

Modellierung und Programmierung 1  
Übungsserie 1

Abgabetermin: 27.10.2013, 23:55 Uhr

Grundsätzlich sind Nebenrechnungen anzugeben und Antworten zu begründen.

Einzureichen sind, bei mehreren Dateien als .zip-Archiv:

Lösungen als .pdf-Datei, Programme als Quellcode, Ergebnisdateien.

1. Konvertierung

Vervollständigen Sie (ohne Angabe der Nebenrechnungen) die nachfolgenden Tabellen:

a) Natürliche Zahlen

Basis 2	Basis 10	Basis 16
	1 337	
1 1010 0001 1010		
		D EC 0D ED

b) Rationale Zahlen

Basis 2	Basis 10	Basis 16
	31.875	
1 0111.0101		
		F0.0D

c) Vorzeichenbehaftete Zahlen

Zweierkomplement	Basis 10
	-47
1001 0100	
0101 0110	

d) Berechnen Sie (mit Angabe der Nebenrechnungen)  $b$  bzw.  $x$  so, dass nachfolgende Gleichungen erfüllt sind:

$$(421)_b = (343)_{10}$$

$$(0.\overline{21})_3 = (x)_{10}$$

2. Maschinenzahlen

a) Berechnen Sie die Codierung von 2520.8125 und  $-0.0625$  als `float`-Maschinenzahl.  
(`float`  $\hat{=}$  `single` nach IEEE-Standard)

b) Für wieviel Prozent der im IEEE-`single`-Standard denormalisiert dargestellten Zahlen  $x > 0$  führt die Berechnung von  $\frac{1}{x}$  zu einem Überlauf?

3. Algorithmen und Programme

**Perfekte Zahlen** sind Zahlen, deren Wert gleich der Summe aller ihrer echten Teiler entspricht.

**Abundante Zahlen** sind kleiner als die Summe aller ihrer echten Teiler. So ist z.B.  $6 = 1 + 2 + 3$  eine perfekte Zahl und  $12 < 1 + 2 + 3 + 4 + 6$  abundant.

- 
- a) Entwerfen Sie für den in der Vorlesung vorgestellten Modellrechner ein Programm *perfect(n)* mit

$$perfect(n) = \begin{cases} 2, & \text{falls } n \text{ eine abundante Zahl ist,} \\ 1, & \text{falls } n \text{ eine perfekte Zahl ist,} \\ 0, & \text{sonst.} \end{cases}$$

Achten Sie dabei auf die Effizienz Ihres Programmes. Protokollieren Sie die Belegungsänderung aller im Programm vorhandenen Variablen für *perfect(28)* und interpretieren Sie das Ergebnis Ihres Programms.

- b) Die kleinsten abundanten Zahlen lauten 12, 18, 20, 24, ...  
Gibt es keine ungeraden abundanten Zahlen? Beweis oder Gegenbeispiel!

#### 4. JavaScript

Verändern Sie a) das Hallo-Welt-Applet und b) die dazugehörige HTML-Seite aus der Vorlesung so, dass man mittels Mausklick die Vorder- und Hintergrundfarben vertauschen kann.

Hinweise:

- Als Farben können Sie die Klassenkonstanten der Klasse **Color** verwenden, zum Beispiel: **Color.white** für Weiß und **Color.black** für Schwarz.
- Speichern Sie die aktuelle Vordergrundfarbe und die aktuelle Hintergrundfarbe in dafür vorgesehene Farbvariablen, z.B. **Color vordergrundFarbe = Color.white;**
- Mit **setForeground( Color);** und **setBackground( Color);** können die aktuellen Werte innerhalb der Hauptmethode **paint** gesetzt werden.
- Um die Farben zu wechseln, vertauschen Sie in einer speziellen Methode **set** die Werte der Farbvariablen.
- **onClick** ist die Java Script Anweisung im HTML-Code, mit der man einen Mausklick abfragen kann. Jeder Mausklick ruft dann die Methode **set** zum Farbenwechseln auf.

##### *HalloWeltAppletPlus.java (Grobstruktur)*

```
import java.applet.*;                                     //Applet
import java.awt.*;                                         // Graphics, Color
public class HalloWeltAppletPlus extends Applet
{
    // Vordergrundsfarbe und Hintergrundsfarbe vereinbaren

    public void paint( Graphics g)
    {
        // Vordergrundsfarbe und Hintergrundsfarbe setzen
        // Bildschirmausschrift erzeugen

    }
    public void set()
    {
        // Farben wechseln

        repaint( 100L);
    }
}
```