LEHRSTUHL FÜR PRAKTISCHE INFORMATIK IV

UNIVERSITÄT MANNHEIM

Holger Füßler
A5, 6, Raum B 219
68131 Mannheim
Telefon: (0621) 181–2605
Email: fuessler@informatik.uni-mannheim.de

Robert Schiele
B6, 29, Raum C0.04
68131 Mannheim
Telefon: (0621) 181–2214
Email: rschiele@uni-mannheim.de

Praktische Informatik I — 6. Übungsblatt

Wintersemester 2005/2006 — Abgabe: 7. Dezember 2005

Allgemeine Bearbeitungshinweise

Verwenden Sie für die Programmieraufgaben die in den Aufgaben angegebenen Klassennamen und geben Sie den Klassen-Quellcode sowohl auf Papier als auch per e-mail bei Ihrem Tutor ab. Verwenden Sie Javadoc-Kommentare zur Dokumentation.

Aufgabe 1 — 5 Punkte

Beschreiben Sie den Algorithmus „Zähneputzen“, d.h. “bürstengestützte Zahnpflege in vernünftigem Umfang mit nicht-elektrischen Hilfsmitteln”. Beschreiben Sie den Algorithmus zunächst grob in Pseudocode und benutzen Sie dann „schrittweise Verfeinerung“. Der Detailierungsgrad soll so fein werden, dass ein Mensch ihn ohne zahnputztechnische Vorkenntnisse ausführen kann. Beachten Sie auch, welche „Module“ sie hierbei benötigen.

Aufgabe 2 — 8 Punkte

Der größte gemeinsame Teiler (ggT) zweier Ganzzahlen n und m $(n, m \geq 0)$ ist die größte natürliche Zahl, durch die beide Zahlen ohne Rest teilbar sind. Für die Bestimmung des ggT gibt es ein sehr altes und sehr berühmtes Verfahren von Euklid. Er funktioniert, indem man so lange iterativ die jeweils kleinere der beiden Zahlen von der gröseren abzieht, bis beide Zahlen gleich groß sind, d.h. dem ggT der Ausgangszahlen entsprechen.

Aufgabe 2 a) — 5 Punkte

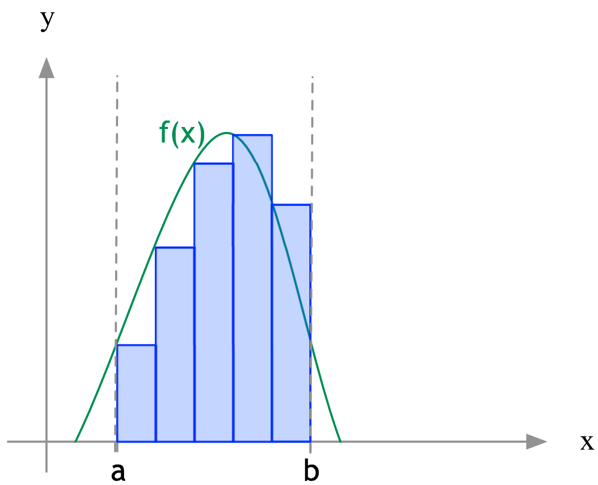
Formulieren Sie das Verfahren in Pseudocode, als Flussdiagramm und als Struktogramm. Beachten Sie dabei die Mächtigkeit von Java, z.B. kann man in Java nicht in einem Schritt zwei Werte vertauschen. Führen Sie ggf. schrittweise Verfeinerung durch.

Aufgabe 2 b) — 3 Punkte

Schreiben Sie ein kleines Java-Programm, das Ihren Algorithmus als (static-) Methode implementiert (Klassenname `EuklidGGT`) und in der Main-Routine geeignet testet.

Aufgabe 3 $\Sigma = 18$ Punkte

Gegeben sei der folgende Algorithmus zur näherungsweisen Berechnung eines Integrals, d.h. der Flächensumme zwischen einer Funktionskurve und der x-Achse:



Die Kurve sei gegeben durch $f(x)$, die linke Grenze des gesuchten Integrals durch a und die rechte durch b. Ferner sei N die Anzahl der Rechtecke, deren Flächensumme den Schätzer $\hat{A}$ für das tatsächliche Integral A bilden, d.h. in Formeln:

$$\hat{A}_N = \sum_{i=0}^{N-1} \left(f\left(a + i \cdot \frac{b-a}{N} \right) \cdot \frac{b-a}{N} \right) \tag{1}$$

Vereinfachend nehmen wir $b > a$ an.
Der eigentliche Algorithms läuft nun so ab, dass man so lange N erhöht, bis sich zwei benachbarte Folgenglieder nur noch um höchstens einen gewählten Betrag ε unterscheiden, also

$$\left| \hat{A}_{N-1} - \hat{A}_N \right| < \varepsilon \qquad\qquad \epsilon > 0 \tag{2}$$

Das tatsächliche Integral von Funktionen kann man unter gewissen Voraussetzungen bestimmen, in dem man die sog. Stammfunktion (auch unbestimmtes Integral genannt) $\int f(x)dx = F(x)$ bildet und dann den Wert der Stammfunktion an der rechten Grenze von dem Wert an der linken Grenze abzieht, also

$$A = F(b) - F(a) \qquad\qquad F(x) = \int f(x)dx \tag{3}$$

Im Folgenden sollen Sie ein Programm schreiben, das sowohl den Integralschätzalgorithmus als auch die analytische Berechnung auf einer Klasse von Funktionen, den Parabeln, berechnen kann. Hierfür seien zunächst das Interface `Funktion` und die abstrakte Klasse `IntegrierbareFunktion` wie folgt definiert:

```java
public interface Funktion {
    /** @returns f(wert) */
    double getFunktionswert(double wert);

    /** @returns A-Dach zwischen links (a) und rechts (b) fuer anzahlRechtecke
    (N), sie Formel (1) */
    double getIntegralschaetzung(double links, double rechts, int anzahlRechtecke);

   /** @returns A-Dach zwischen links (a) und rechts (b) fuer
   gegebenes epsilon (epsilon), sie Formel (2) */
    double getIntegralschaetzung(double links, double rechts, double
    epsilon );

}
```

```java
  public abstract class IntegrierbareFunktion implements Funktion {

    public abstract double getStammfunktionswert(double x);

    public double getIntegralflaeche(double links, double rechts) { /* ... */ }

    public double getIntegralschaetzung(double left, double right,
                                        int anzahlRechtecke) { /* ... */ }

    public double getIntegralschaetzung(double left, double right,
                                        double epsilon) { /* ... */ }

}
```

Aufgabe 3 a) — 7 Punkte

Implementieren Sie die Methoden `getIntegralflaeche()` und die beiden überladenen `getIntegralschaetzung()` Methoden. Definieren Sie **keine** zusätzlichen Instanzvariablen und vermeiden Sie redundanten Source-Code.
Warum sind diese Methodenimplementierungen schwer zu testen? Was kann/muss man machen, um Testen zu ermöglichen?

Aufgabe 3 b) — 7 Punkte

Implementieren Sie nun eine Klasse `Parabel` als Sub-Klasse von `IntegrierbareFunktion`. Eine Parabel ist ein Polynom zweiten Grades mit folgender Funktion bzw. Stammfunktion:

$$f(x) = \alpha x^2 + \gamma \tag{4}$$

$$F(x) = \frac{\alpha}{3} x^3 + \gamma x \tag{5}$$

Eine Parabel ist also durch α und γ vollständig beschrieben. (Hinweis: Verwenden Sie einen Konstruktor `Parabel(double alpha, double gamma)`). Überschreiben Sie auch die Methode `public String toString()`, die die Parabelinstanz geeignet als String darstellt (keine Grafikdarstellung!).

Aufgabe 3 c) — 4 Punkte

Schreiben Sie eine Klasse `Integral.java`, deren `main`-Methode Parabel-Objekte mit den Parametern $(\alpha = 1, \beta = 0)$, $(\alpha = 1, \beta = 3)$ und $(\alpha = 2, \beta = 5)$ erzeugt und dann das Integral zwischen 1 und 3 resp. 2 und 4 mit allen drei Methoden berechnet. Analytisch korrekt mit `getIntegralFlaeche()` und geschätzt mit fixer Rechteckanzahl (1, 5, und 10) und mit gegebenem ε $(\varepsilon = 10^{-5})$.