



Aufgabenblatt 5 Ausgabe: 15.11., Abgabe: 22.11. 24:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 5.1 (Punkte 10+10+10)

Rundung: Wir betrachten ein Gleitkommaformat im Dezimalsystem mit zwei Stellen für den Exponenten und vier Nachkommastellen für die Mantisse. Addieren Sie die beiden Gleitkommazahlen und geben Sie sowohl die Zwischenrechnungen, als auch das normalisierte und gerundete Ergebnis an.

$$8,617 \cdot 10^5 + 9,9443 \cdot 10^7$$

Führen Sie diese Berechnung zweimal, mit unterschiedlichen Rundungsstrategien, durch:

- Mit einmaliger Rundung am Ende nach der Normalisierung.
- Bei der Berechnung werden alle Zahlen, auch die Zwischenergebnisse, auf vier Nachkommastellen gerundet.
- Welches Verfahren ist vorzuziehen? Beziehungsweise, was wäre ein sinnvolles Rundungsverfahren? Begründen Sie Ihre Antwort.

Aufgabe 5.2 (Punkte 10+10+5)

ULP: Um zu untersuchen, wie sich Rundungsfehler bei der Gleitkomma-Arithmetik auswirken, betrachten wir folgendes Programm, das in einer Schleife immer wieder den Wert 0,1, der sich in binärer Gleitkommadarstellung ja nicht exakt darstellen lässt, aufaddiert.

```
1 // Test floating point arithmetic...
2 public class aufg05_2 {
3     public static void main( String args[] ) {
4         // calculate the sum of 1E8 unprecise numbers
5         int n = 0;
6         float sum;
7         float limit = 1.0E8f;
8         for( sum = 0; sum < limit; sum += 0.1 ) {
9             n++;
10            // if ((n % 1000000) == 0) {
11            //     System.out.println("n="+ n + " sum="+ sum + " target="+ (n*0.1));
```

```

12 //     }
13     }
14     System.out.println("sum is " + sum + " compared to " + (n*0.1));
15     }
16 }

```

- (a) Was erwarten Sie (ungefähr) als Ausgabewert des Programms?
- (b) Was passiert tatsächlich? Warum?
 Tipp: Es kann helfen, die auskommentierten Codezeilen 10...12 wieder zu aktivieren.
- (c) Schreiben Sie das Programm so um, dass es den ursprünglich angedachten Zweck erfüllt.

Aufgabe 5.3 (Punkte 15+5)

ISO-8859-1: Entschlüsseln Sie mit Hilfe der Tabellen aus den Vorlesungsfolien (Zeichensatz nach ISO-8859-1) die folgende Zeichenkette. Schreiben Sie dabei auch alle Steuerzeichen mit auf. Die einzelnen Zeichen sind als Hexwerte angegeben.

- (a) 44 69 65 20 4C F6 73 75 6E 67 0A 20 64 69 65 73
 65 72 20 DC 62 75 6E 67 73 61 75 66 67 61 62 65
 0A 20 20 6C 69 65 67 74 20 76 6F 72 20 49 68 6E
 65 6E 21 0A

- (b) Was verrät Ihnen der Text über den Rechner mit dem erstellt worden ist?

Aufgabe 5.4 (Punkte 10+10+5)

Zeichensatzcodierungen: Die ISO-8859-1 Codierung benutzt 8-bit für jedes enthaltene Zeichen. Die direkte Codierung der basic-multilingual Plane von Unicode (Java Datentyp char) verwendet pro Zeichen 16-bit, während die UTF-8 Codierung Vielfache von 8-bit benutzt.

- (a) Wir betrachten einen deutschsprachigen Text mit insgesamt 200 000 Zeichen. Wir nehmen die folgenden Wahrscheinlichkeiten für die Umlaute an, andere Sonderzeichen kommen nicht vor: Ä/ä: 0,54% Ö/ö: 0,30% Ü/ü: 0,65% ß: 0,37%.

Wie viele Bytes belegt dieser Text bei Codierung nach ISO-8859-1, in direkter Unicode Darstellung und in UTF-8?

- (b) Im aktuellen Unicode-Standard sind für die CJK-Symbole (chinesisch, japanisch, koreanisch) neben einem Basis-Zeichensatz noch mehrere Erweiterungen definiert. Wir betrachten einen chinesischen Text mit insgesamt 200 000 Schriftzeichen, der aus den Basis-Zeichen (von U+4E00 bis U+9FEA) und Symbolen der Extension-A (von U+3400 bis U+4DB5) besteht.

Wie viele Symbole sind das? Berechnen Sie das Ergebnis direkt im Hexadezimalsystem aus und verwenden Sie dabei das 16-Komplement für die Subtraktion. Die Lösung muss mit Zwischenschritten abgeben werden.

- (c) Wie viele Bytes belegt der chinesische Text bei direkter Unicode Darstellung und bei Codierung als UTF-8?