



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

Fakultät für Informatik  
Professur Praktische Informatik  
Prof. Dr. G. Rünger  
Dr. R. Dietze

---

## Funktionale Programmierung/Höhere Programmiersprachen

### Wintersemester 2022/2023

### 2. Übungsserie

Falls Sie eine Prüfungsvorleistung erbringen müssen, geben Sie Ihre Lösungen bitte **jeweils vor Beginn der ersten Übungseinheit (Donnerstag 15:30 Uhr)** beim Übungsleiter ab oder laden sie bis zu diesem Zeitpunkt unter dem Punkt Abgabe im OPAL hoch.

#### Aufgabe 1:

Geben Sie Haskell-Funktionen für die folgenden Berechnungen an.

- a) Berechnung der  $i$ -ten Fibonacci-Zahl.  
`calcFibonacci :: Int -> Int`
- b) Berechnung der Summe der ersten  $n$  aufeinanderfolgenden natürlichen Zahlen.  
`calcSum :: Int -> Int`
- c) Berechnung des größten gemeinsamen Teilers zweier natürlicher Zahlen mit dem Euklidischen Algorithmus.  
`ggT :: Int -> Int -> Int`

#### Aufgabe 2:

Geben Sie Haskell-Funktionen für die folgenden Berechnungen an. Erstellen Sie jeweils eine Variante **mit** und eine **ohne** Nutzung der in der Vorlesung gegebenen Funktionen `threeEqual` bzw. `xor`.

- a) Funktion `fourEqual` zur Berechnung ob vier Integer-Werte gleich.
- b) Funktion `xor3` zur Berechnung des exklusiven Oder für drei boolesche Werte (sog. *odd-parity*, entspricht der Verkettung von zwei binären exklusiven Oder).

#### Aufgabe 3:

Geben Sie Haskell-Funktionen unter Verwendung von Guarded commands für die folgenden Berechnungen an.

- a) Eine Funktion `middleOfThree` zur Ermittlung des (größenmäßig) mittleren Wertes von drei Float-Werten.
- b) Die Funktionen `howManyEqualOfThree` und `howManyEqualOfFour`, die zurückgeben, wie viele von drei bzw. vier gegebenen Integer-Werten gleich sind. Beispielsweise soll die Funktion `howManyEqualOfThree` entweder 1 (wenn alle drei Integer-Werte verschieden sind), 2 (wenn zwei der drei Integer-Werte gleich sind) oder 3 (wenn alle drei Integer-Werte gleich sind) zurückgeben.

#### Aufgabe 4:

Geben Sie eine Haskell-Funktion zur Berechnung des negierten logischen Und (sog. *nand*) an. Erstellen Sie eine Variante **mit** und eine **ohne** Verwendung von Guarded commands.