INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA

Investigación

Muy buena investigaci· n y ejemplo.

25%

Constructores y Destructores en POO

DIVISIÓN DE INGENIERÍA MECATRÓNICA

Autor:

Esmeralda Serrano Velázquez

MTI. Roberto Esteban Guerrero Hernández

Docente

Los destructores se llaman cuando un objeto se destruye. Es el polo opuesto del constructor, que se invoca en la creación.

Estos métodos solo se invocan en la creación y destrucción del objeto. No se llaman de forma manual sino completamente automática.

Un destructor puede hacer lo contrario de un constructor, pero eso no es necesariamente cierto. Puede hacer algo diferente. Un destructor es una función que se llama cuando se elimina o destruye un objeto.

Antes de que se destruya el objeto, puedes hacer algunas tareas finales. Imagina conducir un Tesla y en el código el objeto del motor se destruye. No, primero querrá apagar el motor, asegurarse de que las ruedas no estén girando, etc.

Tenga en cuenta que destruir un objeto con del es opcional, puede crear objetos y nunca eliminarlos. Luego solo se eliminan cuando se cierra el programa. Sin embargo, esto puede consumir mucha memoria en programas grandes.

Siempre es parte de una clase, incluso si no está definida. (Si no está definido, Python asume un destructor vacío).

Ejemplo Destructor en python

La siguiente clase tiene un constructor (**en eso**) y destructor (**del**). Creamos una instancia de la clase y la eliminamos inmediatamente después.

Un ejemplo del uso de destructor se muestra en el siguiente código:

```
class Vehicle:

    def __init__(self):

        print('Vehicle created.')



    def __del__(self):

        print('Destructor called, vehicle deleted.')
```

```
car = Vehicle()

del car
```

Si ejecuta el programa anterior, puede ver este resultado en la terminal:

```
Vehicle created.

Destructor called, vehicle deleted.
```

La salida se muestra, aunque no llamamos a ningún método. Eso es porque un constructor y un destructor son llamados automáticamente.

LISTA DE COTEJO PARA REPORTE DE EJERCICIO

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA	**ASIGNATURA:** PROGRAMACION AVANZADA
NOMBRE DEL DOCENTE: MTI. ROBERTO E. GUERRERO HERNANDEZ	**FIRMA DEL DOCENTE**

DATOS GENERALES DEL PROCESO DE EVALUACIÓN

NOMBRE(S) DEL ALUMNO(S): ESMERALDA SERRANO VELAZQUEZ	**MATRICULA:** 221U0561	**FIRMA DEL ALUMNO(S):**

PRODUCTO: REPORTE DE EJERCICIO	**NOMBRE DE LA PRACTICA:** CLASES PUBLICAS Y PRIVADAS	**FECHA:** OCTUBRE - 2023	**PERIODO ESCOLAR:** SEPTIEMBRE 23 – ENERO 24

INSTRUCCIONES

Revisar las actividades que se solicitan y marque con una "X" en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.

VALOR DEL REACTIVO	CARACTERÍSTICA PARA CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
3%	Presentación El trabajo cumple con los requisitos de: **a**. Buena presentación	X		
3%	**b**. No tiene faltas de ortografía	X		
3%	**c**. Mismo Formato (letra arial 12, títulos con negritas)	X		
3%	**d**. Maneja el lenguaje técnico apropiado	X		
15%	**Desarrollo**: Sigue una metodología y sustenta todos los pasos que se realizaron en el análisis y desarrollo en la aplicación de los temporizadores dentro de la programación del PLC y aplicando los conocimientos obtenidos, es analítico y bien ordenado.	X		
3%	**Responsabilidad**: Entregó la práctica en la fecha y hora señalada.	X		
30%	**CALIFICACIÓN**	30 %		

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA

Práctica

Clases base Públicas y Privadas

DIVISIÓN DE INGENIERÍA MECATRÓNICA

Autores:

Esmeralda Serrano Velázquez

Realiza código en Python en donde se demuestre la “declaración de atributos públicos y privados”

```
var Username // Public
var password // Private

struct User { // Public
   Username string
      Age int
      password string // Private
}
```

```
public String username = 'Cody';
private String password = 'Password123';
```

```
class User:
   def __init__(self, username):
      self.username = username

cody = User('Cody')
print(cody.username)

cody.username = 'Cambio de nombre'
print(cody.username)
class User:

   def __init__(self, username):
```

```
        self.username = username

    def set_username(self, username):
        self.username = username

    def get_username(self):
        return self.username

cody = User('Cody')
print(cody.get_username())

cody.set_username('Cambio de nombre')
print(cody.get_username())
```

INSITUTO TECNOLOGICO SUPERIOR DE SAN ANDRES TUXTLA
ING. MECATRONICA
PROGRAMACION AVANZADA
EXAMEN UNIDAD III

ITSSAT

GRUPO: 311-B **ALUMNO:** SERRANO VELAZQUEZ ESMERALDA

1.- Realiza código en Python en donde muestres la referencia a distintos ámbitos y espacios de nombres y las declaraciones globales.

40 %

```
def scope test()
    def do local():
        spam = "local spam"
    def do nonlocal():
        non local spam
        spam = non local spam
    def do global():
        global spam
        spam = "global spam"
        Spam =

spam = test spam
do local()
print
do nonlocal()
print("After local asignation"); spam
do global()
scope test()
print("global:" spam).
```