Institut für Angewandte und
Numerische Mathematik

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

20.10.2023

Einstieg in die Informatik und Algorithmische Mathematik

Aufgabenblatt 2

Bearbeitungszeitraum: 06.11.2023 – 17.11.2023

Aufgabe 1 *Verkaufsgebühren*

Beim Verkauf eines Gegenstandes über ein Auktionshaus sind abhängig vom erzielten Verkaufspreis Gebühren an das Auktionshaus zu entrichten. Die Gebühren lassen sich nach den folgenden Regeln berechnen:

(a) Bei einem Verkaufspreis bis zu 500 Euro sind pauschal 50 Euro Gebühr zu entrichten.

(b) Bei einem Verkaufspreis zwischen 500 und 2000 Euro ist ein linear ansteigender Prozentanteil zwischen 10% und 20% als Gebühr zu entrichten. Die Gebühr berechnet sich für $500 \leq$`preis`≤ 2000 gemäß der Formel

```
gebuehr=((2000-preis)+2*(preis-500))*preis/15000.
```

(c) Für Verkaufspreise über 2000 Euro sind 20% Gebühren fällig.

Erstellen Sie ein Java–Programm, welches die bei einem Verkauf fälligen Gebühren ermittelt.

- Erstellen Sie eine öffentliche Klassen mit dem Namen `Verkauf`. Importieren Sie die Klassenbibliothek `java.util.*`, und erstellen Sie in der Klasse `Verkauf` eine `main`–Methode. Fügen Sie in die `main`–Methode die folgenden Quelltextzeilen ein:

  ```
  Locale.setDefault(Locale.US);
  Scanner sc = new Scanner(System.in);
  ```

- Legen Sie eine geeignete Anzahl an Variablen an, in denen Gleitkommazahlen abgespeichert werden können. Fordern Sie den Benutzer nun auf den Verkaufspreis einzugeben und speichern Sie diesen ab.

- Ermitteln Sie die Gebühren in Abhängigkeit von der Höhe des Kaufpreises. Unterscheiden Sie dabei die oben genannten drei Fälle.

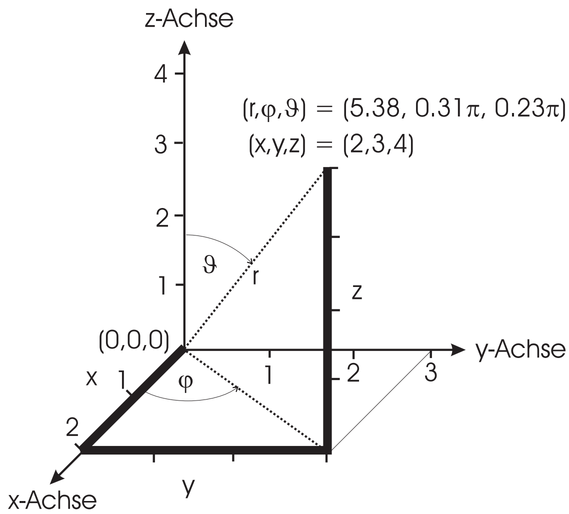
- Ermitteln Sie den effektiven Erlös des Verkaufes (=Verkaufspreis abzüglich der Verkaufsgebühren) und geben Sie die Ergebnisse mit einem geeigneten Text auf dem Bildschirm aus.

Musterlösung:

```
Bitte Verkauspreis in Euro eingeben: 499
Die Verkaufsgebuehr betraegt 50 Euro.
Das Erloes betraegt 449 Euro.
```

Aufgabe 2 (Pflichtaufgabe) *Kugelkoordinaten*

Alternativ zu kartesischen Koordinaten (x, y, z) kann ein Punkt P im dreidimensionalen Anschauungsraum auch in Kugelkoordinaten der Form (r, φ, ϑ) angegeben werden. Dabei bezeichnet r den Abstand des Punktes vom Nullpunkt des Koordinatensystems, φ den Winkel im Bogenmaß, den der Ortsvektor von $(x, y, 0)$ mit der x–Achse einschließt, und ϑ den Winkel, den der Ortsvektor von P mit der z–Achse einschließt. Legt man für die Kugelkoordinaten die Wertebereiche $0 \leq r < \infty,\ -\pi < \varphi \leq \pi,\ 0 \leq \vartheta \leq \pi$ fest, so erhält man eindeutig[1] alle Punkte des Raumes. Den Zusammenhang zwischen kartesischen und Kugelkoordinaten erkennt man an folgender Grafik:



Wir betrachten zwei Programme `kugel2kart` und `kart2kugel`, die jeweils von der einen Form der Darstellung im Anschauungsraum in die andere umwandeln. Dabei gilt für die Umwandlung von Kugelkoordinaten in kartesische Koordinaten:

$$x = r \sin(\vartheta) \cos(\varphi)\ ,$$
$$y = r \sin(\vartheta) \sin(\varphi)\ ,$$
$$z = r \cos(\vartheta)$$

und für die Umwandlung von kartesischen Koordinaten in Kugelkoordinaten:

$$r = \sqrt{x^2 + y^2 + z^2}\ ,$$
$$\varphi = \arctan\left(\frac{y}{x}\right) \quad \text{falls} \quad x \neq 0\ ,$$
$$\vartheta = \arccos\left(\frac{z}{r}\right) \quad \text{falls} \quad r \neq 0\ .$$

[1] mit der Festlegung $\varphi = 0$ für Punkte auf der z–Achse und $(0, 0, 0)$ für den Koordinatenursprung

Das Programm `kart2kugel` zur Umrechnung von kartesischen in Kugelkoordinaten finden Sie im Ordner Praktikum auf den Ilias-Seiten zur Vorlesung. Schreiben Sie in Anlehnung an dieses Programm nun `kugel2kart` für die umgekehrte Berechnung von Kugelkoordinaten in kartesische Koordinaten. Es sollen drei Gleitkommazahlen, welche den Punkt im Kugelkoordinaten beschreiben, eingelesen, dann umgerechnet und anschließend in kartesischen Koordinaten wieder ausgegeben werden. Gehen Sie dabei wie folgt vor:

- Erstellen Sie zunächst eine öffentliche Klasse und eine main-Methode. Importieren Sie dann die Klassenbibliothek `java.util.*` und bereiten Sie Ihr Programm darauf vor Fließkommazahlen einzulesen (vgl. Aufgabe 3).
- Legen Sie anschließend alle benötigten Variablen als Gleitkommazahlen an. Überlegen Sie sich zunächst, wie viele Variablen insgesamt benötigt werden.
- Geben Sie eine klare Anleitung auf der Konsole aus, welche dem Benutzer die Funktion Ihres Programmes erläutert und ihn dazu auffordert das Zahlentripel der Kugelkoordinaten nacheinander einzugeben.

  Hinweis: Bei der Eingabe von Winkeln im Bogenmaß sollen die Winkel jeweils als Vielfache von π angegeben werden. Das heißt, bei einem Beispielwert von $\pi/2$ soll lediglich 0.5 eingegeben werden müssen. Verwenden Sie `Math.PI` als Approximation für π.
- Berechnen Sie die neuen Koordinaten in kartesischer Form nach der oben genannten Berechnungsvorschrift. Geben Sie abschließend die neu berechneten Koordinaten mit einem passenden Text auf der Konsole aus.

  Hinweis: Die entsprechenden mathematischen Funktionen zur Umrechnung in Java lauten: `Math.sin`, `Math.cos`, `Math.sqrt`, `Math.atan` und `Math.acos`. Dabei muss das Argument einer Standardfunktion in Klammern gesetzt werden. Zum Beispiel:

  ```
  double x;
  x = Math.sin(2.0 * Math.PI);
  ```
- Testen Sie Ihr Programm mit den folgenden Koordinaten: $(1, \pi/2, \pi/2)$, $(1, \pi, \pi/2)$, $(16, \pi/2, 0)$

Fragen:

- Nennen Sie eine typische Anwendungsmöglichkeit für Polarkoordinaten.
- Erklären Sie für was das *-Zeichen beim Importieren der Klassenbibliothek steht.

Aufgabe 3 *Kubische Gleichungen*

Die Lösungen einer kubischen Gleichung der Form

$$ax^3 + bx^2 + cx + d = 0, \tag{1}$$

wobei a, b, c und d reelle Zahlen sind, und a ungleich Null ist, können direkt berechnet werden. Man berechnet dazu zunächst die Hilfsgrößen

$$\begin{aligned} p &:= \frac{3ac - b^2}{9a^2}, \\ q &:= \frac{1}{2}\left(\frac{2b^3}{27a^3} - \frac{bc}{3a^2} + \frac{d}{a}\right), \\ D &:= q^2 + p^3 \; . \end{aligned} \tag{2}$$

Falls $p < 0$ und $D \leq 0$ gilt, berechnet man die Hilfsgrößen

$$\begin{aligned} r &:= \operatorname{sgn}(q)\sqrt{-p} \; , \\ s &:= \arccos\left(\frac{q}{r^3}\right) , \end{aligned} \tag{3}$$

und erhält die drei Lösungen x_1, x_2 und x_3 der kubischen Gleichung durch

$$\begin{aligned} x_1 &:= -2r\cos\left(\frac{s}{3}\right) - \frac{b}{3a}, \\ x_2 &:= 2r\cos\left(\frac{\pi - s}{3}\right) - \frac{b}{3a}, \\ x_3 &:= 2r\cos\left(\frac{\pi + s}{3}\right) - \frac{b}{3a}. \end{aligned} \tag{4}$$

Dabei bezeichnet $\operatorname{sgn}(q)$ das *Vorzeichen* q. Im Fall $q < 0$ gilt $\operatorname{sgn}(q) = -1$, im Fall $q = 0$ gilt $\operatorname{sgn}(q) = 0$, und im Fall $q > 0$ gilt $\operatorname{sgn}(q) = 1$.

Erstellen Sie ein Java-Programm, welches die Koeffizienten a, b, c und d einer kubischen Gleichung von der Konsole einliest, und deren Lösungen x_1, x_2 und x_3 berechnet, sofern $p < 0$ und $D \leq 0$ gilt. Gehen Sie dabei folgendermaßen vor:

- Erstellen Sie eine öffentliche Klasse mit dem Namen `KubischeGleichung`, und erstellen Sie in dieser Klasse die `main`-Methode des Programms. Importieren Sie die Klassenbibliothek `java.util.*` und fügen Sie die folgenden Befehlszeilen in die `main`-Methode ein:

  ```
  Locale.setDefault(Locale.US);
  Scanner sc = new Scanner(System.in);
  ```

  Dies bereitet Ihr Programm für das Einlesen von Daten vor.

- Erzeugen Sie für jeden Koeffizienten a, b, c und d eine Variable von geeignetem Typ. Lesen Sie anschließend mit der Methode `sc.nextDouble()` vier Fließkomma-Zahlen von der Konsole ein, und speichern Sie diese in den Variablen ab.

- Berechnen Sie aus den eingelesenen Zahlen die Hilfsgrößen p, q und D und speichern Sie die berechneten Werte in drei Variablen als Gleitkommazahlen ab. Verwenden Sie für die Variablen die Namen `p`, `q` und `D`. Geben Sie die berechneten Werte anschließend auf der Konsole aus.

  Hinweis:

  - Ausdrücke wie a^3 können durch $a \cdot a \cdot a$ berechnet werden.
  - Stellen Sie Zahlkonstanten wie 1, 2, 3 und 27 stets als `1.0`, `2.0`, `3.0` und `27.0` in ihrem Quelltext dar.
  - Verwenden Sie geeignete Klammerungen.

- Fügen Sie folgende Befehlszeile in ihr Programm ein.

```
if ( p >= 0 || D > 0 ) return;
```

  Dies führt dazu, dass Ihr Programm an dieser Stelle beendet wird, sofern $p \geq 0$ oder $D > 0$ gilt.

  Berechnen Sie nach dieser Befehlszeile die Hilfsgrößen r und s, und speichern Sie die berechneten Werte in zwei Variablen als Gleitkommazahlen mit geeigneten Namen ab.

  Hinweis:

  - Die Funktion sgn heißt unter Java `Math.signum()`.
  - Die Wurzelfunktion $\sqrt{\cdot}$ heißt unter Java `Math.sqrt()`.
  - Die Funktion $\arccos$ heißt unter Java `Math.acos()`.

- Berechnen Sie die Lösungen x_1, x_2 und x_3 der kubischen Gleichung. Speichern Sie die berechneten Lösungen in drei Variablen vom Typ **`double`** ab, und geben Sie die Lösungen anschließend auf der Konsole aus.

  Hinweis:

  - Die Cosinusfunktion $\cos$ heißt unter Java `Math.cos()`.
  - Die Zahl π heißt unter Java `Math.PI`.

- Testen Sie Ihr Programm, indem Sie folgende Gleichungen lösen:

  $$x^3 - 1.4x^2 + 0.59x - 0.07 = 0 \qquad \text{Lösungsmenge: } \{0.2, 0.5, 0.7\}$$
  $$5x^3 - x^2 - 0.65x - 0.05 = 0 \qquad \text{Lösungsmenge: } \{-0.2, -0.1, 0.5\}$$

  Welche Lösungen besitzt die folgende Gleichung?

  $$x^3 - 2x^2 + 1.27x - 0.252 = 0$$

  Warum kann das Programm die Lösungen der folgenden Gleichung nicht berechnen?

  $$x^3 - 0.5x^2 + 0.3x + 1 = 0$$