

Načini debugovanja u programskom jeziku Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović,
Matija Pejić

Matematički fakultet, Beograd

28. april 2020.

Sadržaj

Osnovne tehnike debugovanja u Python-u

- Uvod

- Debugovanje sintaksnih grešaka

- Naučni pristup debugovanju

- Debugovanje print metodom

PDB debager

- Pokretanje iz komandne linije

- Pokretanje iz programa

PyCharm

- Šta je PyCharm

- Tačke prekida i pokretanje Debugger-a

- Opcije Debugger-a

Literatura

- ▶ Greške pri programiranju se svima dešavaju
- ▶ Debugovanje je proces nalaženja i otklanjanja grešaka u programu.
- ▶ Ono podrazumeva sledeće:
 1. Znamo kako program treba da radi
 2. Opažamo da je do бага došlo
 3. Pronalazimo bag
 4. Uklanjamو bag

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debugovanja u
Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Debagovanje sintaksnih grešaka

```
def student(name):  
    students = {  
        'Pera': '107/2016',  
        'Mika': '16/2016',  
        'Laza': '252/2015'  
    }  
  
    print('Index of Pera is ' + studenti[name])  
  
student('Pera')
```

File "primer.py", line 5

```
'Laza': '252/2015'  
    ^
```

SyntaxError: invalid syntax

Debagovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debagovanja u
Python-u

Uvod

Debagovanje sintaksnih
grešaka

Naučni pristup debagovanju

Debagovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Debugovanje sintaksnih grešaka

- ▶ Kada u programu postoji sintaksna greška prevodilac izbacuje izuzetak i ispisuje **poruku o grešci**. Ona sadrži:
 1. Tip greške
 2. Opis greške
 3. Traceback
- ▶ Neki izuzeci se ne mogu izbeći, takve izuzetke hvatamo korišćenjem **try** i **except** bloka
- ▶ Kada se naš program prevede, ali ne dobijamo željeni rezultat, takvu grešku nazivamo **semantičkom greškom**

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Debugovanje sintaksnih grešaka

Naučni pristup debugovanju

Debugovanje print metodom

PDB debugger

Pokretanje iz komandne linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Literatura

Naučni pristup debugovanju

Predstavlja formalan pristup pronalaženju problema koji je zasnovan na sledećim koracima:

1. Posmatraj
2. Napravi hipotezu
3. Predvidi
4. Testiraj
5. Zaključči

Za efikasnu primenu, neophodno je vladati tehnikama:

- ▶ reprodukcije grešaka
- ▶ automatizacijom
- ▶ izolacijom grešaka

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

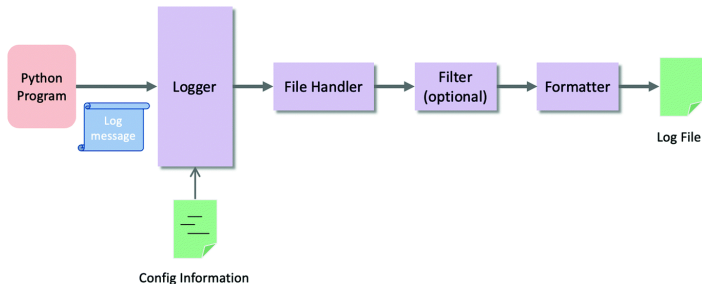
Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Debugovanje print metodom

- ▶ jednostavna, ali moćna metoda
- ▶ biblioteke pprint, logging daju fleksibilnost
- ▶ Neke od klasa u logging biblioteci su:
Logger, LogRecord, Handler, Filter, Formatter



Slika: Rad sa log zapisima

PDB debager

- ▶ Pdb predstavlja interaktivni program za otklanjanje grešaka.
 - ▶ Prati izvršavanje programa korak po korak i pruža pomoć pri rešavanju bagova.
 - ▶ Debugovanje programa pokrećemo:
 1. iz komandne linije
 2. iz samog programa
1. Pokrećemo skript Pdb komandom
`python -m pdb imeprograma.py arg1 arg2`
 2. Umećemo deo koda u program na mesto odakle želimo da započnemo proces debugovanja.
`import pdb; pdb.set_trace()`
`pdb.set_trace()` postavlja debager za pozivajući stek okvir.

Primer

Primer prvi.py

```
my_list = [1,9,13,3,12]
new_list = list(map(lambda x: x*2,my_list))

def sub(a,b):
    print(a)
    return a-b

diff = sub(40,2)
my_list_sum = sum(my_list)
experiment = sum(new_list) / sub(diff,my_list_sum)
```

Debagovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debagovanja u
Python-u

Uvod

Debagovanje sintaksnih
grešaka

Naučni pristup debagovanju

Debagovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

```
> prvi.py(1)<module>()
-> my_list = [1,9,13,3,12]
(Pdb) n
> prvi.py(2)<module>()
-> new_list = list(map(lambda x: x*2,my_list))
(Pdb) n
> prvi.py(3)<module>()
-> def sub(a,b):
(Pdb) n
> prvi.py(6)<module>()
-> diff = sub(40,2)
(Pdb) s
--Call--
> prvi.py(3)<module>()
-> def sub(a,b):
(Pdb) n
> prvi.py(4)sub()
->print(a)
(Pdb) n
40
>prvi.py(5)sub()
->return a-b
(Pdb) n
--Return--
>prvi.py(5)sub->38
->return a-b
```

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Tamo gde želimo da istražujemo postavljamo tačke prekida.

```
import pdb; pdb.set_trace()  
experiment = sum(new_list) / sub(diff,my_list_sum)
```

Ako sada program pokrenemo sa `python prvi.py`, prva linija za izvršavanje korišćenjem debagera biće

```
experiment = sum(new_list) / sub(diff,my_list_sum)
```

U slučaju da dodamo i argumente `-m pdb`, izvršavanje kreće od prve linije.

```
>prvi.py(1)<module>()  
->my_list = [1,9,13,3,12]  
(Pdb)c  
40  
>prvi.py(9)<module>()  
->experiment = sum(new_list) / sub(diff,my_list_sum)  
(Pdb)n  
38  
ZeroDivisionError: division by zero
```

Tačke prekida možemo postavljati i komandom `b` (`break`).

```
(Pdb) b 9
```

Debugovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debugovanja u
Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Šta je PyCharm

PyCharm je integrisano razvojno okruženje koje se koristi za programiranje u jeziku Python. Pruža analizu koda, grafički debager, integraciju sa verzijom kontrolnog sistema(git) i druge pogodnosti.

Bitni pojmovi Pycharm Debugger-a:

1. Detaljno debugovanje
2. Posmatranja
3. Inline Debugger
4. Evaluacija izraza

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debager

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Opcije Debugger-a

Sve opcije Debugger-a se nalaze u Debug Tool Window-u.

Inline Debbuger je opcija koja nam pruža da u vidu komentara u editoru vidimo sve vrednosti promenljivih.

Evaluacija izraza je opcija koja nam omogućava da izračunamo bilo koji izraz sa trenutnim vrednostima promenljivih u kodu, kao i da dodeljujemo vrednosti promenljivama.

Posmatranja su zaseban prozor u kome se nalaze sve promenljive koje su trenutno definisane, kao i njihove vrednosti.

Detaljno debugovanje predstavlja skup opcija za iteriranje kroz kod korak po korak.

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Literatura



Kristian Rother (2017)

Pro Python Best Practices Debugging, Testing and Maintenance



Kalyani Adawadkar (2017)

Python Programming-Applications and Future

International Journal of Advance Engineering and Research
Development



Python 3.8.2 documentation

<https://docs.python.org/3/>



PyCharm

<https://www.jetbrains.com/help/pycharm/>

Debugovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debugovanja u
Python-u

Uvod

Debugovanje sintaksnih
grešaka

Naučni pristup debugovanju

Debugovanje print
metodom

PDB debugger

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura