Institut für Angewandte und
Numerische Mathematik

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

14.11.2022

Einstieg in die Informatik und Algorithmische Mathematik

Aufgabenblatt 5

Bearbeitungszeitraum: 28.11.2022 – 09.12.2022

Aufgabe 1 *Banksparplan*

Bei einem Banksparplan wird zu Beginn des ersten Jahres ein einmaliger Anlagebetrag K_0 eingezahlt. In den folgenden Jahren wird jeweils zu Jahresbeginn ein fester Betrag E hinzugezahlt. Über den gesamten Anlagezeitraum hinweg werde das Guthaben mit einem gleichbleibendem Zinsfaktor q mit $0.01 \leq q \leq 0.08$ verzinst.
Das Kapital K_1 nach einem Jahr ergibt sich dabei gemäß

$$K_1 = K_0 \cdot (1 + q).$$

Für $n \geq 2$ ergibt sich das Kapital zu Ende des n-ten Jahres aus

$$K_n = (K_{n-1} + E) \cdot (1 + q).$$

Erstellen Sie ein Java-Programm, welches den Auszahlungsbetrag am Ende des N-ten Jahres für $N \geq 1$ und die angehäuften Zinsen ermittelt. Gehen Sie dazu folgendermaßen vor:

(a) Erstellen Sie eine Klasse `Zinsen` mit einer `main`-Methode. Definieren Sie zunächst **`double`**-Variablen für den Erstanlagebetrag K_0, die jährliche Einzahlung E und den Zinsfaktor q, sowie eine **`int`**-Variable für die Anlagedauer N in Jahren. Lesen Sie anschließend die jeweiligen Werte vom Benutzer ein. Stellen Sie durch eine **`if`**-Abfrage sicher, dass jeweils sinnvolle Werte eingegeben wurden.

(b) Ermitteln Sie mittels einer **`for`**-Schleife das Kapital zum Ende eines jeden Jahres und den während des Jahres erzielten Zinsertrag. Geben Sie für jedes Jahr während des Anlagezeitraumes beide Werte in der Form (hier mit Beispielwerten)

```
...
Kapital nach 3 Jahren    =  1207.50
Zinsertrag im 3-ten Jahr =    57.50
...
```

auf dem Bildschirm aus.

(c) Geben Sie zuletzt den Auszahlungsbetrag nach N Jahren und die über den gesamten Zeitraum angehäuften Zinsen aus.

(d) Testen Sie Ihr Programm für $K_0 = 10000$, $E = 2000$, $q = 0.05$ und $N = 5$. Die Endauszahlung muss dann rund 21814.08 betragen und der Zinsertrag $= 3814.08$ sein. Außerdem muss der Zinsertrag im 2. Jahr 625 betragen und das Kapital nach 4 Jahren auf 18775.31 angewachsen sein.

Hinweis: Alle Angaben sind auf zwei Nachkommastellen gerundet und in Euro angegeben.

Aufgabe 2 *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 den Datentyp **`double`** zugrunde. Gehen Sie dazu folgendermaßen vor:

- Erstellen Sie eine öffentliche Klasse `Exponential` mit einer `main`-Methode. Hier soll der Benutzer zunächst dazu auffordert 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 dem 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 aus, bei welchem N das Ergebnis erreicht wurde und wie es lautet.

  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 Ergebnis, das wie im vorletzten Punkt beschrieben berechnet wurde, 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
```

Aufgabe 3 (Pflichtaufgabe) *Vektorrechnung*

Erstellen Sie ein Java–Programm, das das Rechnen mit Vektoren im $\mathbb{R}^n$ ermöglicht.

- Die *Addition* und *Subtraktion* zweier Vektoren $\boldsymbol{x} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}] \in \mathbb{R}^{\boldsymbol{n}}$ und $\boldsymbol{y} = [\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}] \in \mathbb{R}^{\boldsymbol{n}}$ ist definiert gemäß

$$\begin{aligned} \boldsymbol{x} + \boldsymbol{y} &:= \big[x_1 + y_1, \ldots, x_n + y_n\big], \\ \boldsymbol{x} - \boldsymbol{y} &:= \big[x_1 - y_1, \ldots, x_n - y_n\big]. \end{aligned}$$

- Die *komponentenweise Multiplikation* bzw. *Division* zweier Vektoren $\boldsymbol{x}$ und $\boldsymbol{y}$ ist definiert durch

$$\begin{aligned} \boldsymbol{x} * \boldsymbol{y} &:= \big[x_1 y_1, \ldots, x_n y_n\big], \\ \boldsymbol{x}/\boldsymbol{y} &:= \big[x_1/y_1, \ldots, x_n/y_n\big], \quad \text{falls } y_i \neq 0 \text{ für } i = 1, \ldots, n. \end{aligned}$$

- Die *skalare Multiplikation* eines Vektors $\boldsymbol{x}$ mit einer reellen Zahl λ ist gegeben durch

$$\lambda \cdot \boldsymbol{x} := \big[\lambda x_1, \ldots, \lambda x_n\big].$$

- Das *Skalarprodukt* zweier Vektoren $\boldsymbol{x}$ und $\boldsymbol{y}$ ist gegeben durch

$$\boldsymbol{x} \cdot \boldsymbol{y} := \sum_{i=1}^{n} x_i y_i.$$

Erstellen Sie ein Java–Programm, das die folgenden Aufgaben erledigt:

(a) Stellen Sie die benötigten Vektoren in Ihrem Java–Programm mit Hilfe von eindimensionalen Feldern dar. Lesen Sie hierzu zuerst die Dimension `n` der Felder ein. Die Variable `n` ist hierbei, wie üblich bei einer Variablen die eine Felddimension angibt, vom Typ `int`. Deklarieren Sie nun die eindimensionalen Felder `x` und `y` vom Typ `double` und mit Länge `n`. Deklarieren Sie weiterhin für alle Rechenoperationen weitere, dem Ergebnis der Operation entsprechende Felder bzw. Variablen (Das Ergebnis der Operation $x + y$ soll zum Beispiel später in einem Feld `sum` der Länge `n` gespeichert werden).

(b) Lesen Sie die zwei Vektoren `x` und `y` sowie eine reelle Zahl `l` ein und führen Sie die oben angegebenen Operationen durch. Speichern Sie die Ergebnisse der Operationen jeweils in einem der weiteren zuvor deklarierten Vektoren bzw. einem Skalar ab. Geben Sie das Ergebnis jeweils mit einem begleitenden Text auf dem Bildschirm aus. Die komponentenweise Division soll nur durchgeführt werden, falls alle Komponenten des Divisors ungleich Null sind, andernfalls soll eine Fehlermeldung ausgegeben werden. Hierfür können Sie eine Variable einführen, die zu Beginn des Programms den Wert 0 (oder `false`) erhält und auf 1 (`true`) gesetzt wird, falls ein Element des Divisors Null ist. Der Wert dieser Variablen kann dann vor der Ausgabe einer entsprechenden Fehlermeldung abgeprüft werden. Testen Sie Ihr Programm für verschiedene Werte.

Achtung: In dieser Aufgabenstellung werden die Indizes von 1 bis n verwendet, wie dies in der Mathematik üblich ist. In Java vereinbarte Felder sind dagegen von 0 bis $n - 1$ indiziert.
Beispiele: Operationen mit Gleitkommazahlen vom Typ `double` werden durch Rundungsfehler in der 16. Dezimalstelle verfälscht. Prüfen Sie, wie sich dies in den Berechnungen dieser Aufgabe auswirkt. Betrachten Sie z.B. das Skalarprodukt und verwenden Sie die beiden folgenden Beispiele:

$$x = \begin{pmatrix} 1 \cdot 10^8 \\ 2 \\ 3 \cdot 10^9 \\ 2 \end{pmatrix}, y = \begin{pmatrix} 6 \cdot 10^{13} \\ 4 \\ -2 \cdot 10^{12} \\ -3 \end{pmatrix}$$

Wie man leicht nachrechnen kann, ist das exakte Ergebnis des Skalarpodukts $x \cdot y = +2$. Vergleichen Sie es mit dem berechneten Ergebnis und begründen Sie die Differenz!

$$x = \begin{pmatrix} 12345678905 \\ 12345678904 \end{pmatrix}, y = \begin{pmatrix} 12345678905 \\ -12345678906 \end{pmatrix}$$

Wie man aufgrund der 3. binomischen Formel $(a - b) \cdot (a + b) = a^2 - b^2$ sieht, ist das exakte Ergebnis des Skalarprodukts $x \cdot y = 1$. Erklären Sie die Differenzen zwischen dem berechneten und dem exakten Ergebnis!

Fragen 3 *Vektorrechnung*

- Was ist der Unterschied zwischen $==$ und $=$?
- Welchen Wert hat diese Zahl $A2E_{16}$ (Hexadezimalsystem) im Dezimalsystem?
- Wann geschehen Typumwandlungen in Java implizit?