

Übung 04

Korrektur: Sebastian Pritz

Name: Lukas Fallmann

Punkte: 92

Allgemeine Anmerkungen:

Beispiel 1: Klasse `rational_t` erweitern

92 von 100

Lösungsidee:

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

Quellcode:

| Anmerkung | Abzug |
|--|----------------|
| 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! | Hinweis |
| Sehr sehr viel Code in der Main → Evtl. Strukturierung je Datentyp oder Zusammenfassung von Tests (Vergleiche, Rechenoperationen, etc...). | -2 |
| 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. | Hinweis |

Testfälle:

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