

Saalübung 01 - Validierung von Kreditkartennummern

Timo Specht

April 16, 2024

Allgemeine Informationen

Allgemeine Informationen

- Validierung erfolgt mittels Luhn-Algorithmus
- Methode zur Berechnung einer Prüfsumme
- entwickelt in den 1950er-Jahren vom deutsch amerikanischen Informatiker Hans Peter Luhn
- eine Art gewichtete Quersumme

Schritt 1: Zahl zu Ziffern

- Schreibe Ziffern der zu prüfenden Zahl in eine Liste.
- Beispiel: 123456789 \rightarrow [1,2,3,4,5,6,7,8,9]

Schritt 1: Zahl zu Ziffern

- Schreibe Ziffern der zu prüfenden Zahl in eine Liste.
- Beispiel: 123456789 \rightarrow [1,2,3,4,5,6,7,8,9]

```
toDigits :: Int -> [Int]
toDigitsRev :: Int -> [Int]
```

```
toDigits 0 = []
toDigits (-17) = []
```

Schritt 2: Verdoppeln

- Verdopple jede zweite Ziffer beginnend von hinten.
- Beispiel: $[8,7,6,5] \rightarrow [16,7,12,5]$
- Beispiel: $[1,2,3] \rightarrow [1,4,3]$

Schritt 2: Verdoppeln

- Verdopple jede zweite Ziffer beginnend von hinten.
- Beispiel: $[8,7,6,5] \rightarrow [16,7,12,5]$
- Beispiel: $[1,2,3] \rightarrow [1,4,3]$

`doubleEveryOther :: [Int] -> [Int]`

Schritt 3: Summieren

- Betrachte von jeder Zahl aus Schritt 2 die Ziffern und summiere diese.
- Beispiel: $[16,7,12,5] \rightarrow 1 + 6 + 7 + 1 + 2 + 5 = 22$

Schritt 3: Summieren

- Betrachte von jeder Zahl aus Schritt 2 die Ziffern und summiere diese.
- Beispiel: $[16,7,12,5] \rightarrow 1 + 6 + 7 + 1 + 2 + 5 = 22$

`sumDigits :: [Int] -> Int`

Schritt 4: Validierung

- Prüfe, ob die Summe aus Schritt 3 modulo 10 gleich 0 ist.
- Gilt dies ist die Zahl valide.
- Beispiel: $15 \bmod 10 = 5 \rightarrow$ nicht valide!
- Beispiel: $120 \bmod 10 = 0 \rightarrow$ valide!

Schritt 4: Validierung

- Prüfe, ob die Summe aus Schritt 3 modulo 10 gleich 0 ist.
- Gilt dies ist die Zahl valide.
- Beispiel: $15 \bmod 10 = 5 \rightarrow$ nicht valide!
- Beispiel: $120 \bmod 10 = 0 \rightarrow$ valide!

`validate :: Int -> Bool`

- `https://www.seas.upenn.edu/~cis1940/spring13/lectures.html`