

# Anwendungsübung zur Rekursion

Jacques Mock Schindler

11.12.2024

In diesem Notebook soll eine graphische Anwendung einer rekursiven Funktion implementiert werden. Als Grafiklibrary wird PyTamaro verwendet. In der folgenden Code Zelle werden die erforderlichen Funktionen geladen.

```
# Import der erforderlichen PyTamaro Funktionen
from pytamaro import (
    rectangle, Graphic, Empty,
    black,
    pin, bottom_left, bottom_right, top_left, top_right, bottom_center,
    rotate, graphic_height, overlay,
    show_graphic, save_graphic,
)

from pytamaro.de import kombiniere
```

Im folgenden soll es darum gehen diese Schneeflocke aus Elementen, wie dem folgenden zusammenzubauen.

Dabei ist die Grundkomponente offensichtlich eine kurze Gerade.

```
def draw_line(length: float) -> Graphic:
    # TODO: implementieren Sie das zeichnen einer geraden
    pass

show_graphic(draw_line(50))
```

Aus dieser kurzen Geraden, kann ein einzelner Zacken zusammengebaut werden.

```
def compose_element(element: Graphic) -> Graphic:
    #TODO: zeichnen Sie aus der Grafik der vorangegangenen
    # Funktion einen einzelnen Zacken
    pass
```

```
basiselement = compose_element(testline)
show_graphic(basiselement)
```

Aus diesem einzelnen Zacken ist als nächstes ein fein gezahnte Element zu bauen. Der Funktion soll als Argument die Rekursionstiefe übergeben werden. Die vorher programmierten Funktionen können als Hilfsfunktionen aufgerufen werden.

```
def draw_zahnung(n: int, m = 1) -> Graphic:
    # TODO: implementieren Sie eine rekursive Funktion
    # zum zeichnen eines gezahnten Elementes
    pass
```

Die so gezeichnete gezahnte Linie bildet die Basis einer Schneeflocke. Implementieren Sie eine Funktion zum zeichnen einer Schneeflocke. Der Funktion soll die Rekursionstiefe als Argument übergeben werden können.

```
def create_snow_flake(n: int) -> Graphic:
    # TODO: kombinieren Sie die bisherigen Element zu einer Schneeflocke
    pass
```

Hier finden sie die Musterlösung der Übung Schneeflocke.