Plagiarism Scan Report

Summary	
Report Genrated Date	27 Aug, 2017
Plagiarism Status	**100% Unique**
Total Words	594
Total Characters	4039
Any Ignore Url Used	

Content Checked For Plagiarism:

Kelas turunan dinyatakan seperti kelas orang tua mereka; Namun, da□tar kelas dasar yang diwarisi dari diberikan setelah nama kelas -

Kelas SubClassName (ParentClass1 [, ParentClass2, ...]):
'String dokumentasi kelas opsional'
Class_suite
#!/usr/bin/python

class Parent: # de□ine parent class
parentAttr = 100
de□ __init__(sel□):
print "Calling parent constructor"

de□ parentMethod(sel□):
print 'Calling parent method'

de□ setAttr(sel□, attr):
Parent.parentAttr = attr

de□ getAttr(sel□):
print "Parent attribute :", Parent.parentAttr

class Child(Parent): # de□ine child class
de□ __init__(sel□):
print "Calling child constructor"

de□ childMethod(sel□):
print 'Calling child method'

c = Child() # instance o□ child
c.childMethod() # child calls its method
c.parentMethod() # calls parent's method
c.setAttr(200) # again call parent's method
c.getAttr() # again call parent's method

Calling child constructor

Calling child method
Calling parent method
Parent attribute : 200

class A: # de□ine your class A
.....

class B: # de□ine your class B
.....

class C(A, B): # subclass o□ A and B
Anda dapat menggunakan □ungsi issubclass () atau isinstance () untuk memeriksa hubungan dua kelas dan contoh.

Fungsi boolean issubclass (sub, sup) mengembalikan true jika sub subclass yang diberikan memang merupakan subclass dari superclass sup.

The isinstance (obj, Class) □ungsi boolean mengembalikan true jika obj adalah turunan dari Class Class atau merupakan instance dari subclass o□ Class.

Metode utama

Anda selalu dapat mengganti metode kelas induk Anda. Salah satu alasan untuk mengesampingkan metode orang tua adalah karena Anda mungkin menginginkan □ungsi khusus atau berbeda di subkelas Anda.
Contoh

#! / Usr / bin / python

class parent: # de□ine parent class
De□ myMethod (diri):
Cetak 'metode induk panggilan'

Kelas anak (orang tua): # de□ine child class
De□ myMethod (diri):
Cetak 'metode memanggil anak'

C = Anak () # contoh anak
C.myMethod () # metode panggilan balik anak

Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut -

Memanggil metode anak

Metode Base Overloading

Berikut da□tar tabel beberapa □ungsionalitas generik yang dapat Anda timpa di kelas Anda sendiri -

Operator overloading

Misalkan Anda telah membuat kelas Vektor untuk mewakili vektor dua dimensi, apa yang terjadi bila Anda menggunakan operator plus untuk menambahkannya? Kemungkinan besar Python akan berteriak pada Anda.

Anda bisa, bagaimanapun, menentukan metode __add__ di kelas Anda untuk melakukan penambahan vektor dan operator plus akan berperilaku sesuai harapan -
Contoh

```
#! / Usr / bin / python

Kelas vektor:
de□ __init __ (diri, a, b):
Sel□.a = a
Sel□.b = b

de□ __str __ (diri):
Return 'Vector (% d,% d)'% (sel□.a, sel□.b)

De□ __add __ (diri sendiri, lainnya):
return Vector (sel□.a + other.a, sel□.b + other.b)

v1 = vektor (2,10)
v2 = vektor (5, -2)
cetak v1 + v2
```

Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut -

```
Vektor (7,8)
```

Persembunyian data

Atribut objek mungkin atau mungkin tidak terlihat di luar de□inisi kelas. Anda perlu memberi nama atribut dengan awalan ganda ganda, dan atribut tersebut kemudian tidak langsung terlihat oleh orang luar.
Contoh

```
#! / Usr / bin / python

Kelas JustCounter:
__secretCount = 0

de□ menghitung (diri):
sel□ .__ secretCount + = 1
cetak diri .__ secretCount

counter = JustCounter ()
Counter.count ()
Counter.count ()
print counter .__ secretCount
```

Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut -

1
2
Traceback (panggilan terakhir):
File "test.py", baris 12, di
print counter .__ secretCount
AttributeError: instance JustCounter tidak memiliki atribut '__secretCount'

Python melindungi anggota tersebut dengan mengganti namanya secara internal untuk memasukkan nama kelas. Anda dapat mengakses atribut seperti object._className__attrName. Jika Anda akan mengganti baris terakhir Anda sebagai berikut, maka akan berhasil untuk Anda -

print counter._JustCounter__secretCount
Bila kode diatas dieksekusi, maka menghasilkan hasil sebagai berikut -

1
2
2