







Základy jazyka Python

Testovanie, ladenie, výnimky a chyby prednáška 5

Katedra kybernetiky a umelej inteligencie Technická univerzita v Košiciach Ing. Ján Magyar

Testovanie programov

- proces, ktorého cieľom je nájsť problémy a chyby v kóde
- zatial' nehl'adáme príčinu chýb, ani možné riešenia
- testovanie sa delí na
 - o verifikáciu
 - kontroluje, či program, alebo jeho časť, splní návrhové požiadavky (požiadavky zo strany programátora)
 - nezaoberá sa iba s tým, čo daný program robí, ale či to robí tak, ako chceme
 - validáciu
 - kontroluje, či program, alebo jeho časť, splní požiadavky používateľa čo program robí
 - cieľom je zvýšiť spoľahlivosť kódu nikdy nedosiahne 100%

Typy testovania

- unit testy
 - o nezávislé testovanie prvkov kódu
 - o testujú sa funkcie, samostatne
- integration testy
 - testuje sa program ako celok
 - o slúži na odhalenie problémov v komunikácii medzi modulmi a prvkami kódu
- vždy je najlepšie začať s unit testami

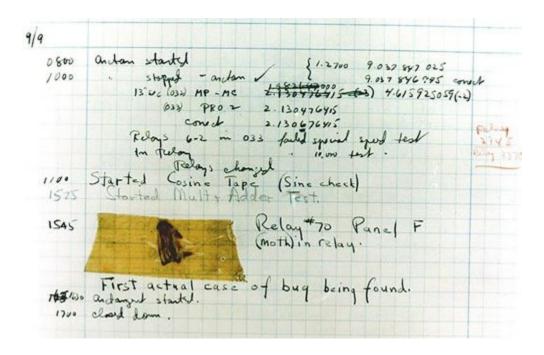
Návrh testov

- testovať program na každom možnom vstupe nepraktické a niekedy nemožné
- testovanie na vhodne zvolenej sade testovacích prípadov
 - sada musí byť dostatočne malá, aby sme test vedeli vykonať v rozumnom časovom intervale
 - o sada musí byť dostatočne veľká, aby reprezentovala všetky možné vstupy
- testovacia sada zvyčajne obsahuje
 - príklady najčastejších vstupov (očakávaný vstup)
 - príklady hraničných vstupov (extrémy)
 - príklady neplatných vstupov (asi najdôležitejšie)

Ladenie

- našli sme chybu v kóde, potrebujeme ju odstrániť
- proces, počas ktorého našim cieľom je odstrániť všetky bugy z kódu
- chyba sa nemusí preukázať nefunkčnosťou programu
- niekedy je počas ladenia naším cieľom optimalizovať beh programu, resp. zvýšiť výkon

Prečo sa bug volá bug?



Ďalšie mýty o bugoch

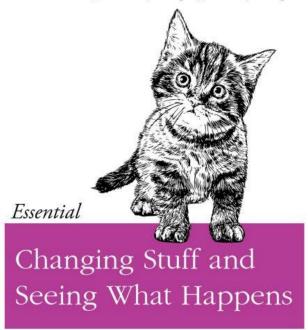
- 1. bug sa zjaví v kóde
 - o bug je v kóde preto lebo programátor ho tam (nechtiac) pridal
 - o bug nie je nič viac ako omyl programátora
- 2. bugy sa rozmnožujú ak človek opraví jeden bug, nahradia ho dva
 - ak zrazu musíte vyriešiť viac bugov, znamená to že ste urobili viac chýb ako ste si mysleli
 - cieľom nikdy nie je opraviť jeden bug, ale z nefunkčného kódu vytvoriť funkčný kód

Nástroje debugovania

- existuje mnoho nástrojov pre rôzne IDE
- najlepší spôsob debugovania je
 - prečítať si kód
 - operácia print

Postup pri ladení kódu

How to actually learn any new programming concept



O RLY?

@ThePracticalDev

Postup pri ladení kódu

- 1. Kde sa nachádza chyba?
- 2. Ako môže kód vyprodukovať chybný výstup?
- 3. Je to ojedinelý problém alebo urobili sme tú istú chybu viackrát v programe?
- 4. Ako opravíme chybu?

- vždy testujte na najjednoduchšom možnom vstupe
- pri identifikácii miesta chyby znižujte priestor vyhľadávania pomocou binárneho vyhľadávania

Príklad

```
def isPalindrome():
   originalList = list()
   done = False
   while not done:
       elem = input("Enter element. Return when done. ")
       if elem == '':
           done = True
       else:
           originalList.append(elem)
   testList = originalList
   testList.reverse()
   return testList == originalList
```

Typické chyby

- nesprávne poradie parametrov/argumentov
- pravopis, preklepy, veľké/malé písmená
- inicializácia chceme to robiť v cykle alebo mimo?
- rovnosť objektov vs. rovnosť hodnôt
- aliasy vs. kópie, deep vs. shallow kópie
- vedľajšie účinky volaných funkcií

+ každý programátor má svoje často sa opakujúce chyby

Dobré zvyky pri ladení

- systematická kontrola
- zapisovať čo sme už vyskúšali
- skontrolovať, či máme správne predpoklady
- ladiť kód a nie komentáre
- poprosit' o pomoc
- vysvetliť niekomu, čo robí program, resp. čo by mal robiť
- dať si pauzu
- zjednodušiť kód
- uložiť staré verzie kódu

Výnimky a chyby

- poukazujú na nesprávnosť programu
- dva základné typy
 - syntaktické (chyby)
 - o run-time (výnimky)

Syntaktické chyby

- interpreter ich identifikuje ešte pred spustením kódu
- riešia sa veľmi jednoducho

SyntaxError: invalid syntax

• chyba sa nie vždy nachádza tam, kde to ukazuje interpreter

Run-time chyby/výnimky

• vznikajú počas behu programu

number = 10

zvyčajne ich spôsobujú iba vybrané vstupy

```
divideBy = 0
print(number / divideBy)

File "C:\Users\Ian\Desktop\test.py", line 3, in <module>
    print(number / divideBy)
ZeroDivisionError: division by zero
```

Spracovanie chýb a výnimiek

• dvojica kľúčových slov try / except

```
number = 10
divideBy = 0
try:
    print(number / divideBy)
except ZeroDivisionError:
    print("You're trying to divide by zero")
```

Spracovanie viacerých chýb

```
number = 10
divideBy = 0
try:
    print(number / divideBy)
except (ZeroDivisionError, NameError, ValueError):
    print("You're trying to divide by zero")
```

Spracovanie viacerých chýb

```
number = 10
divideBy = 0
try:
    print(number / divideBy)
except ZeroDivisionError:
    print("You're trying to divide by zero")
except NameError:
    print("Undefined variable")
except ValueError:
    print("Invalid value")
```

Vyhodenie chýb

- kľúčové slovo raise
- umožňuje zadať ľubovoľnú chybovú správu

```
number = 10
divideBy = 0
if divideBy == 0:
    raise ZeroDivisionError("You're trying to divide by zero")
print(number / divideBy)
```

Vykonanie kódu iba v prípade bez chýb

```
number = 10
divideBy = 0
try:
    print(number / divideBy)
except ZeroDivisionError:
    print("You're trying to divide by zero")
else:
    print("Everything went well")
```

Finally časť

kód sa vykoná bez ohľadu na to, či vznikla chyba

```
try:
    raise ValueError
finally:
   print("I still get printed")
I still get printed
Traceback (most recent call last):
  File "C:\Users\Ian\Desktop\test.py", line 2, in <module>
   raise ValueError
ValueError
```

Definícia vlastných chýb

• výnimky a chyby sú podtriedy Exception

```
class MyError(Exception):
    def __init__(self, expression, message):
        self.expression = expression
        self.message = message
```

Práca s výnimkami

```
myDict = dict()
if 'a' not in myDict:
    myDict['a'] = 1
else:
    myDict['a'] += 1

except KeyError:
    myDict['a'] = 1

print(myDict)
```

Najčastejšie typy výnimiek

- AssertionError
- AttributeError
- IndexError
- KeyError
- NameError
- OSError
- RecursionError
- TypeError
- UnicodeError
- ValueError
- ZeroDivisionError

Otázky?