

C/C++ für Java-Programmierer

Aufgabe 2.1: *RationalNumber à la OO*

```
RationalNumber a(1,2), b(2,4);  
assert( a == b );  
  
RationalNumber c = a+b;  
assert( c.num() == 1 && c.denom() == 1);
```

Aufgabe 2 besteht aus mehreren aufeinander aufbauenden Teilaufgaben und kann/soll über mehrere Wochen bearbeitet werden (siehe *Abgabe*). Dieses Blatt enthält lediglich die Teilaufgabe 2.1; weitere Aufgaben folgen auf separaten Blättern.

Vorbereitung

Laden Sie den Testtreiber `main_a21.cpp` für diese Aufgabe von Moodle herunter. Hier finden Sie eine Reihe von Unit-Tests der zu implementierenden Klasse. Mittels `#if ... #endif` sind viele der Tests zunächst deaktiviert; aktivieren Sie diese schrittweise, wenn Sie die entsprechende Funktionalität implementiert haben.

Aufgabe 2.1: Modul **MyDate**

Implementieren Sie eine C++-Klasse `RationalNumber` bestehend aus einer Headerdatei `rationalnumber.h` und einer Implementierungsdatei `rationalnumber.cpp`. Aller Inhalt dieser Dateien soll zusätzlich in einen **Namensraum** `rn` gekapselt sein (siehe Testtreiber).

- Studieren Sie den bereitgestellten Testtreiber und portieren Sie diejenige Funktionalität aus Aufgabe 1, die Sie zum Bestehen der Tests benötigen.
 - Konstruktor mit 0, 1, oder 2 Argumenten. Implementieren Sie nur einen einzigen Konstruktor, der alle drei Fälle richtig abdeckt.
 - Accessor-Methoden `num()`, `denom()` und `isValid()`
 - Rechnende Funktionen `inverse()`, `operator+()` und (unärer) `operator-()`
 - Vergleichsoperationen `operator==()` und `operator<()`
 - Wie bereits in Aufgabe 1 benötigen Sie intern eine Methode `normalize()`, die Sie als private Methode der Klasse umsetzen sollten. Wenn Sie die Schnittstelle der Klasse minimal halten (Stichwort: *immutable!*), müssen Sie `normalize()` nur an einer einzigen Stelle aufrufen.

- Es handelt sich um einen sehr kompakten Datentyp entsprechend zwei ints. Verwenden Sie für diese Klasse ausschließlich *call by value* und keinerlei Zeiger, Referenzen, oder new-Operatoren.
- Testen Sie Ihre Klasse mit Hilfe des Testtreibers und fügen Sie je nach Bedarf weitere Tests hinzu. Sämtliche Methoden der Klasse werden in der nächsten Teilaufgabe verwendet werden.

Abgabe und Demonstration

Die Abgabe der gesamten Aufgabe 2 soll bis zu dem in Moodle festgelegten Termin erfolgen. Verspätete Abgaben werden wie in den Handouts beschrieben mit einem Abschlag von 2/3-Note je angefangener Woche Verspätung belegt. Geben Sie bitte pro Gruppe jeweils nur eine einzige .zip-Datei mit den Quellen Ihrer Lösung ab. Bitte keine Kompilate in der Abgabe!

Demonstrieren und erläutern Sie dem Übungsleiter Ihre Lösung in der Übungen nach dem Abgabedatum. Die Qualität Ihrer Demonstration ist, neben dem abgegebenen Code, ausschlaggebend für die Bewertung! Es wird erwartet, dass alle Mitglieder einer Gruppe anwesend sind und Fragen beantworten können.