

Curso: Desafios para Maratona de Programação

Atividade Avaliativa A5

Participante: Divani Barbosa Gavinier

Data: 08 de maio de 2022

Código resposta, programa fonte:

```
1 class Registro():
2     def __init__(self, cidadeest, valortotal):
3         self.cidade = cidadeest # chave
4         self.valor = valortotal
5
6 if __name__ == "__main__":
7     # Motor e leitura da entrada
8     r = []
9     while True:
10        linha = input()
11        if linha in '0': break
12        temp = linha.split(';')
13        cidadeest = str(temp[3]) + " " + str(temp[4]) # concatenando cidades com
        espaço em branco e estado
14        valortotal = float(temp[2]) * float(temp[5]) # calculando g*v
15        r.append(Registro(cidadeest, valortotal))
16
17    # Resolvendo o problema
18    dic = {} # Criando um dicionário de índices = cidades+estado
19    for i in range (len(r)):
20        if dic.get(r[i].cidade) == None: # se o índice cidade+estado ainda não
        foi atribuído ao dicionário
21            dic[r[i].cidade] = r[i].valor # atribua o índice e valor
22        else: # se o índice cidade+estado já foi atribuído
23            dic[r[i].cidade] = dic[r[i].cidade] + r[i].valor # some seu valor
        total
24
25    # Imprimindo de forma ordenada o dicionário
26    cont = 0 # contador de itens impressos
27    for i in sorted(dic, key=dic.get, reverse=True): # 0 reverse=True é para
        imprimir do maior para o menor
28        if cont == 3: # Imprimindo somente os 3 primeiros itens
29            break
30        print(i, int(dic[i])) # Imprimindo o valor como inteiro
31        cont = cont + 1
```

Explicação do código:

Para resolver esse problema foi usada uma classe de nome Registro que contém os campos:

cidade: Onde será registrado o nome da cidade e estado concatenados com um espaço entre eles;

valor: Onde será registrado o valor total que corresponde ao valor de rateio multiplicado pelo valor pago a cada jogador.

Além disso foi usado um **dicionário** que relaciona uma chave a um valor específico, a chave nesse caso é o campo cidade da classe Registro e o valor específico é o campo valor dessa mesma classe. Lembrando que diferentemente das listas, onde o índice é um número, dicionários utilizam suas chaves como índice.

Levando em consideração esses importantes pontos no desenvolvimento desse código, abaixo segue sua explicação linha a linha.

Linhas 1 a 4 é onde está declarada a classe Registro com seus campos, como especificado.

Linha 6 inicia o programa principal, e nas linhas 8 a 15 tem-se o motor de leitura dos dados de entrada do programa, onde se lê uma string, declarada como linha, e caso seja atribuído zero a ela, o programa fecha automaticamente. A string linha é “splitada”, ou seja, separada em diferentes pedaços usando como separador o caractere ponto e vírgula. Os seguintes pedaços são obtidos:

temp[0] = número do concurso (N)

temp[1] = a data do sorteio (D)

temp[2] = o número de ganhadores da Sena naquela cidade (G)

temp[3] = a cidade

temp[4] = o estado

temp[5] = o valor do rateio (V) pago a cada ganhador.

Na linha 13 ocorre a concatenação da cidade com espaço em branco e o nome do estado, o comando str() garante que tanto temp[3] como temp[4] sejam definidos como strings para concatenação. A concatenação é atribuída a **string cidadeest**.

Na linha 14 calcula-se o valor de rateio multiplicado pelo valor pago a cada jogador e atribui esse valor calculado a uma variável float denominada **valortotal**. O comando float() garante que tanto temp[2] como temp[5] sejam definidos como floats para multiplicação.

Na linha 15 o Registro da linha lida é atribuído a **lista r**, que foi devidamente declarada na linha 8. O comando append é usado para adicionar itens a lista.

Estando com os dados lidos e inseridos na **lista r de Registros** devidamente ajustados, vamos para a resolução do problema computacional. Na linha 18 tem-se o comando de declaração do **dicionário dic**. Para atribuição dos itens da **lista r** ao **dicionário dic** foi usado um laço de repetição, vide linha 19, que percorre todos os itens da lista. O comando len(r) retorna o comprimento da lista.

Na linha 20 é verificado se o índice chave do dicionário (fornecido pelo campo cidade da classe registro) ainda não foi atribuído ao dicionário, caso não tenha sido, ele é então atribuído ao dicionário com seu valor específico (fornecido pelo campo valor da

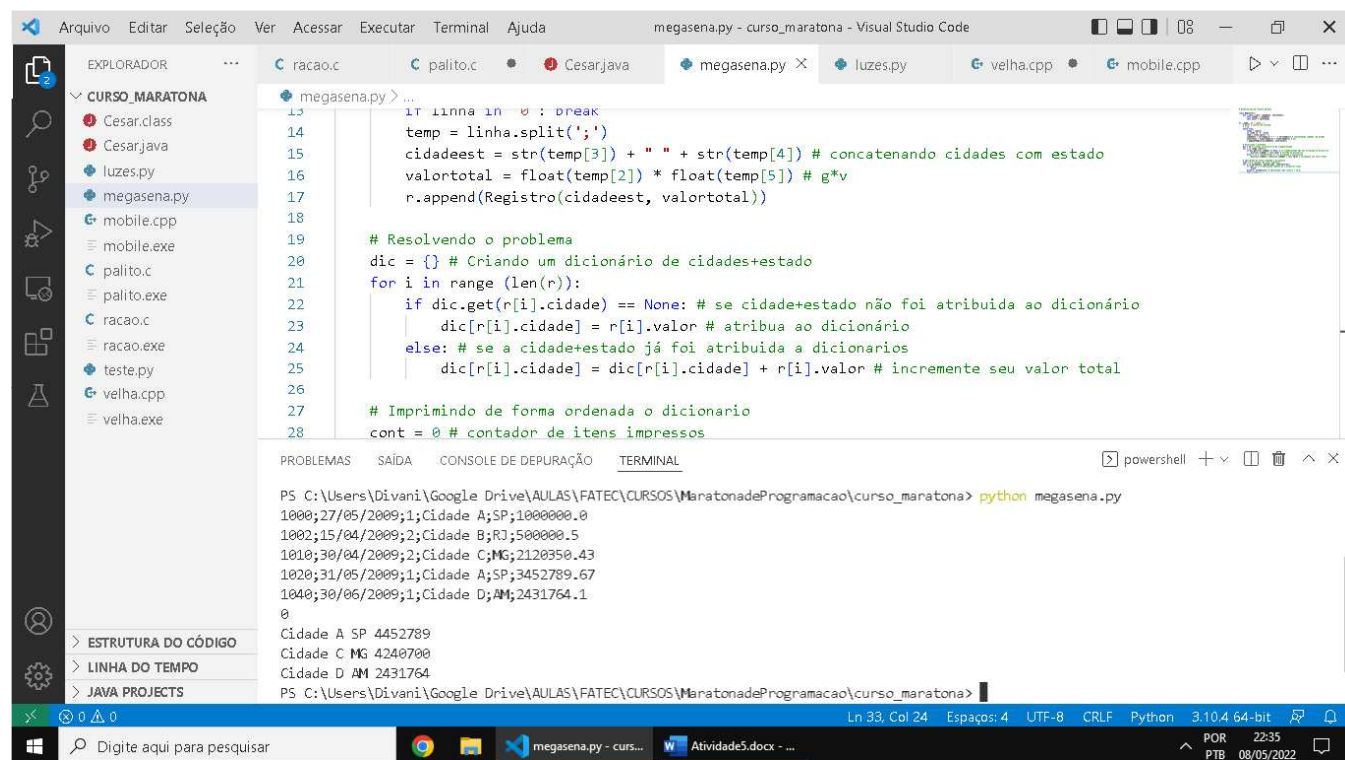
classe registro), como demonstrado na linha 21. Porém, se o índice chave foi atribuído ao dicionário (comando else da linha 22), então, some seu valor específico a ele próprio, como demonstrado na linha 23.

Tendo o problema sido resolvido, falta agora a impressão do **índice e conteúdo do dicionário**. Para isso é usado o comando da linha 27 que percorre o dicionário de maneira ordenada, usando como ordem seus valores específicos, do menor para o maior valor. Porém, como desejamos a ordem inversa, do maior para o menor valor, fazemos uso do comando `reverse=True`.

Na linha 30 temos então a impressão do índice `i` seguido do conteúdo na tela, como o conteúdo estava calculado como float, tem-se o comando `int()` para conversão de real para inteiro antes da impressão, para ficar de acordo com o enunciado do programa.

Nota-se também que na linha 26 tem-se a declaração e atribuição do valor zero a um contador denominado `cont`, as linhas 28, 29 e 31 são responsáveis pela verificação da contagem de linhas impressas na saída, que como especificado pelo enunciado deve ser igual a 3.

Terminal com saída teste:



```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda megasena.py - curso_maratona - Visual Studio Code
```

EXPLORADOR

- CURSO_MARATONA
 - Cesar.class
 - Cesar.java
 - luzes.py
 - megasena.py
 - mobile.cpp
 - mobile.exe
 - palito.c
 - palito.exe
 - racao.c
 - racao.exe
 - teste.py
 - velha.cpp
 - velha.exe

```
13 if linha in r: break
14 temp = linha.split(';')
15 cidadeest = str(temp[3]) + " " + str(temp[4]) # concatenando cidades com estado
16 valortotal = float(temp[2]) * float(temp[5]) # g*v
17 r.append(Registro(cidadeest, valortotal))
18
19 # Resolvendo o problema
20 dic = {} # Criando um dicionário de cidades+estado
21 for i in range(len(r)):
22     if dic.get(r[i].cidade) == None: # se cidade+estado não foi atribuída ao dicionário
23         dic[r[i].cidade] = r[i].valor # atribua ao dicionário
24     else: # se a cidade+estado já foi atribuída a dicionários
25         dic[r[i].cidade] = dic[r[i].cidade] + r[i].valor # incremente seu valor total
26
27 # Imprimindo de forma ordenada o dicionario
28 cont = 0 # contador de itens impressos
```

PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO TERMINAL

```
PS C:\Users\Divani\Google Drive\AULAS\FATEC\CURSOS\MaratonadeProgramacao\curso_maratona> python megasena.py
1000;27/05/2009;1;Cidade A;SP;1000000.0
1002;15/04/2009;2;Cidade B;RJ;500000.5
1010;30/04/2009;2;Cidade C;MG;2120350.43
1020;31/05/2009;1;Cidade A;SP;3452789.67
1040;30/06/2009;1;Cidade D;AM;2431764.1
0
Cidade A SP 4452789
Cidade C MG 4240700
Cidade D AM 2431764
PS C:\Users\Divani\Google Drive\AULAS\FATEC\CURSOS\MaratonadeProgramacao\curso_maratona>
```

Ln 33, Col 24 Espaços: 4 UTF-8 CRLF Python 3.10.4 64-bit

POR 22:35 08/05/2022