

## – Praktikumsaufgabe 6 –

### Thema: *Module, Fehlerbehandlung*

**Zielstellung:** Praktische Arbeit mit obigen Elementen.

1. Modularisieren Sie das Programm zur Addition und Multiplikation komplexer Zahlen aus dem 5. Praktikum folgendermaßen:

- Code und struct sollen in einem Modul `complex_number` gekapselt werden
- ein Submodul `arith` soll die assoziierten Funktionen zur Addition und Multiplikation aufnehmen; diese müssen natürlich public sein
- ein weiteres Submodul `misc` nimmt die Funktion zur Ausgabe einer komplexen Zahl auf; auch diese ist public
- das Kommando `cargo modules generate tree` muss die folgende Ausgabe erzeugen:

```
robge@ilpro122:~/txt/job/htw/rust/src/complex2$ cargo-modules generate
```

```
crate complex2
└─ mod complex_number: pub(crate)
    └─ mod arith: pub
        └─ mod misc: pub
```

- das `struct` zur Modellierung einer komplexen Zahl soll public sein, seine Elemente jedoch private
- Sie benötigen in diesem Fall noch eine assoziierte Funktion (public, `new`, o. ä.), zur Initialisierung der Elemente des structs.

**Hinweis:** Die Ursprungsaufgabe ist “Definieren Sie ein `struct` zur Darstellung komplexer Zahlen. Entwickeln Sie danach zwei Methoden (besser: assoziierte Funktionen) zur Addition und Multiplikation zweier komplexer Zahlen sowie eine Methode zur Ausgabe einer komplexen Zahl und testen Sie diese.”

2. Erweitern Sie das Beispiel zur Dateiarbeit aus der Vorlesung folgendermaßen:
  - Wenn die zu eröffnende Datei nicht existiert, soll diese angelegt werden (`File::create()`).
  - Wenn die zu eröffnende Datei nicht die erforderlichen Rechte besitzt, sollen diese durch das Programm gesetzt werden (`File::(set_permissions())`)

Denken sie daran, dass beide Operationen fehlschlagen können!