Freie Universität Berlin

Prof. Dr. Claudia Müller-Birn, Barry Linnert

Objektorientierte Programmierung, SoSe 17

Übung 01

TutorIn: Thierry Meurers
Tutorium 10

Stefaan Hessmann, Jaap Pedersen, Mark Niehues

26. April 2017

---

1 Aufgabe: Wochentage (10 Punkte)

Berechnet den Wochentag anhand des Datums

Listing 1: Beispiel: Code zu Aufgabe 1

```
def weekdays(day,month,year):
    r"""

    Parameters
    ----------
    day: integer value,
        day of the month, has to be be between 1 and 31
    month: integer value,
        month as in the gregorian calender, has to  be between 1 and 12
    year: integer value,
        year

    Returns
    -------
    weekday: string

    """

    # check input data
    if not (day > 0 and day <= 31):
        raise ValueError('please, choose a day between 1 and 31.')
    if not (month >0 and month <12):
        raise ValueError('please, choose a month between 1 and 12.')
    if (month == 2 and day > 29):
        raise ValueError('The Month February as maximal 29 days')
    if (month in [4,6,9,11] and day> 30):
        raise ValueError('April, June, September and November have just 30 days')

    # initialize weekday list
    weekday = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'
    ]

    #Calculate Weekdays by Georg Glaeser
    #https://de.wikipedia.org/wiki/Wochentagsberechnung
    transformed_month = ((month - 3) % 12) +1
```

```
    century = int(year/100)
    decade = year - century*100

    #adapted decade and century

    if (month == 1) | (month == 2):
        decade = (decade - 1) % 100
        if decade == 99:
            century -= 1

    w = int((day + (2.6 * transformed_month - 0.2) + decade + (decade/4) + (century/4) - 2
    * century) % 7)

    return weekday[w]
```

2 Aufgabe: Summen berechnen (12 Punkte)

Listing 2: Beispiel: Code zu Aufgabe 2

```
# Definition of functions
# The end of the sum is given is passed as an argument

def sum_1(max):
    sum = 0

    for i in range(1, max+1):
        sum += i

    return sum

def sum_2(max):
    sum = 0

    for i in range(1, max+1):
        sum += 1/i

    return sum

def sum_3(max):
    sum = 0

    for i in range(1, max+1):
        sum += 1/i**2

    return sum

def sum_4(max):
    sum = 0

    for i in range(1, max+1):
        sum += 1 / fac(i)

    return sum

def fac(i):
    if i < 2:
        return 1

    faculty = 1

    for k in range(2, i+1):
        faculty *= k

    return faculty
```

```
# User interaction
sum_no = int(input("Which sum would you like to calculate? "
               "Choose a Number between 1 and 8:\n"))

# Process input
if sum_no == 1:
    print(sum_1(100))
elif sum_no == 2:
    print(sum_1(100000))
elif sum_no == 3:
    print(sum_2(100))
elif sum_no == 4:
    print(sum_2(100000))
elif sum_no == 5:
    print(sum_3(100))
elif sum_no == 6:
    print(sum_3(100000))
elif sum_no == 7:
    print(sum_4(20))
elif sum_no == 8:
    print(sum_4(1000))
else:
    print("Wrong input.")
```

3 Aufgabe: Multiplikation (8 Punkte)

Listing 3: Beispiel: Code zu Aufgabe 3

```
def calculate_product():
    print('perform a multiplication with a value greater than 0 or terminate the program
    with 0')
    x = int(input('Enter a number :'))
    #check initial value
    if x == 0:
        raise ValueError('number has to be different from zero')
    #set y=1 to enter while loop
    y = 1
    while y!= 0:
        y = int(input('Enter a number greater or equal zero:'))
        if y<0:
            print('number has to be greater or equal zero')
            y = int(input('Enter a number greater or equal zero'))
        elif y != 0:
            x*= y
            print(x)
        else:
            break

    print('result of the product of all inserted number is: ',x)
    return x
```