Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ	«Информатики и систем управления»
КАФЕДРА	«Системы обработки информации и управления»

Отчет

по Рубежному контролю №1

По дисциплине: Разработка интернет приложений

Студент: Капитонова Д.В.
ИУ5-54Б

Москва
2020

Request: ‘Г’ , Variant: 10

1. «Computer» and «Browser» : one-to-many.
Print a list of all browsers whose name begins with the letter "A" and a list of computers on which they are installed.

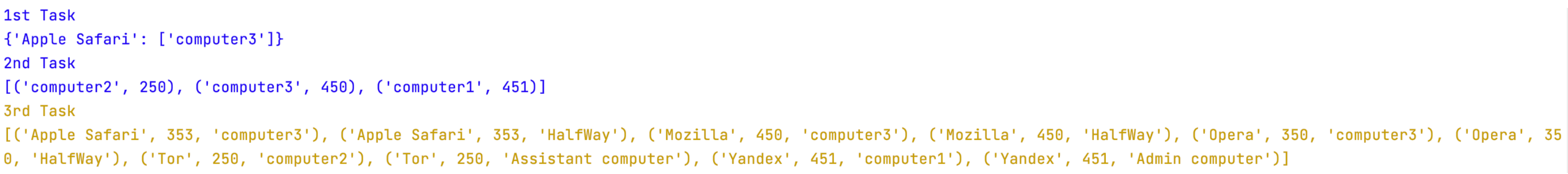
2. «Computer» and «Browser» : one-to-many.
Print a list of the computers with the maximum browser size on each computer, sorted by max browser size .

3. «Computer» and «Browser» : many-to-many.
Print all connected browsers and computers, sorted by browser, sorting by computer is arbitrary.

Listing

```
class Browser:

    def __init__(self, id, name, size, computer_id):
        self.id = id
        self.name = name
        self.size = size
        self.comp_id = computer_id
class Computer:

    def __init__(self,id,name):
        self.id = id
        self.name = name
class BrowserComputer:

    def __init__(self, computer_id, browser_id):
        self.computer_id = computer_id
        self.browser_id = browser_id



Computers = [
    Computer(1, 'computer1'),
    Computer(2, 'computer2'),
    Computer(3, 'computer3'),
    #
    Computer(11, 'Admin computer'),
    Computer(22, 'Assistant computer'),
    Computer(33, 'HalfWay')
]

Browsers = [
    Browser(1, 'Yandex', 451, 1),
    Browser(2, 'Tor', 250, 2),
    Browser(3, 'Apple Safari', 353, 3),
    Browser(4, 'Opera', 350, 3),
    Browser(5, 'Mozilla', 450, 3)
]

Browser_computer = [
    BrowserComputer(1,1),
    BrowserComputer(2,2),
    BrowserComputer(3,3),
    BrowserComputer(3,4),
    BrowserComputer(3,5),
    BrowserComputer(11,1),
    BrowserComputer(22,2),
    BrowserComputer(33,3),
    BrowserComputer(33,4),
    BrowserComputer(33,5)
]

def main():


    # one-to-many

    one_to_many = [(b.name, b.size, c.name)
                   for c in Computers
                   for b in Browsers
                   if b.comp_id == c.id ]

    many_to_many_temp = [ (c.name, bc.computer_id, bc.browser_id)
                       for c in Computers
                       for bc in Browser_computer
                       if c.id == bc.computer_id ]

    many_to_many = [(b.name, b.size, computer_name)
                     for computer_name, computer_id, browser_id in
many_to_many_temp
                     for b in Browsers if b.id == browser_id ]

    print('\033[34m1st Task')
    res1={}

    for d in Browsers:
        if 'A' == d.name[0]:
            brw = list((filter(lambda i: i[0] == d.name, one_to_many)))
            d_computers_names= [a[2] for a in brw]
            res1[d.name] = d_computers_names
    print(res1)


    print('2nd Task')
    res2_unsort = []

    for c in Computers:
        brw = list((filter(lambda i:i[2] == c.name, one_to_many)))
        if len(brw)>0:
            res2_unsort.append((c.name, max([a[1] for a in brw])))
    res2 = sorted(res2_unsort, key=itemgetter(1))
    print(res2)

    print('\033[33m3rd Task')
    res3 = sorted(many_to_many, key=itemgetter(0))
    print(res3)

if __name__ == "__main__":

    main()
```

Examples of program execution



```
1st Task
{'Apple Safari': ['computer3']}
2nd Task
[('computer2', 250), ('computer3', 450), ('computer1', 451)]
3rd Task
[('Apple Safari', 353, 'computer3'), ('Apple Safari', 353, 'HalfWay'), ('Mozilla', 450, 'computer3'), ('Mozilla', 450, 'HalfWay'), ('Opera', 350, 'computer3'), ('Opera', 35
0, 'HalfWay'), ('Tor', 250, 'computer2'), ('Tor', 250, 'Assistant computer'), ('Yandex', 451, 'computer1'), ('Yandex', 451, 'Admin computer')]
```