

Københavns Universitet
PoP Assignment 4 - Addressed Feedback

Victor Vangkilde Jørgensen - kft410

January 27, 2025

Contents

1	Feedback on Question 2	3
2	Feedback on question 3	4
3	Feedback on question 4	4

1 Feedback on Question 2

2B - Forklar hvorfor du har valgt specifikt disse 3 måder.

Jeg har uddybet med:

"Jeg har valgt disse 3 måder at repræsentere farver på, at det var dem jeg højst sandsynligt selv ville havde benyttet, hvis det var mig der havde lavet colour.

Jeg kan især godt lide RGB-listen, da det giver mulighed for at arbejde med mere præcise farver, og kanExtendColour ville kunne bruges til at sammenligne farver, der er meget tæt på hinanden."

2C - Du skal ikke definere en NeighbourRelation klasse, men en Country klasse. Dvs. klassen skal repræsentere et land, og land skal opbevare en liste over landets naboer.

Jeg har rettet min repræsentation til følgende:

```
1 class Country:
2     def __init__(self, name: str):
3         self.name = name
4         self.neighbours = []
5
6     def addNeighbour(self, neighbour: "Country"):
7         if neighbour not in self.neighbours:
8             self.neighbours.append(neighbour)
9             neighbour.addNeighbour(self)
10
11     def __repr__(self):
12         return f"{self.name}: {[neighbour.name for neighbour in self.
13             neighbours]}"
14
15 de = Country("de")
16 da = Country("da")
17 se = Country("se")
18 no = Country("no")
19
20 # Not sure if 'no' and 'da' should be neighbours
21 de.addNeighbour(da)
22 da.addNeighbour(se)
23 se.addNeighbour(no)
24
25 print(de)
26 print(da)
27 print(se)
28 print(no)
29 # Output:
30 de: ['da']
31 da: ['de', 'se']
32 se: ['da', 'no']
33 no: ['se']
```

2 Feedback on question 3

3A - Du mangler at forklare hvordan du har fulgt Kens metode.

Jeg har uddybet, hvordan jeg har fulgt Kens method i rapporten.

3B - Forklar mere specifikt hvordan du bruger paradigmet/paradigmerne.

Min besvarelse er nu som følgende:

"Funktionel programmering fylder en del, da det er det, jeg bruger til at læse bogstavrækkerne og sortere dem i forskellige lister.

Imperativ programmering bruges til at tælle mængden af "DIKU" i hvert af disse lister, men fylder ud over dette ikke meget.

Nu når jeg ser tilbage på opgaven, bør jeg nok havde brugt imperativ programmering i højere grad, da jeg fra forelæsningen havde hørt, at er bedst til små opgaver. Jeg havde dog troet, at funktionel programmering ville være bedst, da det skulle være bedst, når man har meget data, hvilket ikke var tilfældet i denne opgave i samme grad, som jeg havde troet."

3C - Du skal teste programmet lidt mere eksplicit og inkludere testene.

3C - Tænk mere over hvordan de forskellige inputs kunne se ud.

3C - Representative inputs are defined.

3 Feedback on question 4

4A - Programmet skal beregne det totale antal point for alle kortene tilsammen.

4A - Forklar gerne hvordan du har fulgt Kens metode her.

4B - Card klassen i din kode dominerer ret meget, da næsten alle funktioner er lavet som metoder i klassen. Genovervej hvilket paradigme dominerer her.

Jeg var klar over, at OOP fyldte mest, men havde glemt, at skrive det, da jeg opfattede members'ne som en del af funktionel programmering.

4C - Du skal teste programmet mere eksplicit.

4C - Forklar hvordan du har opdelt inputtet.

4C - Input partitions are correctly defined.

4C - Representative inputs are defined.