Načini debagovanja u programskom jeziku Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Matematički fakultet, Beograd

28. april 2020.

Uvod

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

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Uvod
Debagovanie s

grešaka

Debagovanje print

PDB debage

Pokretanje iz komandi linije

Pokretanje iz programa

PyCharn

ta je PyCharm

Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključak

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

Debagovanje sintaksnih grešaka

Naučni pristup debagovanju

DDD dabassas

Pokretanje iz komandni linije

Pokretanje iz programa

PyChar

a je PyCharm ačke prekida i pokretar

Opcije Debugger

Zakliučak

Debagovanje sintaksnih grešaka

- Kada u programu postoji sintaksna greška prevodilac izbacuje izuzetak i ispisuje poruku o grešci. Ona sadrži:
 - 1. Tip greške
 - 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

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Debagovanje sintaksnih grešaka

> aucni pristup debagova ebagovanje print ietodom

PDB debager

Pokretanje iz komandne linije

Pokretanje iz progran

PyCharr

Sta je PyCharm Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključa

Naučni pristup debagovanju

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

- Posmatraj
- 2. Napravi hipotezu
- 3. Predvidi
- 4. Testiraj
- Zaključi

Za efikasnu primenu, neophodno je vladati tehnikama:

- reprodukcije grešaka
- automatizacijom
- izolacijom grešaka

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Uvod

Debagovanje sintaks

grešaka

Naučni pristup debagovanju

Debagovanje print

PDB debagei

Pokretanje iz komandne linije

PvCharm

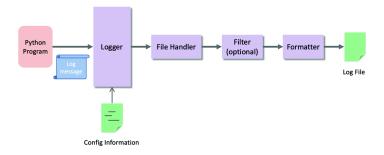
Šta je PyCharm Tačke prekida i pokretan

Opcije Debugger-

Zaključa

Debagovanje 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

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Pvthon-u

Debagovanje sintaksnih grešaka

Debagovanje print metodom

PDB debager

linije

Pokretanje iz prograr

PyChar

ia je PyCharm ačke prekida i pokre

Opcije Debugger

Zaključak

PDB debager

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

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Debagovanje sintaksnih grešaka

Debagovanje print netodom

PDB debager

linije

Pokretanje iz programa

PyChar

Sta je PyCharm Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključa

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

Debagovanje sintaksnih grešaka

Naučni pristup debagovanji Debagovanje print

PDB debager

linije

Pokretanje iz program

PyCha

ta je PyCharm

Tačke prekida i pokretanje Debugger-a

akliučak

```
> prvi.py(1) < module >()
\rightarrow 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.pv(3) < module > ()
-> def sub(a.b):
(Pdb) n
> prvi.pv(6) < module > ()
\rightarrow 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
```

Debagovanje Python

Dimitrije Sekulić. Sandra Radojević, Maia Gavrilović. Matiia Peiić

Osnovne tehnike debagovanja u

Pokretanje iz komandne

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).

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Pvthon-u

Debagovanje sintaksnih

Naučni pristup debagovanju Debagovanje print

PDB debager

Pokretanje iz komandne linije

Pokretanje iz programa

PyChar

Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključak

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 debagovanje
- 2. Posmatranja
- 3. Inline Debugger
- 4. Evaluacija izraza

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Debagovanje sintaksni

Naučni pristup debagovanju Debagovanje print

PDB debage

Pokretanje iz komandni linije

Pokretanje iz programa

PyChar

Šta je PyCharm

Tačke prekida i pokretanje Debugger-a

7-112-1-1

. .

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 debagovanje (Debug Tool Window)

```
def average_speed(self):
    return self.odometer / self.time

def average_speed(self):
    my_car

print("I'm a car!")
    while True:

action = input("What should I do? [A]ccelerate, [B]rake,
    "show [0]dometer, or show average [S]peed?

if action not in, "ABOS" or len(action) != 1:
    if_name_ == '_main_' > while True
```

Slika: Postavljanje tačaka prekida.

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Peijć

Osnovne tehnike debagovanja u Python-u

Debagovanje sintaksnih grešaka Naučni pristup debagovanji

PDB debager

linije
Pokretanje iz programa

PyCharn

Tačke prekida i pokretanje Debugger-a

Zaključa

Opcije Debugger-a

promenljivama.

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

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

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

Debagovanje sintaksnih grešaka

Naučni pristup debagovanju Debagovanje print

PDB debager

Pokretanje iz komandne linije

Pokretanje iz progran

PyCha

ta je PyCharm ľačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključal

Zaključak

Sve tehnike debagovanja koje smo vam danas prikazali su jednako moćne i ne treba vršiti njihovo poređenje, već kao iskusan programer treba biti upoznat sa njima i izabrati metodu koja najviše odgovara nama i onome što radimo. Nadamo se da vam se bar nešto od ovoga dopalo i da ćete biti podstaknuti da se oprobate u nečemu što nije print metoda za debagovanje.

Hvala na pažnji!

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Uvod Debagovan

grešaka Naučni pristup debagovanj

Debagovanje print metodom

PDB debagei

Pokretanje iz komandne linije

Pokretanje iz programa

PyChar

ta je PyCharm

Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a

Zaključak

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/

Debagovanje Python

Dimitrije Sekulić, Sandra Radojević, Maja Gavrilović, Matija Pejić

Osnovne tehnike debagovanja u Python-u

Debagovanje sintaksnih grešaka

Naučni pristup debagovanji. Debagovanje print

PDB debage

linije Pokretanje iz programa

Pokretanje iz progra

²yCharm

Tačke prekida i pokretanje Debugger-a

Opcije Debugger-a