**xercise: 4**

**Name: Oskari Helenius**

**How many tasks did you do: 8 Didn't get the 4**

**Were the tasks easy, ok, difficult: Okay!**

**Do you need help/comments in any task (if yes, to which ones):**

Part 1 and part 3 code

```python
import random

class House:

    def __init__(self, windows, floors, bed, surfaces, fridge, toiletpaper):
        self.windows = windows
        self.floors = floors
        self.bed = bed
        self.surfaces = surfaces
        self.fridge = fridge
        self.toiletpaper = toiletpaper

        #setters

    def setWindows(self, windows):
        self.windows = windows

    def setFloors(self, floors):
        self.floors = floors

    def setBed(self, bed):
        self.bed = bed

    def setSurfaces(self, surfaces):
        self.surfaces = surfaces

    def setFridge(self, fridge):
        self.fridge = fridge

    def setToiletpaper(self, toiletpaper):
        self.toiletpaper = toiletpaper


    #getters

    def getWindows(self):
            return self.windows

    def getFloors(self):
            return self.floors

    def getBed(self):
            return self.bed

    def getSurfaces(self):
            return self.surfaces

    def getFridge(self):
            return self.fridge

    def getToiletpaper(self):
            return self.toiletpaper
```

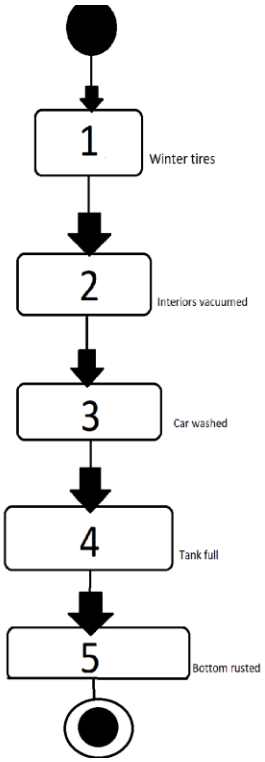
```
55
56      def __str__(self):
57          return "\nWindows are " + format(self.windows)\
58                 + "\nFloors are " +format(self.floors)\
59                 + "\nBed is " + format(self.bed)\
60                 + "\nSurfaces are " + format(self.surfaces)\
61                 + "\nFridge is " + format(self.fridge)\
62                 + "\nToiletpaper is " + format(self.toiletpaper)\
63
64
65  class Cookie:
66
67      def __init__(self, init_shape):
68          self.shape = init_shape
69
70      #setters
71
72
73      def setShape(self, desired_shape):
74          self.shape = desired_shape
75
76      def setFlavor(self):
77          flavorlist = ["Chocolate", "Vanilla", "Doublechocolate", "Nut", "Strawberry", "Lemon", "Rasb
78          self.flavor = random.choice(flavorlist)
79
80      #getters
81
82      def getSHape(self):
83          return self.shape
84
85      def getFlavor(self):
86          return self.flavor
87
88
89
90      #str-method
91
92      def __str__(self):
93          return "\nShape of cookie: " + format(self.shape)\
94                 + "\nFlavor of cookie: " +format(self.flavor)\
95
96
97
98
```

```
97
98
99  def main():
100
101     #State 1
102
103     house = House("dirty", "dirty", "unmade", "dusty", "empty", "running out", )
104     print(house)
105
106     #State 2
107
108     house.setWindows("washed")
109     house.setBed("made")
110     print(house)
111
112     #State 3
113
114     house.setFloors("vacuumed")
115     house.setSurfaces("dusted")
116     print(house)
117
118     #State 4
119
120     house.setFridge("full")
121     house.setToiletpaper("enough")
122     print(house)
123
124     #State 5
125
126     house.setToiletpaper("more than enough")
127     print(house)
128
129     favoriteflavor = "Chocolate"
130
131     Favoriteflavor = input("Gimme flavor! ")
132
133     mainlist = []
134     for i in range(10):
135         cookie = Cookie("round")
136         mainlist.append(cookie)
137
138     for key in mainlist:
139         key.setFlavor()
140         if key.getFlavor() == Favoriteflavor:
141             print("this was your favorite flavor! ")
142         else:
143             print("was not")
144
145
146
147 main()
148
149
150
151
152
```

Part 1 part 3 output

```
Windows are dirty
Floors are dirty
Bed is unmade
Surfaces are dusty
Fridge is empty
Toiletpaper is running out

Windows are washed
Floors are dirty
Bed is made
Surfaces are dusty
Fridge is empty
Toiletpaper is running out

Windows are washed
Floors are vacuumed
Bed is made
Surfaces are dusted
Fridge is empty
Toiletpaper is running out

Windows are washed
Floors are vacuumed
Bed is made
Surfaces are dusted
Fridge is full
Toiletpaper is enough

Windows are washed
Floors are vacuumed
Bed is made
Surfaces are dusted
Fridge is full
Toiletpaper is more than enough

Gimme flavor! Lemon
was not
was not
was not
was not
was not
was not
this was your favorite flavor!
was not
was not
was not
```

Part 2




Part 4

Didn’t know because did not get it working first time.

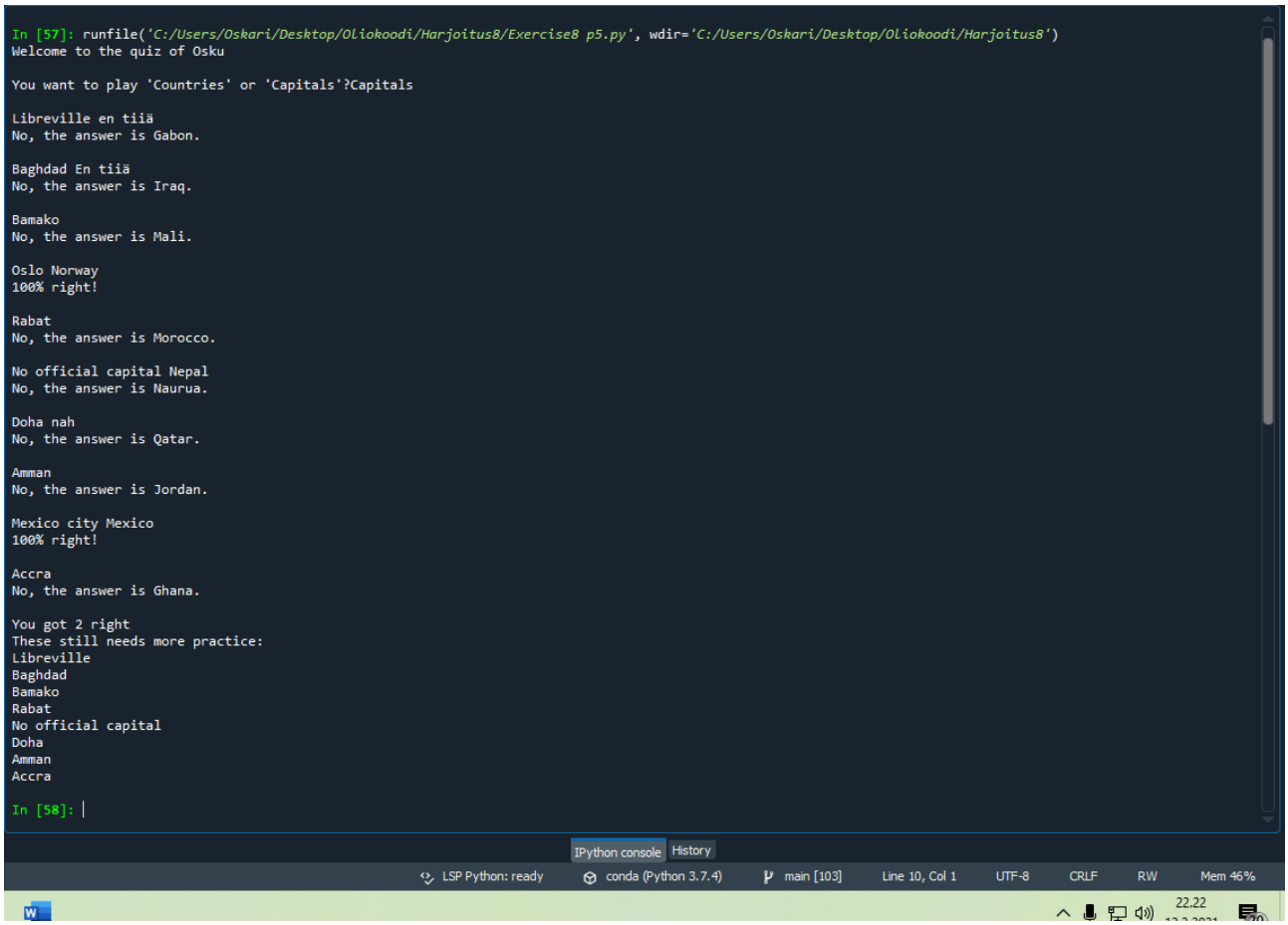
Part 5 code

C:\Users\Oskari\Desktop\Oliokoodi\Harjoitus8\Exercise8 p5.py

Exercise8 p1.py | Exercise 7 p6.py | Exercise 5 p6.py | Exercise 3 p2.py | Exercise8 p5.py

```python
from random import shuffle
print('Welcome to the quiz of Osku')

with open("countrytxt.txt") as f:
    lines = f.readlines()

shuffle(lines)
numRight = 0
wrong = []

Game = input("You want to play 'Countries' or 'Capitals'?")

if Game == "Countries":


    for line in lines[:10]:
        question, rightAnswer = line.strip().split("\t")
        answer = input(question + ' ')
        if answer == rightAnswer:
            print('100% right!')
            numRight += 1
        else:
            print('No, the answer is %s.' % rightAnswer)
            wrong.append(question)

    print('\nYou got %d right' % (numRight))
    if (wrong):
        print('These still needs more practice: ')
        for q in wrong:
            print(q)

else:

    for line in lines[:10]:
        rightAnswer, question = line.strip().split("\t")
        answer = input(question + ' ')
        if answer == rightAnswer:
            print('100% right!')
            numRight += 1
        else:
            print('No, the answer is %s.' % rightAnswer)
            wrong.append(question)

    print('\nYou got %d right' % (numRight))
    if (wrong):
        print('These still needs more practice: ')
        for q in wrong:
            print(q)
```

Part 5 output




## Self-assessment:

This exercise was easy/difficult/ok/etc. for me because… The task were good, because didn't get a headache.

Doing this exercise, I learned… …

I am still wondering…

I understood/did not understand that… ; I did/did not know that… ;