

6. Übungsblatt - Informatik 1 - Lösungsbeispiele

Aufgabe 1 (for-Schleife)

Geben Sie ein Java-Programm an, dass den Wert der folgenden Folge bis zu einem vorgegebenen Wert i berechnet. Verwenden Sie eine for-Schleife:

$$\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^i}$$

Diese Folge konvergiert gegen einen Grenzwert für $i \rightarrow \infty$.

Lösungsvorschlag:

```
double summe = 1.0; // erster Term
double einsGeteiltDurch2HochJ = 0.5;
for (int j = 1; j <= i; j++) {
    summe += einsGeteiltDurch2HochJ;
    einsGeteiltDurch2HochJ *= 0.5;
}
```

Das Programm kann noch ein in eine Funktion eingepackt werden, bei der i als Parameter übergeben wird. Die Summe konvergiert gegen 2.

Aufgabe 2 (Schleifen)

Schreiben Sie Ihr Programm aus der vorherigen Aufgabe so um, dass Ihre Schleife abbricht, wenn der Grenzwert der Folge erreicht ist. Allerdings mit den folgenden Einschränkungen:

- Der Grenzwert selbst darf in Ihrem Programm *nicht* zum Abbruch oder sonst irgendwo verwendet werden.
- In der Abbruchbedingung darf die Laufvariable *nicht* vorkommen.
- Die Schleife darf nur über die Abbruchbedingung verlassen werden.

Für die Implementierung ist keine Laufvariable mehr nötig.

Hinweis: Die Elemente der Folgen werden sehr viel kleiner als der Grenzwert.

Lösungsvorschlag:

```
double summe = 1.0; // erster Term
double einsGeteiltDurch2HochJ = 1.0/2.0;
double vorherigeSumme = 0.0;
while (summe != vorherigeSumme) {
    vorherigeSumme = summe;
    summe += einsGeteiltDurch2HochJ;
}
```

```

    einsGeteiltDurch2HochJ *= 1.0/2.0;
}

```

Die Schleife bricht ab, weil irgendwann $\text{summe} + \text{einsGeteiltDurch2HochH} = \text{summe}$ gilt.

Aufgabe 3 (Aktivitätsdiagramm)

Beschreiben Sie den Ablauf des folgenden Programms zur Berechnung des größten gemeinsamen Teilers mit einem UML-Aktivitätsdiagramm. Jede elementare Anweisung soll mit mindestens einer einzelnen Aktivität angegeben werden. Vergleiche in Ausdrücken bei Kontrollanweisungen sollen ebenfalls als Aktivität modelliert werden.

```

while (a != b) {
    if (a > b) {
        a = a - b;
    } else {
        b = b - a;
    }
}

```

Lösungsvorschlag:

