Exam in Complexity & Innovation Management

June 22, 2022

First Name:	Daniela
Last Name:	Ramirez
Matriculation Number:	00819401
Programme:	Bachelor Health Informatics
Semester:	SS 2022
Module:	Complexity & Innovation Mgm.
Time of exam release:	10:00am
Date:	22 June 2022
Deadline for submission:	22 June, 2022, 11:45am
Examiner:	Prof. Dr. Dominik Böhler

Question 1 - Data Visualization (45 Points)

a) Present a data visualization with 3 (or more) data dimensions derived from a dataset of your choice. (15 Points)

b) Describe the main message of your visualization and why you chose exactly this. (10 Points)

I wanted to compare the amount of deaths that are caused by abusing the usage alcohol and other diff

c) List the source code of you visulization or share accessible links on one of the following platforms: GitHub, GitLab, iLearn. (20 Points)

https://github.com/daniramirezs/Dominik-sExam22

Question 2 - Refactoring (45 Points)

Please analyze and clean the following codebase. Focus on the function named find_by() mainly. Follow the principles discussed in class: Separation of Concerns, Don't Repeat Yourself, Naming Clarity. You can find the data and the test file associated to this exam.

ATTENTION: Please only list changes to the source code you can explain. Incomplete or no explanations will result in a reduction of points.

HINT: Your solution does not need to (but can) include changes to the class itself. Should you attempt doing this, please do not forget to change the test case accordingly.

```
1  import csv
2
3
4  class FundingRaised:
5      @staticmethod
6      def where(options={}):
7          with open("../startup_funding.csv", "rt") as csvfile:
8              data = csv.reader(csvfile, delimiter=",", quotechar='"')
9              csv_data = []
10             for row in data:
11                 csv_data.append(row)
12
13         headers = csv_data[0]
14         csv_data = csv_data[1:]
15
16         # permalink,company_name,number_employees,category,city,state,
               funded_date,raised_amount,raised_currency,round
17         column_order = {
18             "company_name": 1,
19             "city": 4,
20             "state": 5,
21             "round": 9,
22         }
23
24         for column_name, column_number in column_order.items():
25             if column_name in options:
26                 result = []
27                 for row in csv_data:
28                     if row[column_number] == options[column_name]:
29                         result.append(row)
30                 csv_data = result
31
32         output = []
33         for row in csv_data:
34             mapped = dict(zip(headers, row))
35             output.append(mapped)
36
37         return output
```

```
38
39      @staticmethod
40      def find_by(options):
41          with open("../startup_funding.csv", "rt") as csvfile:
42              data = csv.reader(csvfile, delimiter=",", quotechar='"')
43              # skip header
44              next(data)
45              csv_data = []
46              for row in data:
47                  csv_data.append(row)
48
49          if "company_name" in options:
50              for row in csv_data:
51                  if row[1] == options["company_name"]:
52                      mapped = {}
53                      mapped["permalink"] = row[0]
54                      mapped["company_name"] = row[1]
55                      mapped["number_employees"] = row[2]
56                      mapped["category"] = row[3]
57                      mapped["city"] = row[4]
58                      mapped["state"] = row[5]
59                      mapped["funded_date"] = row[6]
60                      mapped["raised_amount"] = row[7]
61                      mapped["raised_currency"] = row[8]
62                      mapped["round"] = row[9]
63                      return mapped
64
65          if "city" in options:
66              for row in csv_data:
67                  if row[4] == options["city"]:
68                      mapped = {}
69                      mapped["permalink"] = row[0]
70                      mapped["company_name"] = row[1]
71                      mapped["number_employees"] = row[2]
72                      mapped["category"] = row[3]
73                      mapped["city"] = row[4]
74                      mapped["state"] = row[5]
75                      mapped["funded_date"] = row[6]
76                      mapped["raised_amount"] = row[7]
77                      mapped["raised_currency"] = row[8]
78                      mapped["round"] = row[9]
79                      return mapped
80
81          if "state" in options:
82              for row in csv_data:
83                  if row[5] == options["state"]:
84                      mapped = {}
85                      mapped["permalink"] = row[0]
86                      mapped["company_name"] = row[1]
87                      mapped["number_employees"] = row[2]
88                      mapped["category"] = row[3]
```

```
                    mapped["city"] = row[4]
                    mapped["state"] = row[5]
                    mapped["funded_date"] = row[6]
                    mapped["raised_amount"] = row[7]
                    mapped["raised_currency"] = row[8]
                    mapped["round"] = row[9]
                    return mapped

        if "round" in options:
            for row in csv_data:
                if row[9] == options["round"]:
                    mapped = {}
                    mapped["permalink"] = row[0]
                    mapped["company_name"] = row[1]
                    mapped["number_employees"] = row[2]
                    mapped["category"] = row[3]
                    mapped["city"] = row[4]
                    mapped["state"] = row[5]
                    mapped["funded_date"] = row[6]
                    mapped["raised_amount"] = row[7]
                    mapped["raised_currency"] = row[8]
                    mapped["round"] = row[9]
                    return mapped

        raise RecordNotFound


class RecordNotFound(Exception):
    pass
```

a) Describe the necessary changes and reason why you need them. List additional changes you can/could not implement separately. (20 Points)

1. Called all the headers and named only those that will be used. Like this: "csv_data = FundingRaised

b) List the optimised source code of your solution below. (25 points)

https://github.com/daniramirezs/Dominik-sExam22

Your answer here