

Aufgabe 1: Störung

Team-ID: 00428

Team: MingXu

Bearbeiter dieser Aufgabe:
Max Obreiter

1. October 2022

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	3

Lösungsidee

Es wird nach Stellen in der Nachricht gesucht, dessen Wörter passen und bei einem „_“ ein Wort haben.

Umsetzung

Mithilfe von regulären Ausdrücken wird nach den passenden Stellen gesucht. Satzzeichen werden hierbei ignoriert, jedoch bei der Ausgabe angezeigt.

Bei jeder Aufgabe werden die zu bearbeiteten Aufgaben als Glob-Pattern (mehrere möglich) in den CmdLine-Argumente angegeben.

Beispiele

Störung0:

Ergebnisse für [das _ mir _ _ vor]:

Das kommt mir gar nicht richtig vor

Störung1:

Ergebnisse für [ich muß _ clara _]:

0: Ich muß in Clara verwandelt

1: Ich muß doch Clara sein

Störung2:

Ergebnisse für [fressen _ gern _]:

0: Fressen Katzen [neue Zeile] gern Spatzen

1: Fressen Katzen gern Spatzen

2: Fressen Spatzen gern Katzen

Störung3:

Ergebnisse für [das _ fing _]:

0: das Spiel fing an

1: Das Publikum fing an

Störung4:

Ergebnisse für [ein _ _ tag]:

0: ein sehr schöner Tag

Störung5:

Ergebnisse für [wollen _ so _ sein]:

0: Wollen Sie so gut sein

Quellcode (Rust)

```
println!("Ergebnisse für [{nachricht}]");  
// für die Nachricht "a _ b" wäre der reguläre Ausdruck /a\s+\w+b/i  
// \s+, da es bei Störung2 eine Lösung mit neuer Zeile + Tab gibt  
// \w+ für eine Reihe von Buchstaben  
// /i (bei dieser Regex-Bibliothek gibt man dies mit "(?i)" an), um  
Gross-/Kleinbuchstaben zu ignorieren  
let regex = Regex::new(&format!(  
    "(?i){}",  
    nachricht.replace(' ', r"\s+").replace('_', r"\w+")  
)).  
unwrap();  
for (i, uebereinstimmung) in regex.captures_iter(TEXT).enumerate() {  
    // uebereinstimmung ist wie ein Array mit dem 0ten Element als "Match"  
    println!("{i}: {}", &uebereinstimmung[0]);  
}
```