Audit Access Expires 5 de ago de 2020
You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise 4

Finger Exercises due Aug 5, 2020 20:30 -03

Exercise 4

7/7 points (graded)

ESTIMATED TIME TO COMPLETE: 12 minutes

Python supports a limited form of multiple inheritance, demonstrated in the following code:

```
class A(object):
    def __init__(self):
        self.a = 1
    def x(self):
        print("A.x")
    def y(self):
        print("A.y")
    def z(self):
        print("A.z")

class B(A):
    def __init__(self):
        A.__init__(self)
        self.a = 2
        self.b = 3
    def y(self):
        print("B.y")
    def z(self):
        print("B.z")

class C(object):
    def __init__(self):
        self.a = 4
        self.c = 5
    def y(self):
        print("C.y")
    def z(self):
        print("C.z")

class D(C, B):
    def __init__(self):
        C.__init__(self)
        B.__init__(self)
        self.d = 6
    def z(self):
        print("D.z")
```

Which `__init__` methods are invoked and in which order is determined by the coding of the individual `__init__` methods.

When resolving a reference to an attribute of an object that's an instance of class `D`, Python first searches the object's instance variables then uses a simple left-to-right, depth first search through the class hierarchy. In this case that would mean searching the class `C`, followed the class `B` and its superclasses (ie, class `A`, and then any superclasses it may have, et cetera).

With the definitions above if we define

```
obj = D()
```

then what is printed by each of the following statements?

1. `print(obj.a)`

2 ✔ **Answer:** 2

2. `print(obj.b)`

3 ✔ **Answer:** 3

3. `print(obj.c)`

5 ✔ **Answer:** 5

4. `print(obj.d)`

6 ✔ **Answer:** 6

5. `obj.x()`

A.x ✔ **Answer:** A.x

6. `obj.y()`

C.y ✔ **Answer:** C.y

7. `obj.z()`

D.z ✔ **Answer:** D.z

Enviar

Answers are displayed within the problem

Exercise 4

Topic: Lecture 9 / Exercise 4

Ocultar discussão

Add a Post

All Posts

obj.a

question posted a day ago by **EthVedBitDesJan**

Why does print(obj.a) return 2 and not 4 as defined in class C?

This post is visible to everyone.

Add a Response

1 response

kiwitrader (Community TA)
a day ago

Short answer: look at the `__inits__` vs inheritance. With values in inits, it runs the inits in sequence and the last one run determines the value (functions just overwrite each others' values). And whenever an attribute is defined by initialisation *it doesn't need to inherit it*.

Experiential answer: run it in python tutor and watch how that works.

Longer answer:

The first thing is that there are two things happening. Some attributes are defined by `__init__` and some by `inheritance`.

If something is defined by `__init__` then when you run the init functions, if two of them define the same thing, then the last one overwrites the first one and you're left with the last version initialized. The last one wins.

Inheritance works the ***opposite*** way - you go up the tree, searching left & up first but as soon as you find the attribute/method you **stop** so there is no overwriting. The first one wins.

So, to a frequently asked question as an example:

1. obj.a was defined in C.init, then overwritten by B.init so it was 4 for a moment then it was 2. When you ask for it, **there is no inheritance** NONE .... why? because you already had that value from running the two inits as part of D's init.

2. obj.y was NOT defined as part of the init process so unlike a, it will have to be found by inheritance. So, left to right depth first. Look at C. Found it. Stop.

So your first one was there because the object was initialized with it. The second one wasn't generated by the initialization process so the interpreter had to use inheritance.

Make sense?

Tnx for the great explanation kiwitrader! BTW, I hope you were long kiwi cuz the price seems to be booming this month! https://www.tridge.com/intelligences/kiwifruit

posted about 23 hours ago by **inovakovic**

Great explanation, thank you.

posted about 21 hours ago by **Ugnetay**

Thank you very much ! Very helpful advice !

posted about 2 hours ago by **leopold_bernard_leo**

I think the explanation in the statement ended up misleading us.

posted less than a minute ago by **leite_am**

Add a comment

Exibindo todas as respostas

Add a response:

Pré-visualizar

Enviar