

Übungsblatt 02

4 Aufgaben, 20 Punkte

Programmierung, Datenstrukturen und Algorithmen (inf030) Wintersemester 2022/2023
 Carl von Ossietzky Universität Oldenburg, Fakultät II, Department für Informatik

Dr. C. Schönberg

Ausgabe: 2022-10-28 12:00

Abgabe: 2022-11-04 10:00

4/4

Aufgabe 1: *Ganze Zahlen*

(1+1+2 Punkte)

Der Astronom Stefan Sterngucker hat mit seinem Teleskop zwei Sterne s_1 und s_2 beobachtet, die direkt hintereinander liegen. Durch Messung der Positionsveränderungen der Sterne über mehrere Monate konnte er deren Entfernung bestimmen:

s_1 ist ca. $d_1 = 8,000,000,000,000,000,000\text{ m} = 8 \cdot 10^{18}\text{ m}$ von der Erde entfernt.

s_2 liegt hinter s_1 und ist zusätzliche $d_{1-2} = 3,000,000,000,000,000,000\text{ m} = 3 \cdot 10^{18}\text{ m}$ von s_1 entfernt.

Um die Entfernung d_2 von s_2 zur Erde zu berechnen schreibt Stefan Sterngucker folgendes Java-Programm:

```

1 public class StarDistance {
2     public static void main(String[] args) {
3         long d1 = 8000000000000000000L;
4         long d12 = 3000000000000000000L;
5         long d2 = d1 + d12;
6         System.out.println(d2);
7     }
8 }
  
```

Mit dem Ergebnis des Programms ist er allerdings nicht sonderlich zufrieden.

- Was gibt das Programm aus?
- Warum gibt das Programm nicht das erwartete Ergebnis aus (11,000,000,000,000,000,000)?
- Wie lässt sich das Problem beheben?

4/4

Aufgabe 2: Wertekombination

(1 + 1 + 2 Punkte)

Schreiben Sie ein Java-Programm, ...

- ... in dem Sie zunächst je eine Variable von den Typen **int**, **long**, **float**, **double** und **char** anlegen. Weisen Sie diesen Variablen jeweils einen beliebigen Wert vom korrekten Typ zu.
- ... in dem Sie anschließend eine Variable **sumDouble** vom Typ **double** anlegen. Ihr Programm soll die Summe aller Variablenwerte aus Aufgabenteil **a)** berechnen und der Variablen **sumDouble** zuweisen.
- ... in dem Sie anschließend eine Variable **sumInt** vom Typ **int** anlegen. Ihr Programm soll die Summe aller Variablenwerte aus Aufgabenteil **a)** berechnen und der Variablen **sumInt** zuweisen.

Hinweis: Verwenden Sie explizite oder implizite Typumwandlung, wo nötig. Nehmen Sie in Aufgabenteil **c)** den nötigen Präzisionsverlust in Kauf.

```

1 package uebung02;
2
3 public class Uebung022 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int a = 20;
8         long b = 1234567891234567890L;
9         float c = 10.6F;
10        double d = 1.5E300;
11        char e = 'A';
12
13        double sumDouble = a + b + c + d + e;
14
15        System.out.println(sumDouble);
16
17        int sumInt = a + (int) b + (int) c + (int) d + e;
18
19        System.out.println(sumInt);
20    }
21
22 }
23
24 }
25
26

```

Console Output:

```

<terminated> Uebung022 [Java Application] /Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java (01.11.2022, 08:29:14 - 08:29:16) [pid: 18459]
1.5E300
1076082480

```

5/6 (Es wäre gut, wenn der abgegebene Code mit dem im pdf übereinstimmen würde, ich habe mir nur den abgegebenen angeschaut)

Aufgabe 3: Kostenberechnung

(6 Punkte)

Es fehlt das Abziehen der 5 kostenlosen Exemplare

Ein Musik-Anbieter im Internet hat folgendes Geschäftsmodell: Jeder Nutzer kann sich zunächst fünf Musikstücke kostenlos herunterladen. Für die nächsten 30 Stücke muss er jeweils 20 Cent bezahlen und danach für jeden weiteren Download 15 Cent.

Schreiben Sie ein Java-Programm, das es einem Nutzer ermöglicht, die Anzahl an insgesamt gewünschten Musikstücken anzugeben und das daraufhin die insgesamt entstehenden Kosten berechnet und diese auf den Bildschirm ausgibt.

Beispielablauf des Programms (Benutzereingaben in <>):

Anzahl Downloads (>=0): <41>

Kosten = 6.9 EURO

```

1 package uebung02;
2
3 import java.util.Scanner; // Scanner Klasse einbinden
4 public class Uebung023 {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         //Scanner erzeugen
9         Scanner scanner = new Scanner(System.in);
10
11         System.out.println("Geben Sie die Anzahl an insgesamt gewünschten Musikstücken an:");
12         //Zahl einlesen
13         int anzahl = scanner.nextInt();
14
15         //Kosten gesamt zu Anfang
16         double kostengesamt = 0;
17
18         //Fällefrage für Anzahl > 5
19         if (anzahl > 5) {
20
21             double anzahlkosten = anzahl - 5;
22
23             if ( anzahlkosten <= 30) {
24                 //Berechnung Kosten für Anzahl < 30
25                 kostengesamt = anzahlkosten * 0.2;
26             }
27
28             else if( anzahlkosten > 30) {
29                 //Berechnung Kosten für Anzahl > 30
30                 double teilkosten1 = 30 * 0.2;
31                 double anzahlrestkosten = anzahlkosten - 30;
32                 double teilkosten2 = anzahlrestkosten * 0.15;
33
34                 kostengesamt = teilkosten1 + teilkosten2;
35             }
36
37         }
38
39         else {
40             // Kosten für Anzahl <5
41             kostengesamt = 0;
42         }
43
44
45         //Scanner schließen
46         scanner.close();
47
48         System.out.println("Für " + anzahl + " Downloads müssen Sie " + kostengesamt + " € bezahlen.");
49     }
50 }
51
52
53

```

Problems @ Javadoc Declaration Console X

<terminated> Uebung023 [Java Application] /Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java (01.11.2022, 09:06:07 - 09:06:10) [pid: 18763]

Geben Sie die Anzahl an insgesamt gewünschten Musikstücken an:

400

Für 400 Downloads müssen Sie 60.75 € bezahlen.

6/6

Aufgabe 4: Ratespiel

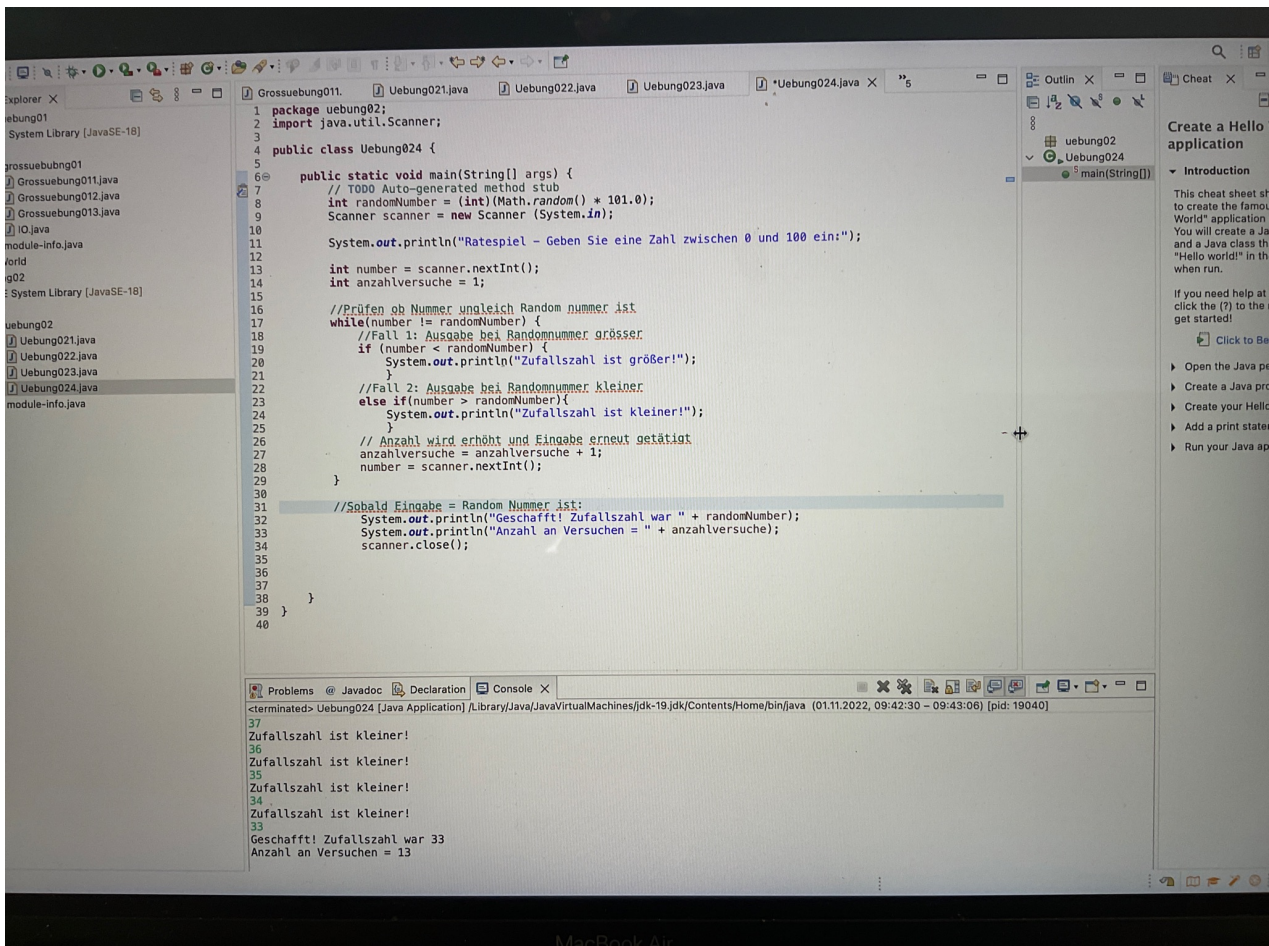
(6 Punkte)

Programmieren Sie ein Ratespiel. Das Programm generiert anfangs eine **int**-Zufallszahl zwischen 0 und 100. Anschließend soll der Benutzer versuchen, diese Zahl zu erraten. Das Programm gibt jeweils aus, ob die vom Benutzer eingegebene Zahl zu groß, zu klein oder korrekt ist. In letzterem Fall soll das Programm stoppen und die Anzahl an Rateversuchen ausgeben. Nutzen Sie zur Zufallszahlenberechnung den Befehl `Math.random`, der **double**-Zufallszahlen zwischen 0.0 (einschließlich) und 1.0 (ausschließlich) generiert:

```
1 int randomNumber = (int)(Math.random() * 101.0); // [0..100]
```

Beispiel für einen Programmablauf (Benutzereingaben in <>):

```
Zahl: <50>
Zufallszahl ist kleiner!
Zahl: <25>
Zufallszahl ist groesser!
Zahl: <37>
Zufallszahl ist groesser!
Zahl: <45>
Zufallszahl ist groesser!
Zahl: <48>
Geschafft! Zufallszahl war 48
Anzahl an Versuchen = 5
```



Übung 2

long $\hat{=}$ 64bit

① a) -7446744073709551616

b) Der Datentyp long hat 64Bit an Speicher für die Darstellung der Zahlen vorhanden. Der größtmögliche Wert der dargestellt werden kann ist 9.223.372.036.854.775.807. Somit kann der Wert 11.000.000.000.000.000.000 nicht mehr dargestellt werden und es kommt zu einem Darstellungfehler.

c) Das Problem lässt sich beheben, indem eine implizite Typumwandlung von long nach double erfolgt. Die Werte werden dann in der schreibweise $1.1 \cdot 10^{19}$ dargestellt.