

Prof. Markus Gross, Bruno Heidelberger, Richard Keiser, Nicky Kern, Edouard Lamboray, Christoph Niederberger, Tim Weyrich, Felix Eberhard, Manuel Graber, Nathalie Kellenberger, Marcel Kessler, Lior Wehrli

## Uebung 11 - Musterlösung

### 1. Konstruktoren

Code: siehe Datei u11\_main.cpp, rational.h und rational.cpp

Es kann nicht ausgeschlossen werden, dass der Nenner einer `Rational`-Zahl jemals `Null` wird. Dazu müssten neben den Konstruktoren auch alle anderen Mitgliedsfunktionen der Klasse `Rational` (z.B. `set()` oder `mult()`) auf diesen Sonderfall testen.

- a) richtig -> `Rational::Rational()`
- b) richtig -> `Rational::Rational(int, int)`
- c) falsch
- d) falsch
- e) richtig -> `Rational::Rational(int, int)`
- f) falsch

### 2. Gültigkeitsbereich und Zugriffskontrolle

- a) `this->value` (wird vom Funktionsparameter `value` überdeckt!)
- b) `value`
- c) `::value` (wird vom Funktionsparameter `value` überdeckt!)
- d) Nein. Die Mitgliedsvariable `value` gehört defaultmässig zur *Private Section* der Klasse `A`. Ein direkter Zugriff von ausserhalb der Klasse, wie durch die gegebene Zuweisung, ist daher nicht erlaubt und ergibt einen Fehler bei der Kompilierung.