



## Programmieren: Einstieg in die Informatik mit Java

WS 2006/2007

Dr. G. Bohlender  
Dipl.–Math. techn. M. Richter

22.01.2007

### Aufgabenblatt 11

#### Aufgabe 27: *Adaptives Quadraturverfahren (Klausuraufgabe WS 2002/2003)*

Gewöhnliche Quadratur-Verfahren berechnen den Wert des Integrals

$$J(f) := \int_a^b f(x) dx \quad (1)$$

auf einem uniform zerlegten Gitter. Besitzt die Funktion  $f$  jedoch eine Singularität, so ist eine gleichmäßige Aufteilung nicht mehr effektiv. Man wählt dann anstelle des uniformen ein *adaptives Gitter*, d.h. die Entscheidung darüber, ob ein Teilintervall verfeinert wird, erfolgt mit Hilfe eines Fehlerindikators, der den lokalen Integrationsfehler schätzt. Dafür wird neben dem eigentlichen Quadratur-Verfahren ein *Kontroll-Verfahren* benötigt, welches eine i.d.R. höhere Genauigkeit als das eigentliche Quadratur-Verfahren aufweisen muß.

Approximiert man also das Integral mit Hilfe der *zusammengesetzten Simpson-Formel*

$$S_2(f) := \frac{h}{6} \left( f(a) + 4 f(a + h/2) + 2 f((a + b)/2) + 4 f(b - h/2) + f(b) \right), \quad (2)$$

$$h := \frac{b - a}{2}, \quad (3)$$

dann erfüllt z.B. die *Romberg-Näherung*  $T_{13}$  die oben gemachte Anforderung an das Kontroll-Verfahren.

$$T_{13}(f) := \frac{1}{15} (16 S_2(f) - S_1(f)), \quad (4)$$

$$S_1(f) := \frac{h}{3} \left( f(a) + 4 f((a + b)/2) + f(b) \right). \quad (5)$$

Man erkennt, daß für die Bestimmung der Romberg-Näherung keine weiteren Funktions-Auswertungen notwendig sind.

Setzen Sie das oben beschriebene *adaptive Simpson Quadratur-Verfahren* in einem Java-Programm `AdaptiveQuadratur` um. Verwenden Sie für die Implementierung die folgenden Elemente:

- Eine Klassenkonstante  $\text{EPSILON} = 10^{-8}$ , welche die Toleranz bei der Fehlerschätzung vorgibt, sowie eine öffentliche Klassen-Variable `counter`, die die Anzahl der benötigten Auswertungen der Funktion  $f$  ermittelt.

- Eine öffentliche Methode `berechneFunktionswert`, welche die Funktion  $f$  aus Gleichung (7) an einer vorgegebenen Stelle  $x$  auswertet.
- Desweiteren eine öffentliche Methode `berechneIntegral`, die den Integralwert  $J(f)$  mit Hilfe des adaptiven Simpson Quadratur-Verfahrens annähert. Übergeben Sie dazu der Funktion die Intervall-Grenzen  $a$  und  $b$  sowie die Funktionswerte an den Stellen  $a$ ,  $\frac{a+b}{2}$  und  $b$ .
  - (i) Berechnen Sie zunächst die Gitterweite  $h$  aus Gleichung (3) sowie die Intervallmitte  $m$ .
  - (ii) Bestimmen Sie die lokalen Integralwerte  $S_2$  und  $T_{13}$ . Minimieren Sie dabei die Anzahl der erforderlichen Funktionsaufrufe von  $f$ , d.h. speichern Sie bereits bekannte Funktionswerte in lokalen Variablen zwischen und werten Sie die Funktion  $f$  höchstens zwei weitere Male aus.
  - (iii) Geben Sie den Wert  $T_{13}$  an die aufrufende Funktion zurück, falls der geschätzte Fehler

$$e := |S_2 - T_{13}| \quad (6)$$

kleiner als die vorgegebene Toleranz `EPSILON` ist. Zerlegen Sie andernfalls das Intervall  $[a, b]$  in zwei Teilintervalle  $[a, m]$  und  $[m, b]$ , berechnen rekursiv auf beiden die lokalen Integralwerte und geben die Summe der beiden Werte an die aufrufende Funktion zurück.

Lesen Sie im Hauptprogramm das Intervall  $I = [0, 1]$  ein und bestimmen Sie numerisch als auch analytisch den Integralwert der Funktion

$$f(x) := \frac{8}{1+x^2} \quad \text{für } x \in \mathbb{R}. \quad (7)$$

Rufen Sie dazu die Funktion `berechneIntegral` mit den entsprechenden Parametern auf und initialisieren Sie den Zähler `counter` mit Drei. Vergleichen Sie die Genauigkeit des implementierten Quadratur-Verfahrens mit der analytischen Lösung, indem Sie den relativen Fehler ermitteln. Geben Sie zum Schluß die Integralwerte, den berechneten Fehler sowie die Anzahl der benötigten Auswertungen der Funktion  $f$  auf dem Bildschirm aus.

### **Aufgabe 28:** *Geometrische Berechnungen (Klausuraufgabe WS 2001/2002)*

Eine Form in der Geometrie hat im Allgemeinen einen Flächeninhalt  $A$  und einen Umfang  $U$ . Je nachdem, wie die Form aussieht, berechnet man  $A$  und  $U$  unterschiedlich. Für einen Kreis mit Radius  $r$  gilt:

$$A = \pi \cdot r^2 \quad (8)$$

$$U = 2 \cdot \pi \cdot r \quad (9)$$

Bei einem Rechteck mit den Seiten  $h$  und  $b$  berechnet man den Flächeninhalt und den Umfang jedoch folgendermaßen:

$$A = h \cdot b \quad (10)$$

$$U = 2 \cdot h + 2 \cdot b \quad (11)$$

Bilden Sie dieses Beispiel mit Hilfe der Klassen `Form`, `Kreis` und `Rechteck` in Java ab.

a) Die Klasse `Form` soll folgende Elemente besitzen:

- Eine Klassenvariable `flaeche` vom Typ `double`, welche den Flächeninhalt der geometrischen Form speichert.
- Eine Klassenvariable `umfang` vom Typ `double`, welche den Umfang speichert.
- Eine Klassenvariable `berechnet` vom Typ `boolean`, welche angibt, ob der Flächeninhalt und der Umfang schon berechnet worden sind.
- Einen Konstruktor, der die Variable `berechnet` mit `false` initialisiert.
- Eine öffentliche Methode `berechnen` vom Typ `void` ohne Parameter, in der die eigentlichen Berechnungen durchgeführt werden. In der Klasse `Form` soll nur die Variable `berechnet` auf `true` gesetzt werden.
- Öffentliche Methoden `getFlaeche` und `getUmfang` vom Typ `double` ohne Parameter, die zunächst prüfen, ob die Werte in den Variablen schon berechnet sind. Wenn dies nicht der Fall ist, dann wird die Methode `berechnen` aufgerufen. Anschließend soll der Wert der Variable `flaeche` bzw. der Variable `umfang` zurückgegeben werden.
- Eine öffentliche Methode `toString` vom Typ `String` ohne Parameter, die zunächst prüft, ob die Werte in den Variablen schon berechnet sind. Wenn nicht, dann wird die Methode `berechnen` aufgerufen. Anschließend soll folgendes Ergebnis zurückgegeben werden:

`Flaeche = f      Umfang = u`

Dabei entspricht `f` dem Wert von `flaeche` und `u` dem Wert von `umfang`.

b) Leiten Sie eine Klasse `Kreis` von der Klasse `Form` ab, die zusätzlich zu `Form` folgendes enthält:

- Eine Klassenvariable `r` vom Typ `double`, das dem Radius des Kreises entspricht.
- Einen Konstruktor mit einem Parameter vom Typ `double`, der zunächst den Konstruktor der Basisklasse aufruft und anschließend `r` mit dem Wert des Parameters initialisiert.
- Überschreiben Sie die öffentliche Methode `berechnen` (Typ `void`, keine Parameter). Zunächst wird der Flächeninhalt und der Umfang nach (8) bzw. (9) berechnet. Danach wird `berechnet` auf `true` gesetzt.

**Hinweis:**  $\pi$  ist in Java definiert als `Math.PI`

- Eine öffentliche Klassenmethode `readKreis` vom Typ `Kreis` ohne Parameter, die den Wert für `r` von der Konsole einliest und dann eine neue Instanz der Klasse `Kreis` zurückgibt.

**Hinweis:** Vergessen Sie nicht, am Anfang des Java-Programms die Klassenbibliothek `java.util.*` zu importieren.

c) Leiten Sie eine Klasse `Rechteck` von der Basisklasse `Form` ab. Zusätzlich besitzt ein `Rechteck` zwei Klassenvariablen `h` und `b` vom Typ `double`, die die Höhe und Breite eines Rechtecks repräsentieren.

Außerdem soll die Klasse `Rechteck` folgende Elemente besitzen:

- Einen Konstruktor mit zwei Parametern vom Typ `double`. Sie dienen zur Initialisierung der Klassenvariablen. Außerdem soll der Konstruktor der Basisklasse aufgerufen werden.
- Überschreiben Sie die öffentliche Methode `berechnen` (Typ `void`, keine Parameter). Zunächst wird der Flächeninhalt und der Umfang nach (10) bzw. (11) berechnet. Danach wird `berechnet` auf `true` gesetzt.
- Eine öffentliche Klassenmethode `readRechteck` vom Typ `Rechteck` ohne Parameter, die den Wert der Klassenvariablen von der Konsole einliest und dann eine neue Instanz der Klasse `Rechteck` zurückgibt.

d) Schreiben Sie ein Hauptprogramm, in dem die folgenden Aktionen mit Hilfe der definierten Konstruktoren, Methoden usw. ausgeführt werden:

- Lesen Sie ein Objekt der Klasse `Kreis` mittels `readKreis` ein, speichern Sie es in der Variablen `kr` und geben Sie `kr` mittels `System.out.println` aus.  
**Hinweis:** Wenn Sie ein Objekt mittels `System.out.println` ausgeben, dann erscheint das Ergebnis der Methode `toString()` des jeweiligen Objektes auf dem Bildschirm.
- Lesen Sie ein Objekt der Klasse `Rechteck` mittels `readRechteck` ein, speichern Sie es in der Variablen `re` und geben Sie `re` mittels `System.out.println` aus.
- Prüfen Sie, ob der Flächeninhalt von `kr` größer ist als der von `re`. Wenn ja, geben Sie eine entsprechende Meldung aus. Andernfalls geben Sie ebenfalls eine entsprechende Meldung aus.