

– Praktikumsaufgabe 4 –

Thema: *Ownership*

Zielstellung: Praktische Beispiele für Ownership-Regeln, ‘Copy-vs-Move’, Borrowing und Referenzen

1. Gegeben ist das folgende Rust-Programm:

```
fn main() {  
    let x = 3.1415;  
    let y = x;  
    println!("x={x}, y={y}");  
  
    let s = String::from("Tim, the Enchanter");  
    let t = s;  
    println!("s={s}, t={t}");  
}
```

- a) Lässt sich das Programm fehlerfrei übersetzen? Warum (nicht)?
 - b) Lässt sich das Programm übersetzen, wenn das zweite `println!`-Makro gelöscht wird?
 - c) Finden Sie weitere 4 Möglichkeiten, den „Fehler“ zu beheben.
2.
 - a) Implementieren Sie eine Funktion (ohne Resultatwert), die die Werte zweier Parameter vom Typ `i32` vertauscht. Verwenden Sie eine temporäre Variable.
 - b) Implementieren Sie nun eine ähnliche Funktion für den Datentyp `String`. Was müssen Sie gegenüber der Lösung aus a) verändern?
 - c) Schaffen Sie es auch, a) oder b) *ohne* Nutzung einer temporären Variable zu realisieren?
 - 3.* Erweitern Sie das Programm zur Berechnung des Apfelmännchens um eine Zoom-Funktion.