

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

13.11.2023

Einstieg in die Informatik und Algorithmische Mathematik

Aufgabenblatt 5

Bearbeitungszeitraum: 27.11.2023 – 08.12.2023

Aufgabe 1 (Pflichtaufgabe) *Berechnung der Exponentialfunktion*

In dieser Aufgabe sehen Sie am Beispiel der Exponentialfunktion, dass es schon bei der Umsetzung einfacher mathematischer Formeln erhebliche Genauigkeitsprobleme geben kann. Die Funktionswerte der Exponentialfunktion e^x können aus der Reihendarstellung

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots$$

berechnet werden. Die Funktionswerte sollen durch endliche Summen

$$e^x \approx S(N) := \sum_{i=0}^N \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots + \frac{x^N}{N!}$$

angenähert werden. Die einzelnen Summanden $y_i := x^i/i!$ lassen sich dabei wie folgt berechnen:

$$\begin{aligned} y_0 &:= 1 \\ y_i &:= \frac{x}{i} y_{i-1} \quad , \quad i = 1, 2, \dots \end{aligned}$$

Schreiben Sie ein Java-Programm, welches die Werte der Exponentialfunktion nach obigen Formeln berechnet. Legen Sie Ihren Berechnungen Variablen, welche Gleitkommazahlen speichern, zugrunde. Gehen Sie dazu folgendermaßen vor:

- Erstellen Sie eine öffentliche Klasse `Exponential` mit einer `main`-Methode. Hier soll der Benutzer zunächst dazu aufgefordert werden, die auszuwertende Stelle x sowie eine obere Summationsgrenze N einzugeben. Speichern Sie die Werte in entsprechende Variablen.
- Führen Sie die obige Summation zur Berechnung von $S(N)$ mittels einer `for`-Schleife durch, wobei N der eingegebenen Summationsobergrenze entspricht. Berechnen Sie dabei den Wert des aktuellen Summanden wie angegeben aus dem Wert des vorherigen Summanden. Definieren Sie sich dazu schon vor der `for`-Schleife eine Variable `summe`. Geben Sie das Ergebnis mit einem passendem Text auf der Konsole aus.

- Führen Sie die obige Summation erneut durch, jedoch soll dieses Mal die Summation erst abbrechen wenn $|S(N) - S(N+1)| \leq 10^{-13}$ gilt. Nutzen Sie dafür eine **do-while**-Schleife. Bei der **do-while**-Schleife wird zur Überprüfung der Bedingung noch eine weitere Variable `summealt` benötigt. Geben Sie das Ergebnis, und bei welchem N das Ergebnis erreicht wurde auf der Konsole aus.

Hinweis: Die Betragsfunktion $|\cdot|$ heißt unter Java `Math.abs()`.

- Vergleichen Sie die berechneten Werte mit dem Ergebnis der Standardfunktion `Math.exp()`, welches Sie ebenfalls auf der Konsole ausgeben lassen sollen. Belegen Sie, dass dieses Verfahren insbesondere für negative x -Werte ungeeignet ist. Testen Sie Ihr Programm mit den Werten $x = \pm 1, \pm 10, \pm 50, \pm 100$.
- Verbessern Sie Ihr Programm, indem Sie die Berechnung mit automatischem Abbruch, nach der erfüllten Bedingung von oben, erneut implementieren, jedoch mit einer Änderung: Für negative x soll nun die Formel $e^{-x} = 1/e^x$ genutzt werden, d.h. berechnen Sie zunächst $e^{|x|}$ und invertieren Sie das Ergebnis wenn nötig. Geben Sie auch dieses Ergebnis mit erklärendem Text auf der Konsole aus.
- Testen Sie Ihr Programm erneut mit $x = \pm 1, \pm 10, \pm 50, \pm 100$. Lassen Sie sich hierbei die Ergebnisse aller jeweils verbesserten Versionen ausgeben und vergleichen Sie vor allem das verbesserte Ergebnis für negative Zahlen mit den Ergebnissen der Standardfunktion.

Musterlösung:

Bitte den Wert fuer x eingeben:

-100

Bis zu welchem Index soll die for-Schleife summieren?

1000

Der berechnete Wert nach 1000 Schritten: 8.144652745098073E25

Der berechnete Wert nach 243 Schritten mit Abbruch nach Genauigkeit: 8.144652745098073E25

Verbesserter Wert fuer negative x-Werte nach 192 Schritten: 3.7200759760208336E-44

Wert der Math-Bibliothek: 3.720075976020836E-44

Bitte den Wert fuer x eingeben:

50

Bis zu welchem Index soll die for-Schleife summieren?

100

Der berechnete Wert nach 100 Schritten: 5.184705527773213E21

Der berechnete Wert nach 118 Schritten mit Abbruch nach Genauigkeit: 5.184705528587081E21

Verbesserter Wert fuer negative x-Werte nach 118 Schritten: 5.184705528587081E21

Wert der Math-Bibliothek: 5.184705528587072E21

Fragen 1 *Berechnung der Exponentialfunktion*

- Welche Indizes besitzen das erste und das letzte Element eines Feldes in Java?
- Wieso ist das Verfahren für negative x -Werte ungeeignet? Erklären Sie.

Aufgabe 2 Mittelwerte

Erstellen Sie ein Java-Programm, das beliebig viele natürliche Zahlen ($a \in \mathbb{N}$) einliest und deren arithmetischen und geometrischen Mittelwert berechnet.

Das arithmetische Mittel A_N von N Zahlen a_1, \dots, a_N berechnet sich zu

$$A_N(a_1, \dots, a_N) = \frac{1}{N} \sum_{i=1}^N a_i = \frac{1}{N} (a_1 + a_2 + \dots + a_N).$$

Das geometrische Mittel G_N von N positiven Zahlen a_1, \dots, a_N ergibt sich zu

$$G_N(a_1, \dots, a_N) = \sqrt[N]{\prod_{i=1}^N a_i} = \sqrt[N]{a_1 \cdot a_2 \cdot \dots \cdot a_N}.$$

Der Benutzer soll so lange Zahlen eingeben können, bis er die Eingabe durch eine Zahl $a \leq 0$ abbricht. Gehen Sie beim Erstellen des Programms folgendermaßen vor:

- Erstellen Sie eine öffentliche Klasse `Mittelwerte` mit einer `main`-Methode. Definieren Sie sich hier zunächst die ganzzahligen Variablen `N`, `summe` und `produkt` zum späteren Speichern der aktuellen Anzahl, der Summe und des Produkts der bisher eingegebenen Zahlen.
- Fordern Sie den Benutzer mit entsprechendem Text dazu auf, die erste natürliche Zahl einzugeben und speichern Sie die Eingabe in einer passenden Variable.
- Schreiben Sie eine `while`-Schleife zur weiteren Berechnung und Eingabe. Die Schleifenbedingung soll überprüfen, ob die zuletzt eingegebene Zahl positiv ist. Ist dies erfüllt, so sollen innerhalb der Schleife die Variablen `N`, `summe` und `produkt` aktualisiert und die nächste Zahl eingelesen werden.

Achten Sie darauf, dass die zu aktualisierenden Variablen vor Eintritt in die Schleife korrekt initialisiert wurden (d.h. sinnvolle Anfangswerte besitzen).

- Berechnen Sie das arithmetische und geometrische Mittel aus der Summe bzw. dem Produkt der eingegebenen Werte.

Hinweis: Allgemein erhalten Sie den Wert x^y durch die Methode `Math.pow(x, y)`.

Beispielergebnisse:

- $A_N(1, 2, 3, 4, 5, 6) = 3.5$
- $G_N(1, 2, 3, 4, 5, 6) \approx 2.994$

Hinweis: Sollten Sie für das arithmetische Mittel immer den Wert 0 erhalten, so könnte es sich lohnen, den Ausdruck $(1/N)$ genauer zu betrachten. Warum erhält man unter Umständen einen anderen Wert als bei $(1.0/N)$?

Aufgabe 3 Statistik

Erstellen Sie mit Java ein Programm mit dem Namen `Statistik`, welches für eine Folge von n reellen Zahlen x_0, \dots, x_{n-1} das *Minimum*

$$x_{\min} := \min\{x_0, x_1, \dots, x_{n-1}\}, \quad (1)$$

das *Maximum*

$$x_{\max} := \max\{x_0, x_1, \dots, x_{n-1}\}, \quad (2)$$

den *Mittelwert*

$$\bar{x} = \sum_{i=0}^{n-1} \frac{x_i}{n} \quad (3)$$

sowie die *Streuung*

$$s = \sqrt{\sum_{i=0}^{n-1} \frac{(x_i - \bar{x})^2}{n-1}} \quad (4)$$

berechnet. Gehen Sie dabei wie folgt vor:

- Lesen Sie in der `main`-Methode eine ganze Zahl n von der Konsole ein, und speichern Sie diese in einer passenden Variable ab. Erzeugen Sie anschließend ein Feld mit dem Namen `daten` der Länge n welches Gleitkommazahlen abspeichert.
- Lesen Sie mit einer `for`-Schleife nacheinander n Gleitkommazahlen von der Konsole ein, und speichern Sie die eingelesenen Zahlen in den Feldkomponenten des Feldes `daten` ab.

Hinweis: Ist `feld` ein Feld und `i` eine ganzzahlige Variable, so bezeichnet der Ausdruck `feld[i]` die Feldkomponente mit dem Index `i`. Achtung: Vergewissern Sie sich zunächst, welchen Index der erste Feldeintrag besitzt.

- Bestimmen Sie das *Minimum* x_{\min} der eingelesenen Zahlen. Erstellen Sie dazu zunächst eine Variable `minimum` vom Gleitkommatyp, und initialisieren Sie diese mit dem Wert der ersten Feldkomponente. Durchlaufen Sie anschließend das Feld `daten` mit einer `for`-Schleife. Prüfen Sie bei jeder Feldkomponente, ob ihr Wert kleiner als der Wert der Variable `minimum` ist. Falls ja, so überschreiben Sie den Wert der Variable mit dem Wert der entsprechenden Feldkomponente.

Bestimmen Sie in ähnlicher Weise auch das *Maximum* x_{\max} der eingelesenen Zahlen. Geben Sie das Minimum und das Maximum auf der Konsole aus.

- Berechnen Sie den *Mittelwert* \bar{x} der eingelesenen Zahlen. Erzeugen Sie dazu eine Variable `mittelwert` vom Gleitkommatyp, und initialisieren Sie diese mit 0. Durchlaufen Sie anschließend das Feld `daten` mit einer `for`-Schleife, und addieren Sie gemäß der obigen Formel den Wert jeder einzelnen Feldkomponente geteilt durch die Anzahl n zur Variable `mittelwert` hinzu.

Berechnen Sie auch das Quadrat der Streuung s^2 nach der entsprechenden Formel. Geben Sie den Mittelwert und die Streuung auf der Konsole aus.

Hinweis: Die Quadratwurzel kann mit der Methode `Math.sqrt()` berechnet werden.

Musterlösung: Testen Sie Ihr Programm für die Zahlenfolge 1, 2, 3, 4, 5. Sie sollten ein Minimum von 1, ein Maximum von 5, einen Mittelwert von 3 und eine Streuung von etwa 1.5811 erhalten.