Aufgabenblatt 03

Bearbeitungsende: 30.10.2022

Empfehlung:

- Bearbeiten Sie die Aufgaben auf der Kommandozeile bzw. in einem einfachen Texteditor.
- Speichern Sie die Dateien zu jeder Aufgabe X in einem Verzeichnis `eidip/b03aX`.

Hinweis: Wenn Sie sich mit einer Aufgabe befassen, zeichnen Sie sich erst auf Papier einige Beispiele auf und denken Sie sich in das Problem hinein. Überlegen Sie sich eine Vorgehensweise und spielen Sie diese an den Beispielen manuell durch. Erst, wenn Sie eine klare Vorstellung und Skizze haben, wie Sie vorgehen wollen, sollten Sie den Editor öffnen und den Code schreiben.

Aufgabe 1 [Programmierung – nicht bewertet]

Hinweis: Eine lange Aufgabenstellung heißt nicht unbedingt, dass die Aufgabe schwer ist (Entsprechendes gilt umgekehrt). Lesen Sie sich alles in Ruhe durch, lassen Sie sich nicht durcheinander bringen, wenn Sie etwas nicht verstehen (streichen Sie es sich einfach an und fragen Sie dazu nach), und unterscheiden Sie zwischen reiner Information und tatsächlich zu erledigender Aufgabe (z.B. indem Sie entsprechende Textstellen markieren).

Der *größte gemeinsame Teiler* („ggT“) zweier nichtnegativer ganzer Zahlen a und b, geschrieben $\mathrm{ggT}(a, b)$, ist die größte ganze Zahl, durch die sich sowohl a als auch b ohne Rest teilen lassen.

Beispiel: Ist $a = 24$ und $b = 78$, so ist $\mathrm{ggT}(a, b) = 6$, denn $24 = 6 \cdot 4$, $78 = 6 \cdot 13$, und es gibt keine größere Zahl, die a und b teilt (weil 4 und 13 keine gemeinsamen Teiler haben).

$\mathrm{ggT}(\mathrm{a}, \mathrm{b})$ kann auf folgende unterschiedliche Weisen berechnet werden (warum dies so ist, spielt an dieser Stelle keine Rolle, sondern ist Thema der Mathematik):

(a) *additiver Euklidischer Algorithmus*:

(1) wenn $a = 0$: Ergebnis ist b
(2) (es ist nun $a > 0$)
wenn $b > 0$:
(2a) wenn $a > b$: setze $a := a - b$
(2b) sonst ($b \geqslant a$): setze $b := b - a$
(2c) wiederhole ab (2)
(3) (es ist nun $b = 0$)
Ergebnis ist a

Hinweis: Falls Sie sich wundern, dass nur einmal geprüft wird, ob $a = 0$: Dies liegt daran, dass durch die darauf folgenden Schritte ein positives a stets positiv bleibt.

(b) *multiplikativer Euklidischer Algorithmus*:

(1) wenn $b > 0$:
(1a) setze $t := a \bmod b$ (t ist eine temporäre Variable)
(1b) setze $a := b$
(1b) setze $b := t$

(1c) wiederhole ab (1)

(2) (es ist nun $b = 0$)
Ergebnis ist a

Schreiben Sie eine Klasse `Mathe` mit folgenden Methoden, die jeweils zwei ganze Zahlen `a` und `b` als Parameter haben. In den Fällen (a)–(d) sind die Parameter nichtnegativ (wovon Sie ausgehen können).

(a) `ggTAdd` führt den additiven Euklidischen Algorithmus iterativ durch und gibt ggT(`a`, `b`) zurück.

(b) `ggTMul` führt den multiplikativen Euklidischen Algorithmus iterativ durch und gibt ggT(`a`, `b`) zurück.

(c) `ggTAddOut` führt den additiven Euklidischen Algorithmus iterativ durch. Zu Beginn jeder Wiederholung der Schleife (also vor Schritt (2a)) werden die Werte von a und b, von einem Leerzeichen getrennt, auf einer Zeile auf dem Bildschirm ausgegeben. Ebenso werden diese beiden Werte zum Schluss ausgegeben, also vor Schritt (3) bzw. im Fall $a = 0$ in Schritt (1). Die Methode gibt als Ergebnis zurück, wie oft die Schleife durchlaufen wurde.

Hinweis: Sie müssen also eine zusätzliche Variable einführen, die die Wiederholungen mitzählt.

Beispiel: Der Aufruf von `ggTAddOut(24,78)` produziert auf dem Bildschirm die Ausgabe

```
24 78
24 54
24 30
24 6
18 6
12 6
6 6
6 0
```

(gefolgt von einem Zeilenumbruch) und gibt den Wert `7` zurück.

(d) `ggTMulOut` führt analog zu `ggTAddOut` den multiplikativen Euklidischen Algorithmus iterativ durch, gibt dabei jeweils vor den Schritten (1a) und (2) die Zwischenwerte auf dem Bildschirm aus und liefert als Ergebnis die Zahl der Wiederholungen des Verfahrens.

(e) `ggT` ruft `ggTMul` mit den Werten |`a`| und |`b`| auf und gibt das Ergebnis dieses Aufrufs zurück.

Hinweis: Verwenden Sie die Methode `Math.abs`.

Was fällt Ihnen im Vergleich der Ausgaben der beiden Verfahren auf?

Aufgabe 2 [Programmierung]

Schreiben Sie eine Klasse `Mathe` mit folgenden Methoden:

(a) `vielfache` hat zwei positive (wovon Sie ausgehen können) ganzzahlige Parameter `a` und `n`. Die Methode gibt eine Zeichenkette zurück, die die ersten `n` Vielfachen von `a` enthält, also die Zahlen vis `a`·`n`. Auf jede Zahl folgt ein Leerzeichen, am Ende der Zeichenkette steht ein Zeilenumbruch.

Beispiel: Hat `a` den Wert `3` und `n` den Wert 5, gibt die Methode die folgende Zeichenkette zurück (␣ steht darin jeweils für ein Leerzeichen):

`"3␣6␣9␣12␣15␣\n"`

(b) `einmaleins` hat einen positiven (wovon Sie ausgehen können) ganzzahligen Parameter `n`. Die Methode gibt eine Zeichenkette zurück, die zeilenweise das „Einmaleins bis `n`“ aus, also alle Produkte $a \cdot b$ von Zahlen $1 \leqslant a, b \leqslant$ `n`. In der ersten Zeile stehen die Vielfachen von 1, in der zweiten die Vielfachen von 2, und so weiter bis zur letzten Zeile, in der die Vielfachen von `n` stehen. Auf jede der Zahlen folgt ein Leerzeichen.

Beispiel: Hat `n` den Wert 5, gibt die Methode eine Zeichenkette zurück, deren Ausgabe auf dem Bildschrim folgendermaßen aussähe (mit abschließendem Zeilenumbruch!):

```
1␣2␣3␣4␣5␣
2␣4␣6␣8␣10␣
3␣6␣9␣12␣15␣
4␣8␣12␣16␣20␣
5␣10␣15␣20␣25␣
```

Hinweis: Implementieren Sie `einmaleins` mit Hilfe der Methode `vielfache`.

Aufgabe 3 [Theorie]

Geben Sie logische Ausdrücke in `Java` für die folgenden Sachverhalte an:

(a) `w` liegt im Intervall $[0; 100[$, d.h. in der Menge von Zahlen von 0 (inklusive) bis 100 (exklusive)

(b) `x` liegt im Intervall $[-20; 20]$, d.h. in der Menge von Zahlen von -20 (inklusive) bis 20 (inklusive), aber nicht im Bereich $[4; 8]$

(c) `y` ist eine (nicht zu prüfen: ganze) ungerade nichtnegative Zahl

(d) `z1` ist entweder kleiner als `z2` oder beide Zahlen sind größer als 12

Aufgabe 4 [Programmierung]

Schreiben Sie eine Klasse `Datum` mit einer Klassenmethode `istSchaltjahr`. Die Methode soll als Argument eine positive (was nicht zu überprüfen ist) ganze Jahreszahl annehmen. Sie soll einen Wahrheitswert zurückgeben, der angibt, ob das entsprechende Jahr ein Schaltjahr war/ist.

Vor 1583 (*Julianischer Kalender*) war ein Jahr ein Schaltjahr, wenn die Jahreszahl durch 4 teilbar war.

Seit 1583 (*Gregorianischer Kalender*) ist ein Jahr ein Schaltjahr, wenn die Jahreszahl

- durch 4 teilbar ist,
- aber nicht durch 100 teilbar ist,
- außer, wenn sie auch durch 400 teilbar ist (2000 war ein Schaltjahr).

Hinweis: Verwenden Sie den Modulo-Operator, um Teilbarkeit zu prüfen.

Schreiben Sie sich zur Hilfe ein Testprogramm, das die Methode mit verschiedenen Jahreswerten aufruft und die Ergebnisse ausgibt.

Probieren Sie verschiedene Möglichkeiten der Implementierung aus:

(a) nur mittels `if-else`
(b) nur mittels logischer Operatoren (knifflig!)

(c) als Mischform

Reichen Sie die Variante in den Praktomat ein, die Ihnen am besten gefällt.

Lösungen zu mit [**Programmierung**] markierten Aufgaben sind im **Praktomat** einzureichen.

Lösungen zu mit [**Programmierung – nicht bewertet**] markierten Aufgaben können ebenfalls im **Praktomat** eingereicht werden, werden jedoch nicht bewertet.

Allgemeine **Fragen** zu den Aufgaben können Sie im **LEA-Forum „Übungsaufgaben“** stellen.

Hilfe bei der Lösung der Aufgaben erhalten Sie in den **Übungen** und in der **Studierwerkstatt** .