

51. Verkettete Listen, Programmieraufgabe, 15 Punkte

Modifizieren Sie die Klasse `VerketteteListe`¹ aus der Vorlesung so, dass sie folgende zusätzliche Methoden unterstützt.

- (a) `public void einfüge(int i, Knoten k)`. Fügt einen neuen Knoten mit Wert i an die Stelle *nach* dem Knoten k in die Liste ein.
- (b) `public boolean lösche()`. Löscht den ersten Knoten aus der Liste. Wenn die Liste vorher leer war, soll das Ergebnis `false` sein, sonst `true`.
- (c) `public boolean lösche(Knoten k)`. Löscht den Knoten k aus der Liste. Wenn der Knoten k gar nicht zur Liste gehört, soll das Ergebnis `false` sein, sonst `true`. (Diese Operation geht nicht in konstanter Zeit.)

Geben Sie dem Programm den Namen `Aufgabe51_Nachname1_Nachname2.java`. Die obligatorische Testsuite soll als `main`-Methode der Klasse `VerketteteListe` ablaufen.

52. Arithmetik in JAVA, 15 Punkte

Arithmetische Operationen werden in JAVA strikt von links nach rechts ausgerechnet, sofern Vorrangregeln und Klammern nichts anderes vorgeben. Wenn beide Operanden einer arithmetischen Operation vom Typ `int`, `long`, `float` oder `double` sind, hat das Ergebnis denselben Typ; bei gemischten Operanden wird ein Operand vorher in den „allgemeineren“ Typ umgewandelt (`int` \rightarrow `long` \rightarrow `float` \rightarrow `double`). Gleitkommakonstanten wie `0.1` werden immer als `double`-Werte interpretiert. Werte vom Typ `int` liegen im Intervall $[-2^{31}, 2^{31} - 1]$, und ganzzahlige Arithmetik mit solchen Werten wird immer modulo 2^{32} ausgeführt.

Die folgende Funktion zeigt acht verschiedene Arten, die Summe $\sum_{i=1}^n 1/i^2$ zu berechnen, die für $n \rightarrow \infty$ (langsam) gegen $\pi^2/6$ konvergiert.

```
static double Summe(int n) {
    double s = 0.0;
    float eins = 1;
    for (int i=1; i<=n; i++)
        Varianten {
            s += 1/i*i;           (1)
            s += 1/(i*i);        (2)
            s += 1.0/i/i;        (3)
            s += 1.0/(i*i);      (4)
            s += 1.0/((double)(i)*i); (5)
            s += 1/(double)(i*i); (6)
            s += eins/i/i;       (7)
            s += 1/(1.0*i*i);    (8)
        }
    return s;
}
```

Testen Sie jede einzelne dieser Varianten jeweils für $n = 100, 10\,000, 10^6, 10^8$ und 10^9 und vergleichen Sie das Ergebnis mit dem Grenzwert. Erklären Sie, was passiert.² Formulieren Sie eine Vermutung, wie groß ungefähr der Abstand zwischen dem Grenzwert und der (exakten) Summe der ersten n Glieder ist.

¹<http://www.inf.fu-berlin.de/lehre/SS16/ALP2/VerketteteListe.java>

²https://de.wikipedia.org/wiki/IEEE_754#Unendlich