Institut für Angewandte und
Numerische Mathematik

PD Dr. Mathias J. Krause
M.Sc. Stefan Karch
M.Sc. Mariia Sukhova

18.11.2022

Einstieg in die Informatik und Algorithmische Mathematik

Aufgabenblatt 6

Bearbeitungszeitraum: 05.12.2022 – 16.12.2022

Aufgabe 1 *ggT und kgV*

Je zwei natürliche Zahlen m und n besitzen einen *größten gemeinsamen Teiler* $\mathrm{ggT}(m,n)$ und ein *kleinstes gemeinsames Vielfaches* $\mathrm{kgV}(m,n)$. Der größte gemeinsame Teiler kann mit dem *Euklidischen Algorithmus* berechnet werden, der folgendermaßen definiert ist:

(1) Falls $m > n$ gilt, setze $p_0 := m$ und $q_0 := n$, ansonsten $p_0 := n$ und $q_0 := m$.

(2) Setze $r_0 := p_0 \bmod q_0$.

(3) Solange $r_k \neq 0$ gilt, setze

$$\begin{aligned} p_{k+1} &:= q_k, \\ q_{k+1} &:= r_k, \\ r_{k+1} &:= p_{k+1} \bmod q_{k+1}, \end{aligned}$$

und erhöhe den formalen Iterationsindex k um Eins.

(4) Der größte gemeinsame Teiler von m und n ist q_{k+1}.

Mit $p_k \bmod q_k$ wird der Rest bezeichnet, der bei ganzzahliger Division von p_k durch q_k entsteht. Für das kleinste gemeinsame Vielfache von m und n gilt

$$\mathrm{kgV}(m,n) = \frac{mn}{\mathrm{ggT}(m,n)}.$$

Schreiben Sie ein Java-Programm, welches den größten gemeinsamen Teiler und das kleinste gemeinsame Vielfache zweier Zahlen berechnet und auf der Konsole ausgibt. Gehen Sie dabei folgendermaßen vor:

(a) Erstellen Sie eine öffentliche Klasse namens `GGTundKGV`. Definieren Sie in dieser Klasse eine öffentliche Klassenmethode namens `zahlenOK` mit zwei formalen Parametern vom Typ **`int`**. In der Methode soll geprüft werden, ob die beiden übergebenen Parameterwerte positive Zahlen sind. Das Ergebnis dieser Überprüfung soll als Wert vom Typ **`boolean`** von der Methode zurück gegeben werden.

(b) Definieren Sie eine öffentliche Klassenmethode namens `ggT` mit zwei formalen Parametern vom Typ **`int`**. Berechnen Sie in dieser Methode den größten gemeinsamen Teiler der übergebenen Parameterwerter. Implementieren Sie dazu den Euklidischen Algorithmus. Der größte gemeinsame Teiler soll von der Methode als Wert vom Typ **`int`** zurück gegeben werden.

   Hinweis: Die Werte p_k, q_k und r_k können in jeweils einer einzelnen Variable gespeichert werden. Der formale Iterationsindex k braucht nicht gespeichert zu werden.

(c) Definieren Sie eine öffentliche Klassenmethode namens `kgV` mit zwei formalen Parametern vom Typ **`int`**. Berechnen sie in dieser Methode das kleinste gemeinsame Vielfache der übergebenen Parameterwerte. Verwenden Sie dazu die Methode `ggT`. Das kleinste gemeinsame Vielfache soll von der Methode als Wert vom Typ **`int`** zurück gegeben werden.

(d) Definieren Sie eine öffentliche Klassenmethode namens `ausgabe` ohne Rückgabewert mit zwei formalen Parametern vom Typ **`int`**. Die Methode soll den größten gemeinsamen Teiler und das kleinste gemeinsame Vielfache der übergebenen Parameterwerte auf der Konsole ausgeben.

(e) Definieren Sie als letztes die `main`-Methode des Programms. Lesen in dieser zwei Zahlen vom Typ **`int`** von der Konsole ein. Überprüfen Sie zunächst, ob beide Zahlen positiv sind. Wenn ja, so geben Sie den größten gemeinsamen Teiler und das kleinste gemeinsame Vielfache beider Zahlen auf der Konsole aus. Wenn nicht, so geben Sie eine entsprechende Meldung auf der Konsole aus. Verwenden Sie die Methoden `zahlenOK` und `ausgabe`.

(f) Testen Sie ihr Programm mit dem Zahlenpaar $(12, 3)$, dessen $\mathrm{ggT}\ 3$ ist, mit dem kgV von 12.

Aufgabe 2 (Pflichtaufgabe) *Das zweistufige Horner–Schema*

Polynome sind Funktionen $f : \mathbb{R} \to \mathbb{R}$ von der Form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} \cdots + a_1 x + a_0 \qquad \text{für alle } x \in \mathbb{R}. \tag{I}$$

Die reellen Zahlen $a_0, \dots, a_n$ werden die *Koeffizienten* des Polynoms genannt, wobei $a_n \neq 0$ vorausgesetzt wird. Die Zahl n nennt man den *Grad* des Polynoms. Sei eine Stelle $x \in \mathbb{R}$ fest gewählt, so kann man durch sukzessives Ausklammern folgende Darstellung für den Funktionswert $f(x)$ gewinnen:

$$f(x) = \big(\cdots\big((a_n x + a_{n-1})x + a_{n-2}\big)x \cdots + a_1\big)x + a_0. \tag{II}$$

Definiert man die reellen Zahlen $b_0, b_1, \dots, b_{n-2}, b_{n-1}$ rekursiv durch

$$b_{n-1} := a_n, \qquad b_{k-1} := x b_k + a_k \qquad \text{für } k = n-1, n-2, \dots, 0,$$

so kann man für den Funktionswert $f'(x)$ der Ableitung des Polynoms die folgende Darstellung herleiten:

$$f'(x) = \big(\cdots(b_{n-1}x + b_{n-2})x \cdots + b_1\big)x + b_0. \tag{III}$$

Dies ist die Grundlage des sogenannten *zweistufigen Horner–Schemas*, einem Verfahren, mit dem Polynome und deren Ableitungen effizient ausgewertet werden können. Das zweistufige Hornerschema wertet die Klammern in Formel (II) und (III) quasi „von innen nach außen“ aus. Es kann als Algorithmus folgendermaßen beschrieben werden:

(1) Sei $x \in \mathbb{R}$ gegeben. Setze $y_0 := a_n$ und $z_0 := a_n$.

(2) Für $k = 1, 2, \dots, n$ setze $y_k := xy_{k-1} + a_{n-k}$, und für $k = 1, 2, \dots, n-1$ setze $z_k := xz_{k-1} + y_k$.

(3) Es gilt $f(x) = y_n$ und $f'(x) = z_{n-1}$.

Schreiben Sie ein Java–Programm, welches ein Polynom und dessen Ableitung nach dem Horner–Schema auswertet. Gehen Sie dabei folgendermaßen vor:

(a) Erstellen Sie eine öffentliche Klasse namens `Hornerschema` mit einer `main`-Methode. Lesen Sie in der `main`-Methode zunächst den Grad n eines Polynoms f ein. Erzeugen Sie als nächstes ein Feld der Länge $n + 1$ vom Typ **`double`**. Lesen Sie die Koeffizienten $a_0, \dots, a_n$ von der Konsole ein, und speichern Sie diese im Feld ab. Verwenden Sie dazu eine **`for`**-Schleife.

(b) Lesen Sie eine Stelle x von der Konsole ein, und berechnen Sie den zugehörigen Wert des Polynoms $f(x)$ sowie den Wert der Ableitung $f'(x)$ mit dem zweistufigen Horner–Schema, und geben Sie beide Ergebnisse auf der Konsole aus. Um die Werte y_k und z_k zwischen zu speichern sollten keine Felder sondern lediglich zwei einfache Variablen verwendet werden.

(c) Testen Sie Ihr Programm mit den Polynomen $f : \mathbb{R} \to \mathbb{R},\ f(x) := x^3 + 2x^2 + 1$. Es gilt

$\boldsymbol{x}$	-2	-1	0	1	2
$\boldsymbol{f(x)}$	1	2	1	4	17
$\boldsymbol{f'(x)}$	4	-1	0	7	20

Fragen 2 *Das zweistufige Horner–Schema*

- Was für Möglichkeiten gibt es Ergebnisse auf schwerwiegende Rundungsfehler zu überprüfen?
- Geben Sie ein Beispiel für einen Rundungsfehler durch Auslöschung an.

Aufgabe 3 *Kapitalwert*

Unternehmen stehen häufig vor der Entscheidung, ob sie eine bestimmte Investition tätigen sollen oder nicht. Eine Möglichkeit zur Beurteilung von Investitionen ist die sogenannte *Kapitalwertmethode*.

Man geht dabei von folgendem Modell aus: Eine Investition wird zum Zeitpunkt $t = 0$ getätigt und läuft dann T Zeitperioden. Am Ende jeder Zeitperiode t verursacht die Investition eine Ein-oder Auszahlung a_t. Dabei entspricht a_0 der Investitionssumme. Unter der Annahme, dass man die Investitionssumme auch zu einem *Kalkulationszinsfuß* i hätte anlegen können, werden die Zahlungen a_t mit dem Faktor $(1+i)^{-t}$ diskontiert. Der Kapitalwert C_0 der Investition errechnet sich daher gemäß

$$C_0 = \sum_{t=0}^{T} a_t (1+i)^{-t}. \tag{I}$$

Eine Investition wird als vorteilhaft beurteilt, wenn ihr Kapitalwert nichtnegativ ist. Erstellen Sie mit Java ein Programm, das den Kapitalwert einer Investition berechnet. Gehen Sie dabei wie folgt vor:

- Erstellen Sie eine `main`–Methode in einer öffentlichen Klasse namens `Kapitalwert`. Lesen Sie zunächst die Anzahl der Zeitperioden T (als ganze Zahl) und den Kalkulationszinsfuß i (als reelle Zahl) von der Konsole ein, und speichern Sie diese Größen in geeigneten Variablen ab.
- Erzeugen Sie ein Feld der Länge $T+1$, um die Zahlungen $a_0, a_1, \ldots, a_T$ (reelle Zahlen) abspeichern zu können. Lesen Sie anschließend die einzelnen Zahlungen von der Konsole ein, und speichern Sie diese im Feld ab.
- Berechnen Sie den Kapitalwert der Investition gemäß (I) und geben Sie diesen auf der Konsole aus.

  Hinweis:

  - Verwenden Sie zur Berechnung der Summe eine **`for`**–Schleife.
  - Ausdrücke wie a^x können in Java mit dem Ausdruck `Math.pow(a,x)` berechnet werden, wobei die Werte von a und x in den Variablen `a` und `x` gespeichert sein müssen.

Musterlösung: Testen Sie Ihr Programm anhand der folgenden fiktiven Investitionen A und B bei einem Kalkulationszinsfuß von 3.0%:

Investition A

t	a_t
0	-1000
1	400
2	300
3	200
4	100
5	100

Kapitalwert: 29.27

Investition B

t	a_t
0	-1000
1	100
2	100
3	200
4	300
5	400

Kapitalwert: -14.04