Programación orientada a objetos

CI-0202 Principios de informática

Sivana Hamer - sivana.hamer@ucr.ac.cr
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica

UNIVERSIDAD DE COSTA RICA

¿Qué es la programación orientada a objetos?

- Es un **paradigma de programación** donde el foco principal son los **objetos** que contiene tanto **datos como funcionalidad**.
- A comparación del **paradigma funcional** donde el foco principal son las **funciones** que **operan sobre datos**.

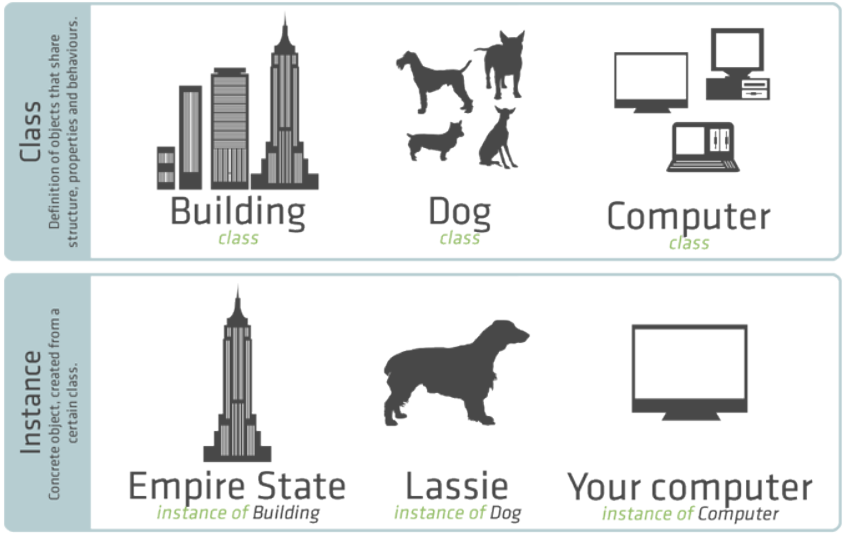
Fun fact!

En Python, los tipos de datos son objetos. Por lo tanto, hemos utilizado este paradigma anteriormente.

Instancia y clases (1)

- Una **clase** es la **plantilla o definición** de los objetos.
  - **Atributos** son las **caracteristicas o propiedades** de una instancia, representados en **variables**. Un ejemplo es el nombre de una persona.
  - **Métodos** son las **acciones** de una instancia, representados en **funciones**. Un ejemplo es comunicar.
- Una **instancia** es el **objeto concreto**. Se puede también llamar objeto.

Instancia y clases (2)



Como usar clases y objetos en Python

Ejemplo en Python

```
#Definicion de una clase
class Persona:
    #Constructor
    def __init__(self):
        #Atributo
        self.nombre = ""

    #Metodo
    def decir_nombre(self):
        print(f"Hola, mi nombre es {self.nombre}")

#Instancias
p1 = Persona()
p2 = Persona()
```

Atributos (1)

Se pueden acceder a las **variables** y aplicar operaciones (e.g. cambiarlo o leerlo).
Se accede al atributo de una clase con el **punto (.)**.

Ejemplo en Python

```
>> p1.nombre = "Pedro"
>> print(p1.nombre)
Pedro
```

Atributos (2)

Cada **instancia es independiente de otras**, por lo que sus atributos también son independientes.

Ejemplo en Python

```
>> p2.nombre = "Ana"
>> print(p1.nombre)
Pedro
>> print(p2.nombre)
Ana
```

Métodos

Se puede igualmente **invocar** como una **función** con el **punto (.)**

Ejemplo en Python

```
>> p1.decir_nombre()
Hola, mi nombre es Pedro
>> p2.decir_nombre()
Hola, mi nombre es Ana
```

Métodos - Self

- Podemos ver que aunque el método tiene un parámetro llamado **self**, no se está pasando.
- **self** es un parámetro que contiene la **referencia** de la **instancia** creada.
- Este parámetro es **obligatorio** para todos los **métodos** y debe ir de **primero**.
- Se pueden agregar otros parámetros en el método.

Constructor (1)

- Es un método especial que se **invoca** siempre cuando se **crea** el objeto. Este método **retorna** el objeto.
- Se le pueden **dar atributos** al constructor para la instancia a crear.
- En ptyhon, se llama con el método **__init__**.

Constructor (2)

Ejemplo en Python

```
#Definicion de una clase
class Persona:
    #Constructor
    def __init__(self, nombre, apellido):
        #Atributo
        self.nombre = nombre
        self.apellido = apellido

    #Metodo
    def decir_nombre_completo(self):
        print(f"Hola soy {self.nombre} {self.apellido}")

#Instancias
p1 = Persona("Ezio", "Auditore")
p2 = Persona("Connor", "Kenway")
p1.decir_nombre_completo()
p2.decir_nombre_completo()
```

Recomendaciones

- Ayuda en manejar el tamaño y complejidad de sistemas de software.
- La programación orientada a objetos es un tema sumamente complejo y extenso, por lo cual vamos a aplicar conceptos básicos de objetos.
- Se recomienda que mantenga los objetos simples y que se encarguen de funciones específicas.
- Además, se recomienda que la interacción entre los objetos sea la encargada de resolver el problema.

Referencias I

L. Villalobos, “Programación orientada a objetos,” Material del curso CI-0202 Universidad de Costa Rica de Leonardo Villalobos, 2019.

C. Swaroop, *A Byte of Python*. Independent, 2020.

J. Elkner, A. B. Downey, and C. Meyers, “How to think like a computer scientist: Learning with python,” 2012.

[Image]. [Online]. Available: https://docs.sencha.com/extjs/6.2.0/guides/other_resources/images/classes_instances.png

[Image]. [Online]. Available: https://stackoverflow.com/questions/2885385/what-is-the-difference-between-an-instance-and-an-object