integral-Abschätzung

(Dateien s.u., 9 Punkte)

Das Integral einer stetigen Funktion f im Intervall $[a, b]$ entspricht der Fläche unter dem Graphen von f, wie in Abbildung a) dargestellt. Diese Fläche kann man bekanntlich näherungsweise berechnen, indem man sie durch viele Rechtecke approximiert wie in Abbildung b). Dabei wird die Abschätzung umso besser, je feiner die Zerlegung des Intervalles $[a, b]$ ist.

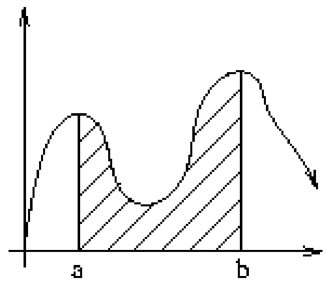

Abb. a)

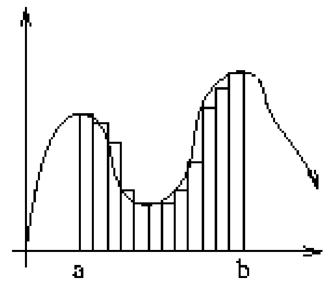

Abb. b)

Es soll nun ein Programm geschrieben werden, das diesen Approximationsalgorithmus für beliebige stetige Funktionen $f(x)$ implementiert, indem es das Intervall $[a, b]$ in n gleich große Teilintervalle $[x, y]$ zerlegt und für jedes dieser Intervalle die Fläche unter der Kurve durch ein entsprechendes Rechteck abschätzt. Die Abschätzung für das Gesamtintegral ergibt sich dann als Summe dieser Rechtecke. Um alle Unklarheiten zu beseitigen, hier noch einmal in formaler Schreibweise:

$$\int_a^b f(x)dx \quad \approx \quad \sum_{k=0}^{n-1} f(a + k \cdot \Delta x) \cdot \Delta x \quad \text{mit} \quad \Delta x = \tfrac{b-a}{n}$$

Zu implementieren sind die folgenden Elemente:

- Ein Interface `Zielfunktion`, das eine Methode `f(double)` definiert.
- Drei Klassen, die das Interface `Zielfunktion` implementieren und die Funktion f wie folgt definieren:

Klasse	*Funktion*
`Konst3`	$f(x) = 3$
`SqurPlus`	$f(x) = x^2 + 3x + 1$
`RootPlus`	$f(x) = \sqrt{x} + 1$

- Eine Methode

```
public static double integral(Zielfunktion g, double a,
                                    double b, int n)
```

  (in der Klasse `Aufgabe3`), die das Integral $\int_a^b g(x)$ durch n Rechtecke nach dem obigen Algorithmus approximiert.

- Ein Hauptprogramm

```
public static void main(String args[])
```

  (ebenfalls in der Klasse `Aufgabe3`), das die Methode `integral` durch geeignete Aufrufe für alle drei Funktionen testet.

Anmerkungen:

- Eine Ein-/Ausgabe-Schnittstelle ist nicht erforderlich; du kannst daher auch davon ausgehen, dass die Funktion `integral` nur korrekte Inputs erhält (stetige Funktion, $b > a$ usw.).
- Die korrekten Werte der Integrale für die Funktionen `Konst3`, `SqurPlus` und `RootPlus` sind leicht zu berechnen (Schulstoff). Auf diese Weise lässt sich auch nachprüfen, ob die Ergebnisse deiner Integral-Funktion näherungsweise stimmen können!