

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

- Izuzeci u Python-u

- Debugovanje naučnom metodom

- 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 nalaženje i otklanjanje grešaka u programu.
- ▶ Ono podrazumeva sledeće:
 1. Znamo kako program treba da radi
 2. Opažamo da je do бага došlo
 3. Pronalazimo баг
 4. Uklanjamo баг

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debugovanja u
Python-u

Uvod

Izuzeci u Python-u

Debugovanje naučnom
metodom

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

Izuzeci u Python-u

```
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

Izuzeci u Python-u

Debagovanje naučnom
metodom

Debagovanje 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

Izuzeci u Python-u

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** blok.

Naš program se prevede ali ne dobijamo željeni rezultat takvu grešku nazivamo **Semantička greška**.

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Izuzeci u Python-u

Debugovanje naučnom
metodom

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 naučnom metodom

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

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

Da bi efikasno primenili ovaj način debugovanja, potrebno je da dobro vladamo tehnikama reprodukcije grešaka, automatizacijom i izolacijom grešaka, kao i da metodu ne primenjujemo za "lake" greške.

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Izuzeci u Python-u

Debugovanje naučnom
metodom

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

Print je jednostavna, ali moćna metoda za debugovanje. Ako je koristimo adekvatno, one postaje jako sistematična i korisna. Za lepši ispis složenih tipova podataka možemo koristiti biblioteku pprint.

Korisna je i biblioteka logging, gde su nam, izmedju ostalih, na raspolaganju klase:

1. Logger
2. LogRecord
3. Handler
4. Filter
5. Formatter

Debugovanje Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike debugovanja u Python-u

Uvod

Izuzeci u Python-u

Debugovanje naučnom metodom

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

PDB debager

- ▶ Pdb predstavlja interaktivni program za otklanjanje grešaka.
 - ▶ Prati izvršavanje programa korak po korak, pruža pomoć pri rešavanju bagova.
 - ▶ Debugovanje programa pokrećemo:
 1. iz komandne linije
 2. iz samog programa
1. Pokrećemo skript koristeći 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 1.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

Izuzeci u Python-u

Debagovanje naučnom
metodom

Debagovanje 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

```
> 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

Izuzeci u Python-u

Debugovanje naučnom
metodom

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 1.py`, prva linija za izvršavanje korišćenjem debagera biće `experiment = sum(new_list) / sub(diff,my_list_sum)`. U slučaju da dodamo i ove 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

Izuzeci u Python-u

Debugovanje naučnom
metodom

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

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

Izuzeci u Python-u

Debugovanje naučnom
metodom

Debugovanje print
metodom

PDB debager

Pokretanje iz komandne
linije

Pokretanje iz programa

PyCharm

Šta je PyCharm

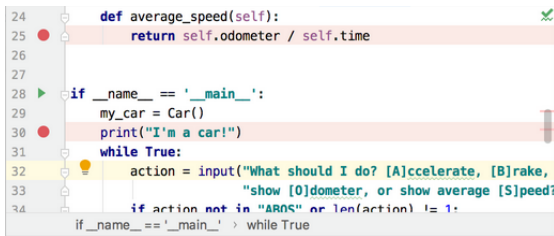
Tačke prekida i pokretanje
Debugger-a

Opcije Debugger-a

Literatura

Tačke prekida i pokretanje Debugger-a

U okruženju PyCharm tačke prekida postavljamo klikom na levu marginu (oznaka tačke prekida je crveni kružić).
Prilikom kompilacija možemo odabrati opciju Debug, nakon čega dobijamo zaseban prozor za Debugovanje (Debug Tool Window)



Slika: Postavljanje Tačaka prekida.

Opcije Debugger-a

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

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 promenljivim.

Posmatranja su zaseban prozor u kome se nalaze sve promenljive koje su trenutno definisane kao i njihove vrednosti, u Posmatranja mozemo ručno dodati bilo koju promenljivu (čak i one koje su trenutno ne definisane), njihova vrednost će biti **Null**.

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

Debagovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debagovanja u
Python-u

Uvod

Izuzeci u Pyhon-u

Debagovanje naučnom
metodom

Debagovanje 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

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



Kalyani Adawadkar (2017)

Python Programming-Applications and Future



Python 3.8.2 documentation

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



PyCharm

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

Debagovanje
Python

Dimitrije Sekulić,
Sandra Radojević,
Maja Gavrilović,
Matija Pejić

Osnovne tehnike
debagovanja u
Python-u

Uvod

Izuzeci u Python-u

Debagovanje naučnom
metodom

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