

PROGRAMACIÓN ORIENTADA A OBJETOS

TALLER No. 8 PYTHON

PROGRAMACIÓN ORIENTADA A OBJETOS

Profesor: Jaime Alberto Guzmán Luna

Contenido del taller:

1. Herencia múltiple

Ejercicio 1

- a) ¿Qué imprime el siguiente código y por qué? (Suponga que se crea una instancia de Cuarto y se llama al método m)

```
1 class Primero:
2     def m(self):
3         print("m desde primero")
4
5 class Segundo(Primero):
6     def m(self):
7         print("m desde segundo")
8         super().m()
9
10 class Tercero(Primero):
11     def m(self):
12         print("m desde tercero")
13         super().m()
14
15 class Cuarto(Segundo, Tercero):
16     def m(self):
17         print("m desde cuarto")
18         super().m()
19
20
21
```

PROGRAMACIÓN ORIENTADA A OBJETOS

Ejercicio 2

- a) ¿Qué imprime el siguiente código y por que (Suponga que se crea una instancia de Fourth)?

```
1 class First(object):
2     def __init__(self):
3         print("First")
4         super().__init__()
5
6
7
8 class Second(First):
9     def __init__(self):
10        print("Second")
11        super().__init__()
12
13 class Third(First):
14     def __init__(self):
15        print("Third")
16        super().__init__()
17
18 class Fourth(Second, Third):
19     def __init__(self):
20        print("Fourth")
21        Second.__init__(self)
22        Third.__init__(self)
23
24
25 a = Fourth()
26
```

- b) Que imprime y por que si hago el siguiente cambio que consta de pasar la linea 22 arriba de la 21?
- Third.__init__(self)
 - Second.__init__(self)

Ejercicio 3

```
1 class Numeros:
2     def __init__(self):
3         self.a = 4
4         self.b = 5
5
6 class Operadores1:
7     def sumar(self):
8         return self.a + self.b
9     def multiplicar(self):
10        return self.a * self.b
11
12 class Operadores2:
13     def dividir(self):
14        return self.a / self.b
```

PROGRAMACIÓN ORIENTADA A OBJETOS

15	def restar(self):
16	return self.a - self.b
17	
18	class Operadores3(Operadores1, Operadores2):
19	pass
20	
21	class Ejecucion1(Numeros):
22	pass
23	
24	class Ejecucion2(Operadores1, Operadores2):
25	pass
26	
27	class Ejecucion3(Operadores1, Numeros):
28	pass
29	
30	class Ejecucion4(Operadores3, Numeros):
31	pass
32	
33	if __name__ == "__main__":
34	ob1 = Ejecucion1()
35	ob2 = Ejecucion2()
36	ob3 = Ejecucion3()
37	ob4 = Ejecucion4()
38	
39	ob1.sumar()
40	ob2.sumar()
41	ob3.sumar()
42	ob4.sumar()
43	
44	ob1.restar()
45	ob2.restar()
46	ob3.restar()
47	ob4.restar()
48	
49	ob1.multiplicar()
50	ob2.multiplicar()
51	ob3.multiplicar()
52	ob4.multiplicar()
53	
54	ob1.dividir()
55	ob2.dividir()
56	ob3.dividir()
57	ob4.dividir()

En base al anterior código responda...

1. Indique por qué funciona o no el método sumar en las líneas 39 a 42
2. Indique por qué funciona o no el método restar en las líneas 44 a 47
3. Indique por qué funciona o no el método multiplicar en las líneas 49 a 52
4. Indique por qué funciona o no el método dividir en las líneas 54 a 57
5. ¿Qué clase puede existir sin necesidad que se herede en otra clase?

PROGRAMACIÓN ORIENTADA A OBJETOS

Ejercicio – GitHub

Link entre ga:<https://classroom.github.com/a/HVd8lr0s>.

PROBLEMA Implementación de Futbolista

En este problema se implementará la clase Futbolista la cual heredará de la clase Persona Y Deportista.

- a) Crear la clase Persona que tiene las siguientes características:
 - Tendra los siguientes atributos privados: nombre(String), edad(Int), altura(String) y sexo(String)
 - Debe tener el constructor que inicialice los atributos anteriores(dándoles un valor)
 - Crear los métodos get y set para obtener y dar valor a los atributos de la clase
- b) Crear la clase Deportista que tiene las siguientes características:
 - Tendra los siguientes atributos privados: deporte(String) y añosPracticando(Int)
 - Debe tener el constructor que inicialice los atributos anteriores(dándoles un valor)
 - Crear los métodos get y set para obtener y dar valor a los atributos de la clase
- c) Crear la clase Futbolista que tiene las siguientes características:
 - Tendra los siguientes atributos privados: golesMarcados(Int), tarjetasRojas(Int) , piernaHabil(String) y además un atributo de clase llamado listaFutbolistas.
 - Debe tener un constructor que reciba todos los parámetros para inicializar tanto la clase futbolista, como la clase Persona y también la clase deportista y además agregar a la lista de futbolistas el futbolista creado
 - Deben utilizar los constructores de su padre modificando tanto esta clase como las previamente creadas para cuando se haga una instancia de futbolista este cree las instancias de Persona y Deportista(Tener en cuenta lo visto en la clase y la primera parte de este taller para saber como se comporta la herencia en estos casos)
 - Crear los métodos get y set para obtener y dar valor a los atributos de la clase
 - Debe tener un método __str__(Este método se redefine para cuando se imprima un objeto no imprima un espacio de memoria si no que imprima lo que se redefinió en el metodo)que cuando sea llamado deberá devolver
“Mi nombre es #Persona.nombre soy profesional en el deporte
#Deportista.deporte Tengo #Persona.edad años de edad y llevo
#Deportista.añosParticipando años en el deporte”

Nota: Cuando se desee instanciar la clase deportista se le mandara que el deporte por defecto es “Futbol”.