Übung 04

Korrektur: Sebastian Pritz

Name: Lukas Fallmann

**Punkte:**

Allgemeine Anmerkungen:

# Beispiel 1: Klasse rational\_t erweitern xx von 100

Lösungsidee: /24

|  |  |
| --- | --- |
| **Anmerkung**  Operationen von Punkt 5 nicht beschrieben  Neutrales Element in nelm nicht beschrieben | **Abzug**  -4  -2 |

Quellcode: /29

|  |  |
| --- | --- |
| **Anmerkung**  Es gibt einige Funktionen, die man im h File direkt implementieren muss (Stichwort: Bart-Nackmann Trick und Inline Implementierung), jedoch sollte trotzdem, dort wo es geht, auf h und cpp aufgeteilt werden!  Sehr sehr viel Code in der Main 🡪 Evtl. Strukturierung je Datentyp oder Zusammenfassung von Tests (Vergleiche, Rechenoperationen, etc...).  Implementierung zwar schön, aber etwas overengineered mit Singleton-Pattern, da es nicht wirklich nötig ist extra Daten zu speichern 🡪 Static Class + Method reicht. Alternativ einfach Testmethoden in Modul, statt Klasse. | **Abzug**  **Hinweis**  **-2**  **Hinweis** |

Testfälle: /38

|  |  |
| --- | --- |
| **Anmerkung**  Sehr gut und ausführlich! Auch mit mehreren Datentypen (was ja der Sinn von Generics ist) getestet! | **Abzug** |