

11. Datoteke

Poznamo dve vrsti datotek: znakovne in dvojiške. Razlika je v načinu, kako so zapisani posamezni podatki. V primeru znakovnih datotek so zapisani z zaporedjem znakov, v primeru dvojiških datotek pa tako, kot so zapisani v pomnilniku. Preden z datoteke kaj preberemo ali nanjo kaj zapišemo, jo moramo odpreti. Ko je ne rabimo več, jo moramo zapreti. Pri tem predmetu bomo uporabljali zgolj znakovne datoteke.

11.1. Odpiranje in zapiranje znakovne datoteke

Znakovno datoteko odpremo s funkcijo **open**. Ta vrne objekt, ki predstavlja datoteko in ga potrebujemo pri nadaljnjem delu s to datoteko.

- **open(ime)** ... odpre datoteko z danim imenom za branje
- **open(ime, 'r')** ... odpre datoteko z danim imenom za branje
- **open(ime, 'w')** ... odpre datoteko z danim imenom za pisanje
- **open(ime, 'a')** ... odpre datoteko z danim imenom za dodajanje
- **open(ime, 'r+')** ... odpre datoteko z danim imenom za branje in pisanje

Po končanem delu moramo datoteko zapreti. Objekt, ki ga vrne funkcija **open**, si moramo zapomniti. To lahko storimo na enega izmed spodnjih dveh načinov. V drugem primeru nam datoteke ni treba zapirati, saj se to zgodi avtomatsko, tudi če pri delu z datoteko pride do kakšne napake.

- ```
f = open(ime)
nekaj naredimo z datoteko f
f.close()
```
- ```
with open(ime) as f:
    # nekaj naredimo z datoteko f
```

11.2. Branje z znakovne datoteke

Z znakovne datoteke lahko beremo s pomočjo različnih funkcij. Če katera od funkcij nima več kaj prebrati, vrne prazen niz. Prebrane vrstice vsebujejo tudi morebiten znak za skok v novo vrsto na koncu niza.

- **f.read()** ... vrne niz z vsebino cele datoteke
- **f.read(n)** ... prebere največ n znakov iz datoteke od trenutnega položaja naprej in vrne niz teh znakov
- **f.readline()** ... vrne niz z vsebino naslednje vrstice
- **f.readlines()** ... vrne seznam nizov, ki predstavljajo posamezne vrstice

Najbolj priporočljivo pa je branje z uporabo zanke `for`.

```
for line in f: ... zanka po vrsticah datoteke f
```

11.3. Pisanje na znakovno datoteko

- **f.write(...)**
- **print(..., file=f)** ... zapiše dane vrednosti na datoteko `f`.

11.4. ASCII proti Unicode

- `open(ime, mode = "r", encoding="utf-8")` Če želimo uporabljati datoteke, ki vsebujejo znake, ki je koda `ascii` ne premore, je priporočljiva uporaba imenovane spremenljivke `encoding` v stavku `open`.

11.5. Dodatne metode za delo z datotekami

Pri delu z datotekami prav pridejo tudi naslednje metode in atributi:

- **f.seek(n, položaj)** ... kazalec v datoteki se premakne za n mest glede na položaj. Veljavne vrednosti za položaj so
 - 0: začetek datoteke,
 - 1: trenutni položaj,
 - 2: konec datoteke.

Za tekstovne datoteke lahko navedemo poljubno n samo, če je položaj enak 0, v preostalih dveh primerih mora n nujno biti 0. Klic te metode vrne trenutni položaj po izvedbi metode.

- **f.tell()** ... vrne trenutni položaj v datoteki.
- **f.closed** ... pove, ali je datoteka zaprta.
- **f.name** ... vrne ime datoteke.

S klicem **f.seek(0,2)** lahko ugotovimo velikost datoteke.

11.6. Naloge.

1. Napišite funkcijo `beri(file)`, ki prebere vse vrstice znakovne datoteke z imenom "file" in jih prepíše na zaslon.
2. Napišite funkcijo `prepis(f1,f2)`, ki datoteko `f1` prepíše na datoteko `f2`.
3. Napišite program, ki izračuna absolutne frekvence znakov z izbrane znakovne datoteke. Privzemite, da je ime datoteke z besedilom oblike `nekaj.txt`. Pare znak, frekvenca izpišite na datoteko z imenom `nekaj.frek.txt`.
4. Napišite program, ki na datoteko izpiše znake in znakovne kode znakov unicode. Koliko znakov pa ima unicode?