

# Übungsaufgabe 1: Schnittstelle für Langzahl-Arithmetik

## Hintergrund

Zum Rechnen mit extrem hoher Genauigkeit reichen double-Zahlen nicht aus. Ersatzweise muss eine andere Speicherform gefunden werden. Z.B. könnte man Strings verwenden. Alternativ lassen sich die Nachkommastellen als Elemente eines int-Feldes speichern. Dabei sind Einzelheiten wie das Vorzeichen oder die Zahlenlänge noch festzulegen.

Damit man mit derartigen Daten arbeiten und rechnen kann, sind diverse Hilfsprogramme notwendig, z.B. zur Eingabe, zum Addieren und Subtrahieren, zur Multiplikation und Division, zur Ausgabe, usw.

Da für den Anwender die interne Darstellung der Zahlen von untergeordneter Bedeutung ist, soll eine einfache Schnittstelle zum Arbeiten mit Langzahlen entwickelt werden. Dafür eignet sich das Klassenkonzept der OOP: Die benötigten Hilfsprogramme sind als Methoden (oder Operatoren) in einer Klasse gebündelt und als Schnittstelle definiert.

Wenn die verschiedenen Darstellungsvarianten genau diese Schnittstelle einhalten, lassen sich die Implementierungen problemlos austauschen.

## Aufgaben

- Für 'beliebig' lange gebrochene Zahlen (z.B. max 100 Nachkomma-Stellen) soll eine derartige Schnittstelle einer Klasse definiert werden. Legen Sie die public-Definition dieser Zahlenklasse so fest, dass sie sich auf verschiedene Arten implementieren lässt (siehe unten).  
*Hinweis:* Sie können entweder die üblichen Grundoperationen ( $>>$ ,  $+$ ,  $-$ ,  $*$ ,  $/$  und  $<<$ ) verwenden oder eigene Methoden wählen, z.B. `eingabe()`, `add()`, `sub()`, `ausgabe()`.
- Für eine einfache Reihenentwicklung Ihrer Wahl ist ein Anwendungsprogramm zu schreiben, das genau diese Klassenschnittstelle verwendet. Die Zahl der Iterationsschritte soll im Dialog eingegeben werden.
- Zum Testen der Anwendung ist eine erste Version der Klasse zu implementieren, bei der der Datentyp `double` für die Zahlendarstellung verwendet wird. Mit dieser Klasse ist das Anwendungsprogramm in der ersten Version zu betreiben.
- Eine zweite Darstellungsart ist in einer Klasse zu realisieren, die nach außen genau gleich aussieht, also auch den gleichen Klassennamen und die gleichen public-Methoden besitzt. Intern wird dabei eine echte Langform (als String oder int-Feld) für die Darstellung verwendet. Legen Sie dazu die Details der Darstellung und ihre Bedeutung so genau fest, dass ein fremder C++-Programmierer die Klasse eindeutig implementieren kann.
- Beschreiben Sie die Addition und die Subtraktion von zwei Langzahlen in Pseudocode.
- Analysieren Sie den Multiplikationsalgorithmus aus der Grundschule, um das Produkt aus zwei Zahlen mit vielen Nachkommastellen zu berechnen. Wie lautet der Pseudocode dafür?
- Bei Reihenentwicklungen sind die Nenner meist ganzzahlig. Wie berechnet sich der Kehrwert einer ganzen Zahl als Langzahl? (Auch in Pseudocode)
- Implementieren Sie die Eingabe und Ausgabe sowie die drei genannten arithmetischen Operationen für diese Klasse. Mit dem gleichen Hauptprogramm soll dann alternativ das viel genauere Ergebnis berechnet werden. (Hier kommt es nicht auf die Genauigkeit an.)

Die Dokumentation mit dem Konzept (entsprechend den Aufgaben a bis g) ist bis zum 30. März abzugeben; die Implementierung bis spätestens 8. April.