ALGEBRA

PROGRAMIRANJE

Predavanje 14

Ishod 6

1

UPRAVLJANJE POGREŠKAMA

ALGEBRA

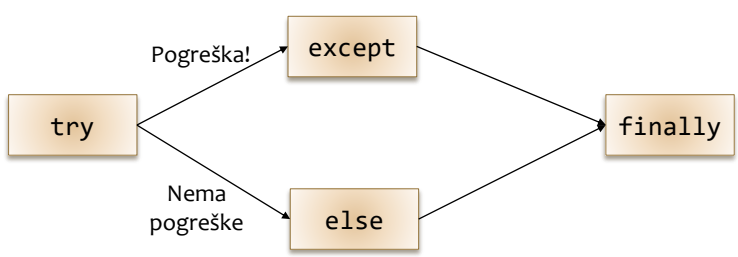
2

Uvod

- Kad se dogodi pogreškom prilikom izvršavanja programa, Python nas obavještava o tome podižući iznimku
  - engl. *raise an exception*
- Iznimka je objekt kojeg možemo uhvatiti (engl. *catch*) i na taj način upravljati pogreškom koja se dogodila
  - Možemo je zapisati u zapisničku (engl. *log*) datoteku i obavijestiti korisnika da se dogodila pogreška (češće)
  - Možemo je pokušati popraviti (rjeđe)

Naredbe `try`, `except`, `else` i `finally`

- `try` definira blok koji može uzrokovati pogrešku
- `except` definira blok koji se izvršava ako se dogodi pogreška
- Naredba `else` definira blok koji se izvršava ako se ne dogodi pogreška
- Naredba `finally` definira blok koji se izvršava uvijek



Dva primjera

```
print('A')
print('B')
print(17 / 2)
print('C')
print('D')
print('Idemo dalje...')
```

```
print('A')
print('B')
print(17 / 0)
print('C')
print('D')
print('Idemo dalje...')
```


ALGEBRA

5

Dva primjera

```
try:
    print('A')
    print('B')
    print(17 / 2)
    print('C')
    print('D')
except:
    print('Dogodila se pogreška')
else:
    print('Sve OK')
finally:
    print('E')

print('Idemo dalje...')
```

```
try:
    print('A')
    print('B')
    print(17 / 0)
    print('C')
    print('D')
except:
    print('Dogodila se pogreška')
else:
    print('Sve OK')
finally:
    print('E')

print('Idemo dalje...')
```


ALGEBRA

6

Više detalja o iznimki

```
try:
    brojevi = [ 1, 2, 3 ]
    print(brojevi[3])
except:
    print("Dogodila se neka
greška, ali ne znamo koja")
```

```
try:
    brojevi = [ 1, 2, 3 ]
    print(brojevi[3])
except Exception as exc:
    print(type(exc))
    print(exc.args)
    print(exc)
```

```
try:
    brojevi = [ 1, 2, 3 ]
    print(brojevi[3])
except IndexError as exc:
    print("Neispravan indeks!")
```

```
try:
    brojevi = [ 1, 2, 3 ]
    print(brojevi[2])
    print(17 / 0)
except IndexError as exc:
    print("Neispravan indeks!")
except Exception as exc:
    print("Greška: " + str(exc))
```


ALGEBRA

7

TEKSTUALNE DATOTEKE

ALGEBRA

8

Uvod

- Funkcije za rad s tekstualnim datotekama u Pythonu su dio ugrađenog imenskog prostora
- Postupak rada je sljedeći:
  1. Otvoriti datoteku za čitanje ili pisanje
  2. Koristiti datoteku za čitanje ili pisanje
  3. Zatvoriti datoteku
- Pošto rad s datotekama može uzrokovati niz različitih pogrešaka, moramo cijeli postupak zaštititi korištenjem `try..except..finally` naredbi

Otvaranje datoteke

- Za otvaranje datoteke koristimo funkciju `open()`:

```
open('datoteka', 'način')
```

- Za datoteku možemo navesti relativnu ili apsolutnu putanju:

```
'datoteka.txt'
'./datoteka.txt'
'../../datoteka.txt'
'/home/miro/datoteka.txt'
'd:/temp/datoteka.txt'
```

- Za način biramo jedan od dva najkorištenija:
  - `'r'` – koristimo kad želimo datoteku otvoriti za čitanje
  - `'w'` – koristimo kad želimo datoteku otvoriti za pisanje
- Funkcija vraća objekt tipa `file`

Zatvaranje datoteke

- Datoteka koristi resurse operacijskog sustava
  - Pravilo kod programiranja: resurse operacijskog sustava **uzmite što kasnije** možete i **otpustite što ranije** možete
- Datoteku zatvaramo na jedan od dva načina:
  - Pozivanjem metode **`close()`** na objektu:

    ```
    dat1.close();
    ```

  - Puštanjem da *Garbage Collector* pokupi objekt
    - Nije dobro jer se to može desiti dosta kasnije
- Datoteku ćemo uvijek zatvarati pozivanjem metode `close()`

Potencijalni problemi

- Rad s datotekom može biti neuspješan iz niza razloga
  - Datoteka ne postoji
  - Nemamo prava na datoteku
  - Datoteka je već otvorena u ekskluzivnom načinu
  - Datoteka na mrežnom disku je uspješno otvorena, ali za vrijeme čitanja mrežni disk postane nedostupan
  - Datoteka na USB disku je uspješno otvorena, ali korisnik isključi USB disk

Zaštita postupka

- Da bismo zaštitili postupak, možemo postupiti ovako:

```
try:
    f = open('datoteka1.txt', 'r ili w')

    try:
        <Čitamo/pišemo iz/u datoteke>
    finally:
        f.close()
except:
    print('Dogodila se pogreška pri otvaranju datoteke')
```


Zaštita postupka

- Prethodni kôd ispravno štiti postupak, ali je kompliciran
- Jednostavniji način kako postići sličan efekt je sljedeći:

```
try:
    with open('datoteka1.txt', 'r ili w') as f:
        #<Čitamo/pišemo iz/u datoteke>
except:
    print('Dogodila se pogreška pri otvaranju datoteke')
```

- U ovom pristupu metoda `close()` se poziva automatski

ČITANJE I PISANJE U TEKSTUALNU DATOTEKU

ALGEBRA

15

Čitanje iz datoteke

- Čitanje podataka iz datoteke ćemo raditi liniju po liniju
- Primjer:

```
try:
    with open('datoteka.txt', 'r') as f:
        for linija in f:
            print(linija)
except:
    print('Dogodila se pogreška pri otvaranju datoteke')
```


ALGEBRA

16

Čitanje iz datoteke

- Čitanjem svake linije iz datoteke dobivamo string
- Na kraju pročitanog stringa se nalazi \n
- Možemo ga ukloniti metodom `rstrip()`

```
try:
    with open('datoteka.txt', 'r') as f:
        for linija in f:
            print(linija.rstrip())
except:
    print('Dogodila se pogreška pri otvaranju datoteke')
```


Pisanje u datoteku

- Za pisanje u datoteku koristimo metodu `write()`
- Primjer:

```
try:
    with open('datoteka.txt', 'w') as f:
        f.write('Linija 1\n')
        f.write('Linija 2\n')
        f.write('Linija 3\n')
        f.write('Linija 4\n')
        f.write('Linija 5\n')
except:
    print('Dogodila se pogreška pri otvaranju datoteke')
```


Primjeri

1. Napišite program koji od korisnika učitava 5 brojeva i upisuje ih u datoteku, svaki u svoj red.
2. Napišite program koji od korisnika učitava 5 brojeva i upisuje ih u datoteku obrnutim redoslijedom i odvojene zarezima. Iza zadnjeg broja nemojte staviti zarez.
3. Definirajte strukturu za čuvanje podataka o kravama (ime, težina, godina rođenja). Napravite polje od 3 krave, dodijelite im vrijednosti te ih spremite u datoteku tako da svaka krava bude u svom retku, a njeni podaci međusobno odvojeni tabovima ('\t').

ALGEBRA

19

Primjeri

4. Napravite datoteku i u nju upišite 5 cijelih brojeva, svaki u svoj red. Napišite program koji korisniku ispisuje tih 5 brojeva.
5. Napravite datoteku i u nju upišite *n* cijelih brojeva, svaki u svoj red. Napišite program koji korisniku ispisuje sve brojeve.
6. Napravite datoteku i u nju upišite *n* cijelih brojeva, svaki u svoj red. Napišite program koji u novu datoteku upisuje sve parne brojeve iz prve datoteke, svaki u svoj red.
7. Učitajte naziv datoteke od korisnika i ispišite njen sadržaj.

ALGEBRA

20

Primjeri

8. Napišite funkciju koja prima više stringova i radi sljedeće: za svaki primljeni string gleda broj znakova. Ako je manji od 5, upisuje string u datoteku "kratke_rijeci.txt", ako nije, upisuje ga u datoteku "duge_rijeci.txt". Napišite program koji od korisnika učitava onoliko riječi koliko korisnik želi, a nakon toga prosljeđuje riječi u funkciju.
9. Zadajte si listu s podacima o količini padalina u zadnjih 5 dana. Napišite funkciju koja vraća prosječnu i maksimalnu vrijednost padalina, ali tako da uzme samo pozitivne vrijednosti. Ako nema niti jedne pozitivne vrijednosti, neka funkcija vrati -1. Pozovite funkciju te listu i rezultat funkcije spremite u datoteku.

Primjeri

10. Napišite klasu Ribar koja se inicijalizira s rječnikom riba, primjerice { "Losos": 10 , "Pastrva": 20 , "Tuna": 5 }. Klasa treba imati metodu ulovi() koja prima naziv ribe i povećava broj primjeraka za 1. Klasa treba imati metodu prikazi() koja prima naziv ribe i vraća koliko primjeraka te ribe je ulovljeno. Klasa treba imati i metodu pusti() koja prima naziv ribe i smanjuje količinu te ribe za 1. Dodajte ostale metode prema potrebi.

Za sljedeće predavanje

- Ponoviti sve iz ovog predavanja
- Pročitati i isprobati primjere:
  - https://dbader.org/blog/python-file-io
  - https://docs.python.org/3/tutorial/inputoutput.html
  - https://realpython.com/python-exceptions/

ALGEBRA