

resitev

January 28, 2024

1 Miklavževa darila

V poddirektoriju trenutnega direktorija se nahaja direktorij pisma, v katerem so otroška pisma Miklavžu. (Torej: svoj program boste pisali v nekem direktoriju, na primer, da bo v datoteki `c:/programiranje1/naloga7/resitev.py`. Pisma so v poddirektoriju `c:/programiranje1/naloga7/pisma`, tako da je v njem, recimo `c:/programiranje1/naloga7/pisma/ana.txt`.) Ime datoteke je vedno sestavljeno iz imena otroka, ki mu sledi `.txt`. Moja datoteka, recimo, bi bila `janez.txt`.

V datotekah ni drugega kot seznam stvari (po ena v vsaki vrstici), ki si jih je otrok zaželel. (Otroci so se odločili, da letos ne bodo obremenjevali sv. Miklavža s spiskom obljub, za katere je jasno, da se jih ne bodo držali.) Vsebina datoteke `dani.txt` je, recimo,

```
glavnik
orglice
zvezek
```

Napišite naslednje funkcije:

- `preberi_datoteko(ime)` prejme ime otroka (npr. `"dani"`) in vrne množico stvari, ki si jih otrok želi. Za to mora prebrati ustrezno datoteko v poddirektoriju `"pisma"`.

```
>>> preberi_datoteko("dani")
{'orglice', 'zvezek', 'glavnik'}
```
- `pisci()` ne dobi nobenih argumentov, vrne pa množico z *imeni* vseh otrok, ki so napisali pisma Miklavžu. Ta dobi iz imen datotek.

```
>>> pisci()
{'cilka', 'ema', 'ana', 'dani', 'berta'}
```
- `zelje(imena)` prejme množico imen otrok in vrne slovar, katerega ključi so ta imena, pripadajoče vrednosti pa množice stvari, ki si jih je zaželel otrok s tem imenom. Funkcija pridobi podatke iz datotek.

```
>>> zelje(imena)
{'ana': {'sanke', 'pero', 'zvezek'}, 'cilka': {'notni zvezek', 'lok za violino'},
 'ema': {'copati', 'glavnik'}, 'berta': {'copati', 'zvezek'},
 'dani': {'zvezek', 'orglice', 'glavnik'}}
```
- `prestej_stvari(spiski)` prejme spiske v obliki slovarja, kakršnega vrača prejšnja funkcija, in vrne slovar, katerega ključi so imena stvari, ki so si jih zaželeli ti otroci, vrednosti pa so število teh stvari.

Če so si trije otroci zaželeli zvezek, bo ključu "zvezek" pripadala vrednost 3.

```
>>> stvari = prestej_stvari(spiski)
{'zvezek': 3, 'glavnik': 2, 'copati': 2, 'pero': 1, 'notni zvezek': 1,
'sanke': 1, 'lok za violino': 1, 'orglice': 1}
```

Namesto slovarja je dovoljeno uporabiti tudi kaj, kar je v sorodu z njim in se obnaša enako kot slovar. :)

- `prosilci(spiski)` prejme spiske v obliki slovarja, kot ga vrača funkcija `zelje`, in vrne slovar, katerega ključi so darila, vrednosti pa množice otrok, ki so prosili za to darilo.

1.0.1 Rešitev

`preberi_datoteko` sestavi množico, vrstico za vrstico bere datoteko in dodaja stvari v množico, mimogrede odstripavši odvečni `\n`.

```
[1]: def preberi_datoteko(ime: str):
      stvari = set()
      for stvar in open("pisma/" + ime + ".txt".format(ime)):
          stvari.add(stvar.strip("\n"))
      return stvari
```

Malo kasneje, po koncu predavanj, znamo tudi boljše - preprosteje sestavimo ime in, predvsem, preprosteje sestavimo množico.

```
[2]: def preberi_datoteko(ime):
      return {vrstica.strip() for vrstica in open("pisma/{ime}.txt")}
```

Funkcija `pisci` je podobna: sprehodi se po direktoriju in zлага imena datotek v množico, mimogrede odstranivši odvečni `.txt`.

```
[3]: def pisci():
      otroci = set()
      for ime in os.listdir("pisma"):
          otroci.add(ime[:-4])
      return otroci
```

In, glej no, tudi to znamo po koncu predavanj narediti preprosteje:

```
[4]: def pisci():
      return {ime[:-4] for ime in os.listdir("pisma")}
```

Funkcija `zelje` gre prek imen otrok; v slovar doda ime kot ključ in podatke iz datoteke kot vrednost.

```
[5]: def zelje(imena):
      spiski = {}
      for ime in imena:
          spiski[ime] = preberi_datoteko(ime)
      return spiski
```

Začuda znamo po predavanjih tudi to narediti preprosteje.

```
[7]: def zelje(imena):  
      return {ime: preberi_datoteko(ime) for ime in imena}
```

Za štetje stvari bomo uporabili defaultdict.

```
[8]: def prestej_stvari(spiski):  
      spisek = defaultdict(int)  
      for zelje in spiski.values():  
          for zelja in zelje:  
              spisek[zelja] += 1  
      return spisek
```

V razdelku “preprostejše” tokrat ni ničesar.

Funkcija `prosilci` pa je praktično enaka gornji, le da imamo namesto števil množice in namesto `+= 1` imamo `add(ime)`. Pa namesto prek `spiski.values()` gremo prek `spiski.items()`, da poleg želja dobimo še želiteља.

```
[9]: def prosilci(spiski):  
      spisek = defaultdict(set)  
      for ime, zelje in spiski.items():  
          for zelja in zelje:  
              spisek[zelja].add(ime)  
      return spisek
```