

## Objektorientierte Programmierung

### Übungsaufgabe: Funktionen zur Klasse der Brüche, Verwendung von Referenzen

#### Lernziele:

- o Klassendefinition mit Konstruktor / Vorgabewerten
- o Initialisierungsliste
- o Mitgliedsfunktionen / Freund-Funktionen
- o Systematisches Vorgehen

#### Aufgabenstellung:

Erstellen Sie eine Klasse **cBruch**, die es ermöglicht, Brüche zu verwalten und mit ihnen zu rechnen. Ein Bruch wird mittels zweier Ganzzahlen als Mitgliedsvariablen umgesetzt. Sie kennen die Klasse **cBruch** bereits aus einer früheren Übungsaufgabe und können diese weiter verwenden.

Implementieren Sie folgende Funktionalitäten:

- Konstruktor **cBruch(Zaehler, Nenner)**, der den Bruch aus 2 Ganzzahlen initialisiert. Setzen Sie die Vorgabewerte so, dass der Bruch den Wert 0 erhält, falls keine Parameter angegeben werden.
- Erstellen Sie einen Kopierkonstruktor für die Klasse **cBruch**.
- Achten Sie darauf, dass der Nenner eines Bruches immer positiv ist. Insbesondere darf er **nie Null** sein! Ersetzen Sie in diesem Fall den Wert 0 durch den Wert 1. Erweitern Sie Brüche mit negativem Nenner mit -1.
- Eine Methode **kuerzen()** zum Kürzen eines Bruchs als Mitgliedsfunktion der Klasse **cBruch**.
- Eine private Methode **ausgabe()**, die den Bruch in der Form Zähler/Nenner sowie als Gleitkommazahl ausgibt.
- Funktionen **add()**, **subt()**, **mult()**, **div()** für die vier Grundrechenarten **+** **-** **\*** **/** für Brüche (Objekt **cBruch** mit Objekt **cBruch**). Vergessen Sie nicht das Kürzen der Brüche nach der Berechnung.
- Eine Freund-Funktion **vergleich()** für Brüche (Bruch mit Bruch). Der Typ der Funktion ist integer, der Rückgabewert:  
**-1, wenn der erste Operand kleiner ist als der zweite**  
**0, wenn beide Brüche gleich groß sind (auch z.B. 3/4 und 9/12 )**  
**1, wenn der erste Operand größer ist als der zweite**
- Erstellen Sie eine Funktion **tausch()**, die zwei Instanzen von **cBruch** miteinander vertauscht.
- Erstellen Sie eine Funktion **sortier()**, die das Array der Brüche aufsteigend sortiert. Verwenden Sie den Bubblesort-Algorithmus (s.u.).

Implementieren Sie diese 5 Funktionen global, daher müssen diese in der Klasse als Freunde vorgestellt werden.

Verwenden Sie bei Aufrufparametern und Rückgabewerten wo immer es sinnvoll ist **Referenzen**.

Erstellen Sie ein Hauptprogramm:

- Instanzieren Sie 8 Brüche als ein Array aus **cBruch**-Objekten.
- Initialisieren Sie dabei 7 der 8 Brüche mit den folgenden Werten über eine Initialisierungsliste:  
3 / 6  
21 / -7  
8 / -10  
-4 / 3  
-8 / 13  
4 / 5  
21 / 37
- Geben Sie in einer Schleife die Werte aller 8 Brüche aus.
- Führen Sie folgende Berechnungen durch:
  - o Addieren Sie die ersten beiden Brüche und geben Sie das Ergebnis aus mit der Funktion `ausgabe()`.
  - o Subtrahieren Sie die nächsten beiden Brüche und geben Sie das Ergebnis aus mit der Funktion `ausgabe()`.
  - o Multiplizieren Sie die nächsten beiden Brüche und geben Sie das Ergebnis aus mit der Funktion `ausgabe()`.
  - o Dividieren Sie die nächsten beiden Brüche und geben Sie das Ergebnis aus mit der Funktion `ausgabe()`.
- Sortieren Sie die Brüche im Array.
- Geben Sie in einer Schleife die Werte aller 8 Brüche aus.

---

Hinweis 1: Die Bruchrechnungs-Grundrechenarten sind wie folgt definiert:

#### Addition

$$\frac{a}{b} + \frac{c}{d} = \frac{a \cdot d + c \cdot b}{b \cdot d}$$

#### Multiplikation

$$\frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot c}{b \cdot d}$$

#### Subtraktion

$$\frac{a}{b} - \frac{c}{d} = \frac{a \cdot d - c \cdot b}{b \cdot d}$$

#### Division

$$\frac{a}{b} : \frac{c}{d} = \frac{a}{b} \cdot \frac{d}{c} = \frac{a \cdot d}{b \cdot c}$$

---

---

Hinweis 2: Der Algorithmus des Bubblesort kann wie folgt umgesetzt werden:

```
// Bubble Sort Algorithmus (unoptimiert) bei ANZ Objekten

for (int n = ANZ; n>1; n--) {
    for (int i=0; i<n-1; i++) {
        if (elem[i] > elem[i+1]) {           // Elemente vergleichen
            tausch(elem[i], elem[i+1]);     // Elemente tauschen
        }
    }
}
```

> : Für das Vergleichen von Objekten müssen Sie Ihre Funktion **vergleich()** entsprechend einsetzen

---