Exam in Complexity & Innovation Management

June 22, 2022

First Name: Last Name:

Matriculation Number:

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)

3

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

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:
       @staticmethod
5
6
       def where(options={}):
           with open("../startup_funding.csv", "rt") as csvfile:
7
               data = csv.reader(csvfile, delimiter=",", quotechar='"')
8
9
               csv_data = []
               for row in data:
11
                    csv_data.append(row)
12
13
           headers = csv_data[0]
14
           csv_data = csv_data[1:]
15
           # permalink,company_name,number_employees,category,city,state,
16
               funded_date,raised_amount,raised_currency,round
17
           column_order = {
               "company_name": 1,
18
19
               "city": 4,
20
               "state": 5,
               "round": 9,
21
           }
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)
                    csv_data = result
31
32
           output = []
33
           for row in csv_data:
34
               mapped = dict(zip(headers, row))
35
                output.append(mapped)
37
           return output
```

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

```
mapped["city"] = row[4]
89
                         mapped["state"] = row[5]
90
                         mapped["funded_date"] = row[6]
91
                         mapped["raised_amount"] = row[7]
93
                         mapped["raised_currency"] = row[8]
94
                         mapped["round"] = row[9]
                         return mapped
96
            if "round" in options:
97
98
                 for row in csv_data:
                     if row[9] == options["round"]:
99
100
                         mapped = {}
101
                         mapped["permalink"] = row[0]
                         mapped["company_name"] = row[1]
102
                         mapped["number_employees"] = row[2]
103
104
                         mapped["category"] = row[3]
                         mapped["city"] = row[4]
105
106
                         mapped["state"] = row[5]
                         mapped["funded_date"] = row[6]
107
                         mapped["raised_amount"] = row[7]
108
109
                         mapped["raised_currency"] = row[8]
                         mapped["round"] = row[9]
110
111
                         return mapped
112
113
            raise RecordNotFound
114
115
116 class RecordNotFound(Exception):
117
        pass
```

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

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