



Übung 11

Aufgabe 1: Klasse Bruch - Überladen von Operatoren

In Übung 8 haben Sie ein Programm geschrieben, womit Sie Bruchrechnen können. Erweitern Sie dieses Programm indem Sie das Überladen von Operatoren benutzen, so dass Sie die Operatoren $+$, $-$, $*$ und $/$ zum Bruchrechnen benutzen können.

Entwerfen Sie eine Klasse `Bruch`, die alle benötigten Konstruktoren, Destruktoren und Operatoren beinhaltet. Überlegen Sie sich, welche dieser Operatoren als `public` deklariert werden müssen und welche als `private` deklariert werden können.

Als Zwischenschritt entwerfen Sie die Klasse `Bruch` noch mit den Prozeduren aus der Übung 8. Dadurch ändern Sie im ersten Schritt nicht allzuviel an Ihrem Programm und Fehler werden leichter zu finden sein.

Testen Sie Ihre Bibliothek wieder mit dem selben Programmstück als Hauptprogramm wie in Übung 8. Hier nochmals das Programmstück in Pseudonotation:

$m = \frac{1}{2}$ $t := \frac{1}{1}$ $v = t$
für alle $i \in \mathbb{Z}$ von 0 bis 14

$t = m \cdot t$ $v = v - t$

$t = m \cdot t$ $v = v + t$

Bildschirmausgabe: reeler Wert von v

berechnet:

$$1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots = \sum_{n=0}^{31} \frac{(-1)^n}{2^n} = \frac{715827883}{1073741824} \approx \frac{2}{3}$$

Das Programm erzeugt folgende Ausgabe, anhand derer Sie Ihre Version testen können:

0.75
0.6875
0.671875
0.667969
0.666992
0.666748
0.666687
0.666672
0.666668
0.666667
...

Das Hauptprogramm mit überladenen Operatoren wird dann etwa folgende Form haben:

```
#include "bruch.hh"

/*****
/* Hauptprogramm
*****/
void main()
{
    int i;

    Bruch m(1, 2);
    Bruch t(1, 1);
    Bruch v;
    v = t;

    for (i=0; i<15; i++) {
        t = t * m;
        v = v - t;

        t = t * m;
        v = v + t;
        cout << v.eval() << endl;
    }
    cout << v.zaehler() << "/" << v.nenner() << endl;
}
```