

# Softwareentwurf und Programmierung – Lösungen

### Aufgabe 1 – Informatikaufgabe, 20 Punkte

Unter einer Fließkommazahl versteht man die Exponentialdarstellung einer reellen Zahl. Diese entspricht der Form:

Vorzeichen \* Mantisse \* Basis ^ Exponent

Dabei bestimmt die Mantisse die Genauigkeit und der Exponent die Größe des Wertebereichs der Zahl.

Bsp. (dezimal):  $-5.4321 \cdot 10^8$

Bei der Addition zweier Fließkommazahlen müssen die Exponenten der beiden Zahlen verglichen werden. Sind diese unterschiedlich, werden sie angeglichen, d.h. normiert. Dabei wird die Mantisse der kleineren der beiden Zahlen solange nach rechts geschoben, bis die kleinere Zahl den gleichen Exponenten wie die größere Zahl hat. Allerdings ist die Anzahl der Ziffern in der Mantisse fest vorgegeben, beispielsweise bei *IEEE 754* 23 Stellen für die Mantisse, dadurch gehen alle nach rechts ‚über den Rand hinaus‘ geschobenen Ziffern verloren.

Bsp. (binär in IEEE 754 Darstellung) mit 5 und 0,2:

V	E	E	E	E	E	E	E	E	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0

Bei der Normalisierung der Exponenten für die Addition ergibt sich für die 0,2:

V	E	E	E	E	E	E	E	E	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

Durch die Angleichung der Exponenten wird die Mantisse bei 0.2 mit fünf Nullen aufgefüllt, dafür fallen hinten aber auch fünf Ziffern durch das Hinausschieben weg. Gerade wenn die Exponenten stark voneinander abweichen, hat man einen besonders großen Auslöschungseffekt. Deshalb ist es sinnvoll, zunächst die kleineren Werte zunächst zu addieren, damit man einen etwas größeren Wert erhält und sich damit den größeren Werten schon annähert.

**Aufgabe 2 – Informatikaufgabe, 20 Punkte**

- a) Vorteile eines Versionskontrollsystems:
- Es kann genau kontrolliert werden, wer wann was geändert hat
  - Alte Versionen können wiederhergestellt werden, also auch falsche oder versehentliche Änderungen sind reversibel
  - Alle Versionen des Projektes sind archiviert
  - Die parallele Arbeit mehrerer Entwickler am selben Projekt wird vereinfacht (Blockieren und Zusammenführen von Dateien)
  - Es kann verschiedene Entwicklungsstränge, branches, eines Projektes geben
- b) Geeignet für die Überwachung durch Versionskontrollsysteme sind Primärdateien, wie Quellcodes, Rohdaten und Dokumentationsdateien.
- c) Subversion, GIT, CVS, RCS, SCCS, Bazaar, Darcs, Fossil, GNU arch, Mercurial, Monotone

**Aufgabe 3 – Algorithmus Aufgabe, 60 Punkte**

```
public class Gesamtzahlung {

    private int anzahl;
    private double zahlungen [];
    private int zeitpunkte [];

    public Gesamtzahlung(double [] zahlungen, int [] zeitpunkte){
        this.anzahl = zahlungen.length;
        this.zahlungen = new double [this.anzahl];
        this.zeitpunkte = new int [this.anzahl];
        for(int i = 0; i < this.anzahl; i++){
            this.zahlungen[i] = zahlungen[i];
            this.zeitpunkte[i] = zeitpunkte[i];
        }
    }

    public double berechneBarwert(){
        double barwertSum = 0;
        for(int i = 0; i < this.anzahl; i++){
            double barwertEinzel = this.zahlungen[i] / Math.pow(1.01,
                this.zeitpunkte[i]);
            barwertSum += barwertEinzel;
        }
        return barwertSum;
    }

    public double berechneBarwertV2(){
        double [] barwerte = new double[this.anzahl];
        for(int i = 0; i < this.anzahl; i++){
            barwerte[i] = this.zahlungen[i] / Math.pow(1.01,
                this.zeitpunkte[i]);
        }
        this.sortBarwerte(barwerte);
        double barwertSum = 0;
        for(double wert : barwerte){
            barwertSum += wert;
        }
    }
}
```

```
        return barwertSum;
    }

    private void sortBarwerte(double barwerte []){

        for(int i = 0; i < this.anzahl; i++){
            double wert = barwerte[i];
            for(int j = i+1; j < this.anzahl; j++){
                if(barwerte[j] < wert){
                    barwerte[i] = barwerte[j];
                    barwerte[j] = wert;
                    wert = barwerte[i];
                }
            }
        }
    }

    public double berechneKonstanteRate(int m){
        double normbarwert = 0;
        double faktor = 1/1.01;
        double x = faktor;
        for(int i = 1; i <= m; i++){
            normbarwert += x;
            x*= faktor;
        }
        return berechneBarwertV2()/normbarwert;
    }
}
```