



Aufgabenblatt 4

Ausgabe: 09.11., Abgabe: 16.11. 12:00

Gruppe	
Name(n)	Matrikelnummer(n)

Aufgabe 4.1 (Punkte 5+5+5)

Größenvergleich von Gleitkommazahlen: Für den Vergleich von Gleitkommazahlen bietet Java alle sechs Vergleichsoperatoren:

```
a == b    a != b    a > b    a >= b    a < b    a <= b
```

Aufgrund der unvermeidlichen Rundungsfehler bei Gleitkommarechnung ist jedoch Vorsicht bei Verwendung dieser Operatoren geboten. Zum Beispiel liefert

```
double a = 0.1;
double b = 0.3;
System.out.println( (3*a) == b );
```

den Wert false.

- (a) Ein naheliegender Ansatz ist daher, zwei Zahlen als „gleich“ anzusehen, wenn der Absolutwert ihrer Differenz kleiner als eine (vom Benutzer) vorgegebene Konstante ist:

```
final double eps = 1.0E-12;

if (Math.abs( a - b ) <= eps) { // Zahlen fast gleich
    ...
}
```

Welchen offensichtlichen Nachteil hat dieses Verfahren?

- (b) Überlegen Sie sich ein Verfahren zum Vergleich von zwei Gleitkommazahlen, das nicht die absolute (wie oben), sondern eine relative Abweichung berücksichtigt.
- (c) Hat auch dieses Verfahren Nachteile? ... und wenn ja, wie könnte man sie beheben?

Aufgabe 4.2 (Punkte 12+3)

ASCII-Code (ISO-8859-1): Entschlüsseln Sie mit Hilfe der Tabellen aus dem Vorlesungsskript (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 0D 0A 20 4C F6 73 75 6E 67 0D 0A 20 20 64 65 72 0D 0A 20 20 20
 DC 62 75 6E 67 73 61 75 66 67 61 62 65 0D 0A 20 20 20 20 6C 69 65 67 74
 0D 0A 20 20 20 20 20 76 6F 72 0D 0A 20 20 20 20 20 20 49 68 6E 65 6E 21

- (b) Auf welcher „Art“ Rechner wurde der Text erstellt?

Aufgabe 4.3 (Punkte 8+2+5)

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

- (a) Wir betrachten einen deutschsprachigen Text mit insgesamt 800 000 Zeichen. Wir nehmen die folgenden Wahrscheinlichkeiten für die Umlaute an, andere Sonderzeichen kommen nicht vor:
- | Ä/ä | Ö/ö | Ü/ü | ß |
|-------|-------|-------|-------|
| 0,56% | 0,29% | 0,62% | 0,31% |

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

- (b) Wir betrachten einen chinesischen Text mit insgesamt 800 000 Schriftzeichen. Im Unicode-Standard sind für die CJK-Symbole (chinesisch, japanisch, koreanisch) die Bereiche von U+3400 bis U+4DBF und U+4E00 bis U+9FCF reserviert.

Wie viele Symbole sind das?

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

Aufgabe 4.4 (Punkte 5+5+5+5)

Shift-Operationen statt Multiplikation: Ersetzen Sie die folgenden Berechnungen *möglichst effizient* durch eine Folge von Operationen: <<, +, -. Nehmen Sie für die Variablen x und y den Datentyp int (32-bit Zweierkomplementzahl) an.

- (a) $y = 10 * x$
 (b) $y = 30 * x$
 (c) $y = -48 * x$
 (d) $y = 60 * (x + 6)$

Aufgabe 4.5 (Punkte 5+5+10+15)

Logische- und Shift-Operationen: Realisieren Sie, die folgenden Funktionen als *straightline*-Code in Java, das heißt ohne Schleifen, If-Else Abfragen oder den ternären Operator `.. ? .. : ...`. Außerdem dürfen nur einige der logischen und arithmetischen Operatoren benutzt werden:

```
! ~ & ^ | + << >> >>>
```

Alle Eingabeparameter und Rückgabewerte sind jeweils (32-bit) Integerwerte.

- (a) `bitNor(x,y)` Diese Funktion soll das bitweise NOR liefern: $\overline{x_i \vee y_i}$. Als Operatoren dürfen nur `&` und `~` (AND, Negation) benutzt werden.
- (b) `bitXor(x,y)` Diese Funktion soll die XOR-Verknüpfung (Antivalenz) realisieren: $x_i \oplus y_i$. Als Operatoren dürfen nur `&` und `~` (AND, Negation) benutzt werden.
- (c) `rotateRight(x,n)` Die Funktion soll den in Java nicht vorhandenen Rotate-Right Operator für `x` nachbilden. Für das zweite Argument `n` gilt: $0 \leq n \leq 31$.
- (d) `abs(x)` Der Absolutwert (Betrag) von `x`. Welchen Wert liefert ihre Funktion für den Eingabewert -2^{31} ? Beschreiben Sie, wie Ihre Lösung funktioniert.