

Aufgabenblatt 08

Bearbeitungsende: 01.12.2019

Aufgabe 1 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse **Zahlen** mit den folgenden Klassenmethoden, die die wichtigsten arithmetischen Ganzzahloperationen aus den Grundoperationen++, -- und - (Negation eines Wertes, nicht Differenz zweier Werte) nachbilden.

Das Ergebnis soll jeweils mittels einer Schleife sukzessive berechnet werden.

Verwenden Sie dabei **keine arithmetischen Operatoren** außer den genannten Grundoperatoren. Für höhere Operationen wie Multiplikation oder Division verwenden Sie die von Ihnen erstellten Methoden wie Addition oder Subtraktion.

Hinweis: Gehen Sie zunächst davon aus, dass beide Argumente nichtnegativ sind.

- (a) `addLoop(int a, int b)` soll den Wert $a+b$ berechnen und zurückgeben, ohne jedoch den Operator `+` zu verwenden.
- (b) `subLoop(int a, int b)` soll den Wert $a-b$ berechnen und zurückgeben, ohne jedoch den (Subtraktions)Operator `-` zu verwenden.
- (c) `mulLoop(int a, int b)` soll den Wert $a*b$ berechnen und zurückgeben, ohne jedoch den Operator `*` zu verwenden.
- (d) `divLoop(int a, int b)` soll den Wert a/b berechnen und zurückgeben, ohne jedoch den Operator `/` zu verwenden.

Es soll $a \geq 0$ und $b > 0$ gelten, damit die Operation eindeutig definiert ist.

- (e) `modLoop(int a, int b)` soll den Wert $a \bmod b$ berechnen und zurückgeben, ohne jedoch den Operator `%` zu verwenden.

Die Voraussetzungen sind dieselben wie für `divLoop`.

- (f) `powLoop(int a, int b)` soll den Wert a^b berechnen und zurückgeben (z.B. für 3 und 2 den Wert $3^2 = 9$).

Verwenden Sie nicht die Methode `Math.pow`.

Es muss $b \geq 0$ gelten, damit das Ergebnis immer ganzzahlig ist.

0^0 sei hier 1.

Erweitern Sie die Methoden schließlich so, dass sie mit allen zulässigen (auch negativen) ganzzahligen Argumenten arbeiten. Gehen Sie davon aus, dass keine ungültigen Argumente (Division durch 0) übergeben werden.

Hinweis: Ggf. können Sie einen einmaligen rekursiven Aufruf verwenden.

Aufgabe 2 [Programmierung]

Sinn dieser Aufgabe ist, dass Sie sich mit der Verwendung der Rekursion vertraut machen.

Schreiben Sie eine alternative Version der Klasse `Zahlen` aus Aufgabe 1 mit folgenden Methoden, die sich so verhalten wie die schleifenbasierten Versionen, jedoch mittels Rekursion implementiert sind:

- (a) `addRek(int a, int b)`
- (b) `subRek(int a, int b)`
- (c) `mulRek(int a, int b)`
- (d) `divRek(int a, int b)`
- (e) `modRek(int a, int b)`
- (f) `powRek(int b, int e)`

Hinweis: Entwickeln Sie die Lösung zunächst unter der Annahme nichtnegativer Argumente. Verwenden Sie dann diese rekursiven Methoden als Hilfsmethoden; die Einstiegsmethoden behandeln negative Argumente.

Aufgabe 3 [Programmierung – nicht bewertet]

Schreiben Sie eine Klasse `Zahlen` mit zwei Klassenmethoden, die jeweils eine nichtnegative Zahl `p` annehmen und als Wahrheitswert liefern, ob `p` eine Primzahl ist.

Man kann die Aufgabe mittels „brute force“ lösen, indem man alle Fälle durchprobiert: Wenn $p > 1$ und `p` durch keine Zahl `n` mit $1 < n < p$ teilbar ist, ist sie eine Primzahl.

Implementieren Sie dieses Verfahren

- in Methode `isPrimLoop` mittels einer Schleife und
- in Methode `isPrimRek` mittels Rekursion.

Hinweis: Verwenden Sie im zweiten Fall eine rekursive Hilfsmethode, die den aktuell geprüften Wert `n` als zweites Argument annimmt.

Aufgabe 4 [Programmierung]

Ein Palindrom ist eine Zeichenkette, welche vorwärts und rückwärts gelesen gleich ist, abgesehen von Groß- und Kleinschreibung, z.B. "Rentner" oder "4163614" oder "Lager-3-Regal".

Schreiben Sie eine Klasse `Zeichenketten` mit Klassenmethoden, die eine Zeichenkette als Argument annehmen und als Wahrheitswert zurückgeben, ob die übergebene Zeichenkette ein Palindrom ist.

Schreiben Sie die Methode in einer Variante `isPalindromLoop` als Schleife und in einer Variante `isPalindromRek` als Rekursion.

Verwenden Sie keine `String`-Methoden außer `charAt` und `length`.

Hinweise:

- Vergleichen Sie jeweils zwei Zeichen miteinander.

- Vereinfachen Sie den Vergleich mit Hilfe der Methode `Character.toLowerCase`.
- Verwenden Sie in der rekursiven Variante eine Hilfsmethode, die als Argument außer der Zeichenkette die Positionen der aktuell zu vergleichenden Zeichen erhält.

optionale Ergänzung (nicht bewertet): Erweitern Sie die Lösung so, dass nur Buchstaben und Ziffern berücksichtigt werden, Leer-, Satz- und Sonderzeichen hingegen ignoriert werden. So wird ein Palindromsatz über Wortgrenzen möglich: "Sie, mit Frust, surft im Eis."

Hinweis: Verwalten Sie dazu die beiden zu untersuchenden Positionen separat. Rücken Sie, falls ein Zeichen irrelevant ist, die entsprechende Position vor und prüfen Sie erneut. In der Schleifenversion kann `continue` hilfreich sein.

Aufgabe 5 [Programmierung – nicht einzureichen]

Schreiben Sie eine schnellere Variante der Berechnung einer Potenz a^b aus ganzen Zahlen a und $b \geq 0$.

Die Idee dazu ist, dass b

- entweder gerade ist, so dass $b == b/2 + b/2$, also $a^b = a^{2 \cdot b/2} = (a^2)^{b/2}$,
- oder ungerade ist, so dass $b == b/2 + b/2 + 1$, also $a^b = a^{2 \cdot b/2 + 1} = (a^2)^{b/2} \cdot a$.

Auf diese Weise kann in jedem Rechenschritt die Größe von b halbiert werden (statt lediglich um 1 verringert zu werden), bis sich für $b == 0$ das Resultat 1 ergibt.

Schreiben Sie eine Klasse `Zahlen` mit einer Klassenmethode `powLoop`, die das Verfahren in einer Schleife ausführt, und einer Klassenmethode `powRek`, die rekursiv arbeitet. Beide Methoden nehmen die Argumente a und b als Argumente und geben das Ergebnis als ganze Zahl zurück.

Für ungültige Argumentwerte soll eine `IllegalArgumentException` geworfen werden.

Zusatzfrage (optional): Was passiert in Ihren Lösungen, wenn Sie stattdessen die mathematisch äquivalente Zerlegung $(a^{b/2})^2$ zu verwenden versuchen?

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**.