INSTITUTO SUPERIOR POLITÉCNICO CÓRDOBA

Clases estáticas e Interfaces

Contenido

Clases estáticas

En Python, una clase estática es una clase que no se puede heredar y no se puede instanciar directamente. Se puede implementar una clase estática haciendo que sus métodos y variables sean estáticos.

Para definir una clase estática debemos utilizar el decorador @classmethod como sigue:

```
class StaticClass:
    @classmethod
    def static_method(cls):
        print("Este es un método estático de la clase.")

StaticClass.static_method() # Imprime "Este es un método estático de la clase."
```

Como podemos observar arriba, se define una clase llamada StaticClass con un método llamado static_method, que se define como un método estático utilizando el decorador @classmethod. El método estático se puede llamar sin instanciar la clase como puedes observar en la última línea del ejemplo.

Interfaces

Una interface es un contrato que toda clase que la implemente debe cumplir. En Python, aunque no existe una keyword interface como en otros lenguajes de programación, es posible definir interfaces de dos formas: informales y formales.

Las interfaces formales se pueden definir en Python utilizando el módulo por defecto llamado ABC (Abstract Base Classes). Los ABC fueron añadidos a Python en la PEP3119 y permiten definir una forma de crear interfaces a través de metaclases. En estas interfaces se definen métodos, pero no se implementan, y se fuerza a las clases que usan esa interfaz a implementar los métodos.

A continuación un ejemplo de cómo definir una clase abstracta utilizando el módulo ABC:

```
from abc import ABC, abstractmethod


class Mando(ABC):
    @abstractmethod
    def encender(self):
        pass
```

```
    @abstractmethod
    def apagar(self):
        pass

class Television(Mando):
    def encender(self):
        print("La televisión está encendida.")

    def apagar(self):
        print("La televisión está apagada.")

tv = Television()
tv.encender() # Imprime "La televisión está encendida."
tv.apagar() # Imprime "La televisión está apagada."
```

En este ejemplo, se define una clase abstracta Mando que hereda de ABC y define dos métodos abstractos: encender y apagar. Luego, se define una subclase Televisión que hereda de Mando y sobrescribe los métodos abstractos para implementar el comportamiento específico de la televisión. Se puede instanciar la subclase Televisión y llamar a los métodos encender y apagar en la instancia, lo que hace que la televisión se comporte de manera específica en función de su relación con la clase abstracta Mando. Este es un ejemplo de interfaz formal en Python utilizando ABC.

Referencias

Apuntes Programa Clip - Felipe Steffolani

https://www.freecodecamp.org/

Laura Álvarez, Helmer Avendaño, Yeison García, Sebastián Morales,
Edwin Bohórquez, Santiago Hernandez, Sebastián Moreno, Cristian Orjuela.
Programación Orientada a Objetos.
https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoria/concepts.html

https://www.typescriptlang.org/

https://barcelonageeks.com/composicion-de-funciones-en-python/

111Mil. Módulo Programación Orientada a Objetos.
https://github.com/111milprogramadores/apuntes/blob/master/Programacion%20Orientada%20a%20Objetos/Apunte%20Teorico%20de%20Programacion%20OO.pdf