FH Osnabrück
Fakultät Ingenieurwissenschaften und Informatik
Prof. Dr.-Ing. B. Lang

Objektorientierte Programmierung
Praktikumsübung 2

Aufgabe (10 Punkte)

Die Rechnung mit rationalen Zahlen bleibt über viele Operationen hinweg sehr genau, wenn man jede Zahl als Bruch aus Zähler und Nenner darstellt.

Erstrebenswert ist, daß Zähler Z und Nenner N keinen gemeinsamen Teiler besitzen (d.h. teilerfremd sind). Dann werden für Zähler und Nenner möglichst kleine Werte gespeichert und der Wertebereich der Darstellung wird optimal ausgenutzt.

Bezeichnet man Zähler und Nenner des Ergebnisses mit Z_R und N_R und die Operanden mit Z_1 und N_1 sowie Z_2 und N_2, dann können die arithmetischen Grundoperationen +, -, $\times$ und / wie folgt dargestellt werden:

1. Addition, Subtraktion

   Zur Addition zweier rationaler Zahlen müssen beide Zahlen auf den Hauptnenner gebracht werden. Mit dem größten gemeinsame Teiler der Nenner $ggT(N_1, N_2)$ ergibt sich die Gleichung zur Addition:

   $$\frac{Z_1}{N_1}+\frac{Z_2}{N_2}=\frac{Z^*{}_R}{N^*{}_R}=\frac{Z_1\times n_2+Z_2\times n_1}{n_1\times n_2\times ggT(N_1,N_2)} \qquad \text{mit: } n_1=\frac{N_1}{ggT(N_1,N_2)} \text{ und } n_2=\frac{N_2}{ggT(N_1,N_2)}$$

   Somit kann der Zähler $Z^*{}_R$ und der Nenner $N^*{}_R$ der Summe berechnet werden. Aus diesen Werten muß noch der größte gemeinsame Teiler herausgekürzt werden:

   $$Z_R=\frac{Z^*{}_R}{ggT(Z^*{}_R,N^*{}_R)} \qquad N_R=\frac{N^*{}_R}{ggT(Z^*{}_R,N^*{}_R)}$$

   Die Subtraktion rationaler Zahlen wird entsprechend mit der Subtraktion im Zähler realisiert.

   Beispiel: Addieren der beiden Zahlen ($Z_1=3$, $N_1=10$) und ($Z_2=1$, $N_2=6$).

   Die Aufgabe lautet somit: $\frac{3}{10}+\frac{1}{6}$.

   $ggT(N_1, N_2)=ggt(10,6)=2 \Rightarrow n_1 = 10/2 = 5,\ n_2 = 6/2 = 3$

   $$\frac{Z^*{}_R}{N^*{}_R}=\frac{Z_1\times n_2+Z_2\times n_1}{n_1\times n_2\times ggT(N_1,N_2)}=\frac{3\times 3+1\times 5}{5\times 3\times 2}=\frac{14}{30}$$

   Gemeinsame Teiler von Zähler und Nenner im Ergebnis entfernen:

   $ggT(Z^*{}_R,N^*{}_R)=ggT(14,30)=2 \Rightarrow Z_R=14/2=7,\ N_R=30/2=15$

   Somit lautet das Ergebnis: $\frac{3}{10}+\frac{1}{6}=\frac{7}{15}$.

2. Multiplikation, Division

   Bei der Multiplikation werden die Zähler und die Nenner der beiden Operanden miteinander multipliziert, vorher sollten jedoch (um Bereichsüberschreitungen wenn möglich zu vermeiden) gemeinsame Teiler von Z_1 und N_2 sowie von Z_2 und N_1 gekürzt werden:

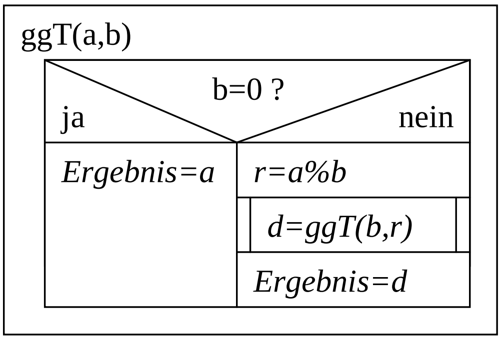
   $$Z_R=z_1\times z_2 \qquad N_R=n_1\times n_2$$

   $$\text{mit } z_1=\frac{Z_1}{ggT(Z_1,N_2)},\ z_2=\frac{Z_2}{ggT(Z_2,N_1)},\ n_1=\frac{N_1}{ggT(Z_2,N_1)},\ n_2=\frac{N_2}{ggT(Z_1,N_2)}$$

   Nimmt man an, daß die Operanden teilerfremd zur Verfügung stehen, besitzen Zähler und Nenner des Ergebnisses keinen gemeinsamen Teiler mehr. Somit braucht dann keine Kürzung mehr vorgenommen werden.

   Wie lässt sich die Division geschickt auf die Multiplikation abbilden?

Zur Erinnerung wird nochmals das Struktogramm des *ggT*-Algorithmus gezeigt:



Operatoren des rationalen Datentyps sollen die Kombinationen von Zähler und Nenner wie folgt interpretieren:

Z	N	
0	0	keine Zahl
0	>0	0
>0	0	+∞/-∞ (je nach Vorzeichen)
>0	>0	rationale Zahl

Bei der Berechnung soll folgende Behandlung von Sonderfällen erfolgen:

(rationale Zahl)/0	Ergebnis ist +∞ oder -∞ (je nach Vorzeichen)
∞/∞	Wirft eine Ausnahme vom Typ `UnendlichDurchUnendlich`
0/0	Wirft eine Ausnahme vom Typ `NullDurchNull`
0×∞, ∞×0	Wirft eine Ausnahme vom Typ `NullMalUnendlich`
(-∞)+(+∞), (+∞)+(-∞) (-∞)-(-∞), (+∞)-(+∞)	Wirft eine Ausnahme vom Typ `MinusUnendlichUndPlusUnendlich`
Operand ist keine Zahl	Wirft eine Ausnahme vom Typ `KeineZahl`

Aufgabenstellung: (Siehe vorgegebenen Programmrahmen in `Test_RationaleZahl.cxx`, `RationaleZahl.cxx` und `RationaleZahl.h`)

- Erläutern Sie, wie im Testprogramm die einzelnen Testfunktionen aufgerufen werden.
- Überladen Sie für den vorgegebenen Datentyp `RationaleZahl` die vier arithmetischen Grundoperatoren +, -, × und /. Prüfen Sie vor jeder Rechnung, ob die übergebenen Operanden wirklich teilerfremd sind. Beachten Sie das Vorzeichen der Operanden.
- Überladen Sie den <<-Operator als Ausgabeoperator für Ausgabeströme für den Datentyp `RationaleZahl`. Der Wert soll in der Form `(`*`Vorzeichen Zähler/Nenner`*`)` ausgegeben werden. Beispiel: -3/7 ergibt die Ausgabe `(-3/7)`.
  Für den Wert ∞ soll die Ausgabe `(`*`Vorzeichen`*` unendlich)`, also z.B. `(-unendlich)` erfolgen. Für Wert 0 soll `(0)` und für keine Zahl `(NaN)` ausgegeben werden.
- Erweitern Sie das vorgegebene Testprogramm so, dass es die vier Grundoperationen mit allen möglichen Ausnahmen testet. Testen Sie Ihre Operatoren des rationalen Datentyps mit dem Testprogramm.
- Verwenden Sie `make` zum Übersetzen und Binden des Programmcodes.

Zusatz: (Freiwillig)

- Detektieren Sie bei der Berechnung der Grundoperationen mögliche Überläufe in Zähler und Nenner und werfen Sie bei Überlauf die passende Ausnahme `UeberlaufImZaehler`, `UeberlaufImNenner` oder `UeberlaufImZaehlerUndNenner`.

Hinweis: Überprüfen Sie, ob sie zur Einsparung von Programmierarbeit einige der Operatoren durch Verwendung anderer von Ihnen erstellter Operatoren realisieren können.