



## Programmieren 3 AI – Praktikum

# Übungsblatt 10

2 Punkte

Sebastian Flothow

2011-12-19

**Abnahme:** 2012-01-09

Zur Abnahme ist die persönliche Anwesenheit erforderlich. Bearbeiten Sie die Aufgaben so, dass Sie nicht nur funktionierenden Code vorzuweisen haben, sondern diesen auch erläutern und Fragen dazu beantworten können. Zusätzlich sind die Lösungen am Tag der Abnahme per Email an [sebastian@flothow.de](mailto:sebastian@flothow.de) einzusenden; dies dient u.a. der Prüfung auf eventuelle Plagiate.

Statten Sie Ihr Projektverzeichnis mit einer Qt-Projektdatei aus, die Compiler-Warnungen aktiviert. Rufen Sie vor dem Packen `make distclean` auf (d.h. die eingesandten Archive sollen nur Quelltexte enthalten, keine Kompilate).

Senden Sie Ihre Abgabe als tar.gz-Archiv(e). Diese müssen so gepackt sein, dass jedes Archiv genau ein Verzeichnis enthält, dessen Name (bis auf das Suffix) mit dem des Archivs übereinstimmt; ein solches Archiv kann durch einen Aufruf der Form `tar -czf verzeichnis.tar.gz verzeichnis` erzeugt werden.

Beim Kompilieren der Programme dürfen keine Fehler oder Warnungen auftreten.

### Aufgabe 1

Übernehmen Sie die Klasse `Rational` von Blatt 8 Aufgabe 2.

(a) Dokumentieren Sie die Klasse mit Doxygen. Versehen Sie dazu in der Header-Datei die Klasse selbst und alle öffentlichen Komponenten mit entsprechenden Kommentaren. Achten Sie darauf alle Aspekte der Methoden zu dokumentieren, also Parameter, Rückgabewert und ggf. welche Exceptions in welchen Situationen geworfen werden. Legen Sie ein geeignetes Doxyfile an und erzeugen Sie die Dokumentation; dabei dürfen keine Warnungen oder Fehler auftreten.

(b) Schreiben Sie einen kleinen stack-basierten Rechner mit grafischer Oberfläche. Das Programm soll einen Stack von Brüchen (`std::stack<Rational>`) verwalten; das Programmfenster soll stets anzeigen wieviele Elemente der Stack enthält, und, falls er nicht leer ist, welchen Wert das oberste Element hat.

Das Programm soll die folgenden Funktionen bieten:

- Eingeben eines Bruchs. Der eingegebene Bruch wird dem Stack hinzugefügt. Vorbedingung: keine.
- Addition. Entnimmt zwei Werte vom Stack, addiert sie, und legt das Ergebnis wieder auf den Stack. Vorbedingung: Stack muss mindestens zwei Elemente enthalten.
- Multiplikation. Entnimmt zwei Werte vom Stack, multipliziert sie, und legt das Ergebnis wieder auf den Stack. Vorbedingung: Stack muss mindestens zwei Elemente enthalten.
- Additiv Inverses. Entnimmt einen Wert vom Stack, kehrt das Vorzeichen um, und legt das Ergebnis wieder auf den Stack. Vorbedingung: Stack muss mindestens ein Element enthalten.
- Multiplikativ Inverses. Entnimmt einen Wert vom Stack, bildet den Kehrwert, und legt das Ergebnis wieder auf den Stack. Vorbedingung: Stack muss mindestens ein Element enthalten.

Statten Sie das Programmfenster dazu mit geeigneten Knöpfen und Eingabefeldern aus. Vor der Ausführung jeder Operation ist zu prüfen ob die jeweilige Vorbedingung erfüllt ist, wenn dies nicht der Fall ist oder bei der Ausführung der Operation ein Fehler auftritt soll das Programm eine entsprechende Fehlermeldung in einem Dialogfenster anzeigen.

Testen Sie das Programm, indem Sie einige Berechnungen durchführen. Überprüfen Sie insbesondere, ob alle Fehlerfälle korrekt gehandhabt werden.