Freie Universität Berlin

Prof. Dr. Claudia Müller-Birn, Barry Linnert

Objektorientierte Programmierung, SoSe 17

Übung 05

TutorIn: Thierry Meurers
Tutorium 10

Stefaan Hessmann, Jaap Pedersen, Mark Niehues

30. Mai 2017

---

1 Aufgabe 1 siehe handschriftliches Blatt

2 Aufgabe 2 siehe handschriftliches Blatt

3 Aufgabe 3 siehe handschriftliches Blatt

4 Türme von Hanoi in Python iterative lösung

Listing 1: Code zu Aufgabe 4

```
r"""
Übung 5
Aufgabe 4

Iterative Solution for Problem "Towers of Hanoi"

Similar to solution:
http://www.geeksforgeeks.org/iterative-tower-of-hanoi/

"""

def hanoi_iterative(n, source, helper, target):

    # calculate total number of moves
    tot_mov = 2**n - 1

    # if numbers of discs is even interchange target and helper
    if n%2 == 0:
        [aux,trg] = [target,helper]
    else:
        [aux, trg] = [helper,target]

    for i in range(1,tot_mov+1):
        if i%3 == 1:
            legal_move(source,trg)
```

```
        if i%3 == 2:
            legal_move(source,aux)

        if i%3 == 0:
            legal_move(aux,trg)

        print("{}: {} \t{}: {} \t{}: {}\n" .format(source[1], source[0], aux[1], aux[0],
    trg[1], trg[0]))


def legal_move(pole1, pole2):
    if not pole2[0]:
        pole2[0].append(pole1[0].pop())
        print("move disk {} from {} to {}" .format(pole2[0][-1], pole1[1], pole2[1]))

    elif not pole1[0]:
        pole1[0].append(pole2[0].pop())
        print("move disk {} from {} to {}".format(pole1[0][-1], pole2[1], pole1[1]))

    elif pole2[0][-1]>pole1[0][-1]:
        pole2[0].append(pole1[0].pop())
        print("move disk {} from {} to {}".format(pole2[0][-1], pole1[1], pole2[1]))

    else:
        pole1[0].append(pole2[0].pop())
        print("move disk {} from {} to {}".format(pole1[0][-1], pole2[1], pole1[1]))


if __name__ == '__main__':
    n = int(input("How many discs?:\n"))

    source = list(range(1,n+1)) # Creates list from 1 to n
    source = (source[::-1], "Anfang")       # Invertes the order
    target = ([], "Ziel")
    helper = ([], "Hilfsstab")
    hanoi_iterative(n,source,helper,target)
```