Programiranje 1

Ulaz i izlaz na programskom jeziku Python

Univerzitet u Beogradu
Elektrotehnički fakultet
2019/2020.

Koncept ulaza i izlaza podataka

- Uključuje razmenu podataka računara sa okolinom
- Podrazumevano se koriste standarni ulaz i standardni izlaz
  - Standardni ulaz: tastatura
  - Standardni izlaz: monitor
- Mnoštvo ulazno-izlaznih uređaja i drugih izvora podataka
  - Hard-diskovi, optički uređaji, USB diskovi...
  - Tokovi podataka sa interneta
- Podaci su reprezenzovani na različite načine
  - Potreba za interpretacijom

Rad sa standardnim ulazom i izlazom

- Nizovi znakova razdvojeni upravljačkim znacima
- Funkcija **`input()`** učitava string sa tastature
  - Uz eventualno ispisivanje poruke
- Funkcija **`print()`** ispisuje sadržaj na ekran
  - Uz mogućnost formatiranja ispisa
- Obe funkcije obrađene ranije
  - Za rad sa standardnim ulazom i izlazom
- Rad sa standardnim ulazom i izlazom omogućava obradu relativno malih količina podataka

Datoteke

- Datoteke (fajlovi) sadrže podatke smeštene na trajnim (spoljnim) memorijama
  - Tekst, slika, zvuk, video, izvorni kod, mašinski kod...
- Ista datoteka može biti čitana i pisana od strane različitih programa
- Po načinu zapisa podataka se dele na:
  - Tekstualne datoteke
  - Binarne datoteke
- Programer mora biti svestan načina zapisa podataka unutar datoteke
  - Da bi korektno vršio čitanje i pisanje
- Postoji specijalan znak *End-of-File* (EOF) koji označava kraj datoteke

Tekstualne datoteke (1)

- Tekstualna datoteka sadrži linije teksta
  - Najčešće kodirane u ASCII kodu
  - Postoje i drugi kodni rasporedi: UTF-8, UTF-16, Unicode
- Svaka linija se završava oznakom kraja linije
- Terminator može da bude jedan karakter ili kombinacija dva:
  - CR – *carriage return*, LF – *line feed*
  - Windows, DOS – <CR><LF> (**`"\r\n"`**)
  - Unix – <LF> (**`"\n"`**)
  - Mac OS – <CR> (**`"\r"`**)
- ASCII kod <CR> je 13, a <LF> je 10

Tekstualne datoteke (2)

- Karakterišu ih uređenost u redove i postojanje belih znakova
  - Beli znaci: razmak (*space*), tabulacije
- Primeri:
  - Bilo koja *plain-text* datoteka, formirana editorom teksta
    - **`primer.txt, README`**
- Izvorni kodovi programa u nekom programskom jeziku
  - **`program.pas, skript.py, helloworld.c`**
- HTML dokumenti
- XML dokumenti

Binarne datoteke

- Binarna datoteka sadrži podatke zapisane u binarnoj predstavi
  - Samo programer ili dizajner zna kako da ih interpretira
    - U skladu sa formatom datoteke
  - Različiti programi ili računari mogu istu datoteku na različite načine da interpretiraju
- Primeri:
  - Izvršni programi: **`explorer.exe`**
  - Datoteke koje sadrže slike: **`moon.jpg, logo.png, animation.gif`**
  - Audio datoteke: **`Riders on the storm.mp3`**
- Svaka datoteka se može tretirati kao binarna

Smeštanje datoteke

- Datoteka se smešta negde u okviru fajl sistema
  - Fajl sistem je određen korišćenim operativnim sistemom
- Identifikuje se imenom (uz opcionu ekstenziju)
  - **`skripta.py`**, **`test.txt`**, **`explorer.exe`**, **`README`**
- Pristupa joj se putem odgovarajuće putanje u okviru fajl sistema
  - Apsolutne putanje u odnosu na koren fajl sistema
    - Windows: **`C:\AppData\readme.txt`**
    - Linux: **`/home/user/appdata/readme.txt`**
  - Relativne putanje u odnosu na mesto pokretanja programa
    - **`readme.txt`** – u istom direktorijumu
    - **`manual\readme.txt`** – u poddirektorijumu **`manual`**
    - **`..\manual\readme.txt`** – u poddirektorijumu **`manual`** koji se nalazi u naddirektorijumu u odnosu na tekući

Koraci pri radu sa datotekom

- Osnovni koraci u radu sa datotekom su:
  - Otvaranje u odgovarajućem režimu
  - Pristup
    - Čitanje ili pisanje
  - Zatvaranje datoteke
- Postoje razlike pri radu sa tekstualnim i binarnim datotekama
  - U skladu sa njihovom organizacijom

Otvaranje datoteke

- Datoteka se mora otvoriti u odgovarajućem režimu pre pristupanja
- Koristi se funkcija **`open()`**:

  `open(`*`ime_datoteke`*`, `*`režim_pristupa`*`)`

  - **`ime_datoteke`** – string koji sadrži ime i (opcionu) putanju datoteke koja se otvara
  - ***`režim_pristupa`*** – definiše način (režim) otvaranja datoteke
- Funkcija vraća **`file`** objekat koji se koristi za rad sa datotekom

Režimi otvaranja datoteke

- Definisani drugim argumentom funkcije **`open()`**
- Prosleđuje se string koji sadrži opis režima
  - **`"r"`** – čitanje (*read*)
    - Podrazumevani režim, može se izostaviti drugi argument
  - **`"w"`** – upis (*write*)
    - Ukoliko ne postoji, napravi se nova datoteka nulte veličine
    - Ukoliko postoji, obriše se, pa se ponovo kreira prazna
  - **`"a"`** – dodavanje na kraj (*append*)
- Za binarne datoteke se dodaje slovo **`b`** na režim
  - **`"rb", "wb", "ab"`**
- Moguće i istovremeno čitanje i pisanje dodavanjem **`+`**
  - **`"r+", "w+", "a+", "rb+", "wb+", "ab+"`**

Zatvaranje datoteke

- Poziva se funkcija **`close()`** za objekat **`f`** koji predstavlja otvorenu datoteku
  - Može se koristiti i polje **`f.closed`** za proveru da li je zatvorena datoteka
- Dobra praksa je korišćenje naredbe with sa datotečkim objektima
  - Obezbeđuje automatsko zatvaranje datoteke kada se blok koda završi

```
with open('workfile') as f:
      read_data = f.read()
>>> f.closed
True
```

Čitanje datoteke (1)

- Veći broj funkcija za čitanje iz datoteke
  - Pozivaju se za objekat **`f`** koji predstavlja otvorenu datoteku
- **`f.read([size])`**
  - Čita **`size`** znakova (bajtova) iz datoteke i vraća string
  - Ukoliko se parametar **`size`** izostavi, čita celu datoteku
  - Kada dođe do kraja reda vraća string koji sadrži **`"\n"`**
  - Kada dođe do kraja datoteke vraća prazan string
- **`f.readline()`**
  - Čita jednu liniju teksta i vraća string
  - Isto ponašanje za kraj reda i datoteke kao prethodna funkcija

Čitanje datoteke (2)

- **`f.readlines()`**
  - Vraća listu stringova, red po red
- Alternativno,
  kroz objekat datoteke se može iterirati **`for`** petljom

```
for line in f:
    print(line, end='')
```

- Ukoliko je potrebno iterirati
  kroz pojedinačne znakove:

```
for line in f:
    for c in line:
        print(c, end='')
```

Čitanje datoteke (3)

- Primer sadržaja datoteke **`test.txt`**

```
The quick
brown fox
jumps over the lazy dog
0123456789
```

- Primer koda koji čita datoteku:

```
with open('test.txt') as f:
    for line in f:
        print(line, end='')
```

Pisanje datoteke (1)

- **`f.write(`*`string`*`)`**
  - Upisuje u datoteku **`f`** zadati string
  - Podaci koji nisu u obliku stringa moraju najpre da se konvertuju
  - Ne dodaje znak za novi red
- **`f.writelines(`*`linije`*`)`**
  - Upisuje u datote **`f`** zadatu listu linija
  - Ne dodaje znak za novi red na kraju linije
- **`print(`*`string`*`, file=f)`**
  - Podrazumevani argument **`file`** incijalizuje objektom **`f`**
  - Dostupna sva formatiranja i opcije funkcije **`print()`**

Pisanje datoteke (2)

- Primer korišćenja **`write()`** funkcije

```
with open('C:\\Temp\\test2.txt', 'w') as f:
    year = 2000
    f.write("Hi there year "+str(year)+" !\n")
    f.write("Python is a great language!\n")
```

- Primer korišćenja **`print()`** funkcije

```
with open('C:\\Temp\\test2.txt', 'w') as f:
    year = 2000
    print("Hi there year", year, "!", file=f)
    print("Python is a great language!", file=f)
```

- Sadržaj datoteke **`test2.txt`** nakon upisa:

```
Hi there year 2000 !
Python is a great language!
```

Pisanje datoteke (3)

- Primer – upisivanje liste realnih brojeva u datoteku:

```
numbers = [1.22, 123.23, 3E-2]
num_out = open('num_out.txt', 'w')
for num in numbers:
    num_out.write("{}\n".format(num))
num_out.close()
```

num_out.txt:

1.22
123.23
0.03

- Alternativno, korišćenjem pravila:

```
num_out = open('num_out.txt', 'w')
num_out.writelines(\
["{}\n".format(num) for num in numbers])
num_out.close()
```

Uslužne funkcije za rad sa datotekom

- Određivanje trenutne pozicije u okviru datoteke

  `f.tell()`

  - Pozicija se određuje u broju bajtova u odnosu na početak datoteke

- Promena trenutne pozicije u okviru datoteke

  `f.seek(offset[, from])`

  - Argument **`offset`** se odnosi na broj bajtova za koji se vrši pomeranje
  - Opcioni argument **`from`** uvodi repernu poziciju
    - 0 – početak datoteke, 1 – trenutna pozicija, 2 – kraj datoteke
    - Za tekstualne datoteke dozvoljena samo 0 i **`seek(0, 2)`**

Rad sa fajl sistemom (1)

- Moguće kroz korišćenje modula **`os`**
  - Koristi odgovarajuće usluge operativnog sistema
- Preimenovanje datoteke:

```
os.rename(current_file_name,
          new_file_name)
```

- Uklanjanje (brisanje) datoteke

```
os.remove(file_name)
```

- Dohvatanje trenutnog direktorijuma

```
os.getcwd()
```

Rad sa fajl sistemom (2)

- Pravljenje novog direktorijuma u postojećem

```
os.mkdir(new_dir)
```

- Promena aktivnog (tekućeg) direktorijuma
  - Korišćenjem apsolutne ili relativne putanje

```
os.chdir(new_dir)
```

- Uklanjanje (brisanje) direktorijuma

```
os.rmdir(dir_name)
```

- Dohvatanje liste datoteka i direktorijuma
  - U tekućem direktorijuma bez argumenta
  - U zadatom direktorijumu navođenjem argumenta `path`

```
os.listdir(path=None)
```

Rad sa fajl sistemom (3)

- Za potrebe rada sa putanjama se koristi zaseban modul **`os.path`**
- Provera da li je putanja apsolutna

  `os.path.isabs(p)`
- Vraćanje putanje direktorijuma za argument

  `os.path.dirname(p)`
- Vraćanje imena fajla za argument

  `os.path.basename(p)`

Rad sa fajl sistemom (4)

- Provera postojanja datoteke

```
os.path.exists(p)
```

- Provera da li je argument datoteka

```
os.path.isfile(p)
```

- Provera da li je argument direktorijum

```
os.path.isdir(p)
```

Literatura - knjige

- M. Kovačević, Osnove programiranja u Pajtonu, Akademska misao, Beograd, 2017.
- M. Lutz, Learning python: Powerful object-oriented programming, 5th edition, O'Reilly Media, Inc., 2013.
- J. Zelle, Python Programming: An Introduction to Computer Science, 3rd Ed., Franklin, Beedle & Associates, 2016.
- D. Beazley, B. K. Jones, Python Cookbook, 3rd edition, O'Reilly Media, 2013.
- A. Downey, J. Elkner, C. Meyers, How To Think Like A Computer Scientist: Learning With Python, free e-book

Literatura – *online* izvori

- Python 3.8.0 documentation, https://docs.python.org/3/index.html
- Colin Morris, 7-day Python course, https://www.kaggle.com/learn/python
- Learn Python, Basic tutorial, https://www.learnpython.org/
- TutorialsPoint, Python tutorial https://www.tutorialspoint.com/python/index.htm