

# Programmieren in Rust

## Praktikumsaufgabe 8: Pattern Matching

### 1. Kehrwerte

- a) Schreiben Sie eine Funktion `fn reciprocal(n: i64) -> Option<f64>` zur Kehrwertbildung!
- b) Beim Test dieser Funktion im Hauptprogramm soll die folgende Ausgabe erscheinen. Implementieren Sie dies!

```
1/-3 = -0.3333333333333333
1/-2 = -0.5
1/-1 = -1
1/0 = n.d.
1/1 = 1
1/2 = 0.5
1/3 = 0.3333333333333333
1/4 = 0.25
1/5 = 0.2
```

- c) Sowohl in der Funktion als auch im Hauptprogramm sollen dabei `match`-Konstrukte verwendet werden. Außerdem ist eine `for`-Schleife zu empfehlen. Führen Sie ggf. entsprechende *Refactorings* durch!
- d) Ersetzen Sie das `match`-Konstrukt im Hauptprogramm nun durch ein `if-let(-else)`! Welche Variante ist kürzer? Welche ist besser lesbar?
2. **Obfuscation?** Gegeben sei das folgende Quellcode-Stück.

```
for number in 1..=18 {
    print!("{}",
        match number {
            1 => "E",
            n if i32::abs(6-n%10)==1 => "E",
            n if n%10==6 => "S",
            n if n%10==8 => "L",
            14 => "W",
            n if n%5==4 => " ",
            13 => " ",
            2 => "I",
            3 => "N",
            10 => "A",
            11 => "U",
            12 => "S",
            _ => ""
        }
    );
}
println!("");
```

Was macht es und warum funktioniert es? Welche Unterstützung kann Ihnen bei diesem Code der Compiler geben und warum?