

resitev

January 28, 2024

1 Poprečja

Imamo datoteko, ki je videti, recimo, tako

```
Ana 4 2 3 4 5
Berta 4 3
Cilka 5 5 3 5
Dani 4 3 5
Ema 3
```

Vsaka vrstica vsebuje ime šolarke, sledijo njene ocene (pri matematiki).

V splošnem lahko datoteka vsebuje tudi kakšna druga imena. Tudi ime datoteke ni vedno enako; podano je kot argument funkciji.

Napiši funkcije

- `redovalnica(ime_datoteke)` prejme ime datoteke in vrne slovar, katerega ključi so imena, pripadajoči seznam pa ocene (kot števila, ne nizi).

Za gornje podatke mora vrniti `{"Ana": [4, 2, 3, 4, 5], "Berta": [4, 3], "Cilka": [5, 5, 3, 5], "Dani": [4, 3, 5], "Ema": [3]}`

- `poprecne_ocene(ime_datoteke)` prejme ime datoteke in vrne slovar, katerega ključi so imena, vrednosti pa poprečne ocene, zaokrožene navzgor. Če imamo neko (ne nujno celo) število `x`, dobimo navzgor zaokroženo celo število s klicem `int(ceil(x))`, pri čemer funkcijo `ceil` (zaokrožanje navzgor) uvozimo z `from math import ceil`.

Za gornje ocene mora vrniti `{'Ana': 4, 'Berta': 4, 'Cilka': 5, 'Dani': 4, 'Ema': 3}`.

- `zapisi_poprecja(vhod, izhod)` prejme ime vhodne in izhodne datoteke. Iz vhodne prebere podatke, v izhodno pa izpiše imena in poprečja.

V gornjem primeru mora v datoteko z imenom `izhod` zapisati

```
Ana 4
Berta 4
Cilka 5
Dani 4
Ema 3
```

Metoda `write` sprejme le en argument in ta mora biti niz. Če imamo `ime` in poprečno oceno `x`, bomo to sestavili v niz takole: `ime + " " + str(x)`. (Prihodnji teden se bomo to naučili narediti preprosteje.)

Testi so napisani tako, da za branje datoteke uporabijo kar vašo funkcijo `redovalnica`. Če boste napisali kaj neumnega, se torej utegne zgoditi, da bo napako javila `redovalnica`.

Testi si bodo izmislili naključno ime izhodne datoteke. Če bo vse v redu, jo bodo potem pobrisali. Če ni vse v redu, jo bodo pustili, tako da jo lahko pogledate in vidite, kaj ste napisali vanjo.

1.1 Rešitev

1.1.1 redovalnica

Gremo čez vrstice. Vsako razbijemo s `split()` in shranimo v `polja`. Prvi element, `polja[0]` vsebuje ime, ostali, `polja[1:]` pa so ocene. Iz ostalih sestavimo seznam - se pravi seznam, ki vsebuje `int(x)` za vsak iz `polja[1:]`. To damo v slovar.

```
[1]: def redovalnica(ime_datoteke):
    red = {}
    for vrstica in open(ime_datoteke):
        polja = vrstica.split()
        red[polja[0]] = [int(x) for x in polja[1:]]
    return red

redovalnica("ocene.txt")
```

```
[1]: {'Ana': [4, 2, 3, 4, 5],
      'Berta': [4, 3],
      'Cilka': [5, 5, 3, 5],
      'Dani': [4, 3, 5],
      'Ema': [3]}
```

1.1.2 poprecne_ocene

Pomagali si bomo s funkcijo `redovalnica`; šli bomo prek vseh parov (ključ, vrednost) iz tega slovarja (torej: `items()`). Poimenovali jih bomo `ime` in `ocene`. Torej:

```
for ime, ocene in redovalnica(ime_datoteke).items()
```

Sestaviti moramo slovar, katerega ključi bodo imena, vrednosti pa poprečne ocene. Če so ocene v seznamu `ocene`, bomo poprečje izračunali s `sum(ocene) / len(ocene)`. Vsako oceno moramo zaokrožiti navzgor (`ceil`) in spremeniti v celo število (`int`). Torej:

```
[2]: from math import ceil

def poprecne_ocene(ime_datoteke):
    return {ime: int(ceil(sum(ocene) / len(ocene)))
            for ime, ocene in redovalnica(ime_datoteke).items()}

poprecne_ocene("ocene.txt")
```

```
[2]: {'Ana': 4, 'Berta': 4, 'Cilka': 5, 'Dani': 4, 'Ema': 3}
```

1.1.3 zapisi_poprecje

Tu ni kaj modrovati. Odpremo datoteko in zapisujemo.

```
[3]: def zapisi_poprecja(vhod, izhod):  
      f = open(izhod, "wt")  
      for ime, popr in poprecne_ocene(vhod).items():  
          f.write(ime + " " + str(popr) + "\n")
```