Programación orientada a objetos

Introducción a la Programación

Agenda

- Definición
- Historia
- Ventajas de P.O.O
- Conceptos básicos
- Objetos en Python

Programación Orientada a Objetos

- En la Programación Orientada a Objetos (POO), el elemento fundamental es el objeto
- Un objeto es una extensión de un Tipo Abstracto de Datos (o ADT, por sus siglas en inglés)
- Un ADT es un tipo definido por el usuario, que encapsula un conjunto de datos o información y operaciones que se realizan sobre esos datos

Programación Orientada a Objetos

- Cuando se definen ADTs se usa el concepto de abstracción, que ayuda a representar la realidad mediante un modelo a nivel informático

- La abstracción es un proceso por el que se evitan detalles de implementación para centrarse en aspectos más importantes que faciliten la comprensión del problema

    - Ejemplo: arquitecto al diseñar un edificio

- La diferencia entre los ADTs y los objetos es que el segundo usa además otros conceptos fundamentales en la programación orientada a objetos: herencia y polimorfismo

Historia

- Los primeros conceptos de Orientación a Objetos aparecieron por primera vez en el MIT, en la década de 1950
- El lenguaje Lisp tuvo mucha influencia en el desarrollo de O.O.
- En 1960 se empezaron a usar los conceptos de objetos, clases y en ciertos lenguajes, como por ejemplo Simula
- En 1970 Alan Kay y otros colaboradores desarrollaron Smalltalk, y se empezó a usar formalmente el término Programación Orientada a Objetos

Historia

- Las ideas de O.O. en Smalltalk fueron usadas en otros lenguajes, como en Lisp con su Common Lisp Object System

- A partir de la década de 1990 se convirtió en el paradigma dominante, con la popularización de lenguajes como Delphi, C++, Java, y un poco mas adelante todo el framework .Net y lenguajes más modernos como Ruby y Python

Ventajas de la POO

- Mantenibilidad: usar POO ayuda a comprender mejor los programas, y se reduce la complejidad de los mismos mediante la encapsulación de información, siempre que se mantengan visibles solo los detalles relevantes

- Reusabilidad: aplicar los principios de POO permite crear objetos que pueden ser usados múltiples veces (asumiendo que hayan sido correctamente diseñados), como por ejemplo las librerías de clases

- Modificabilidad: Se pueden cambiar elementos del programa agregando, modificando o eliminando objetos

- Fiabilidad: Reutilizar los objetos en los programas permite asegurarse de que se está usando código que ha sido probado con anterioridad, lo que perimte aumentar la fiabilidad de los programas

Ventajas de la POO

- La POO facilita tanto las labores de programación, como las etapas previas de análisis y diseño

- Una buena práctica a la hora de hacer el diseño de un programa es empezar por un nivel conceptual e ir detallando poco a poco conforme se avance en el modelado del problema

- La POO de manera inherente soporta esa práctica, ya que se busca que las soluciones a los problemas se piensen de manera conceptual primero en término de los objetos del problema, y que se vayan pensando los detalles conforme se avance

Objeto

- Un objeto es una abstracción que modela un concepto de algún área
- Todos los objetos tienen nombres, estados, datos y comportamientos
- En ocasiones el término “objeto” se usa como sinónimo de “instancia”

Clase

- Describe un conjunto de objetos similares

- Contiene los datos o atributos y las operaciones sobre esos datos, para lograr que la clase tenga el comportamiento deseado

```
class Circulo:
    """Clase que representa a un círculo"""
    #implementación de la clase
```

Atributo

- Es una característica concreta de una clase
- Son los datos de la clase
- Ejemplo:
    - Clase: Persona
    - Atributos: nombre, edad, dirección, etc

Métodos

- Un método es una operación asociada a una clase

```
class Circulo:
    def __init__(self, radio):
        self.radio = radio

    def calcularArea(self):
        return math.pi*(self.radio**2)
```

Métodos

- Los métodos pueden invocar a otros métodos de la misma clase, mediante el uso del atributo `self`

```
class Circulo:
    def __init__(self, radio):
        self.radio = radio

    def calcularArea(self):
        return self.obtenerPi()*(self.radio**2)

    def self.obtenerPi(self):
        return math.pi
```

Instancia

- Es una manifestación concreta de una clase, i.e., un objeto que tiene valores concretos

```
class Circulo:
    def __init__(self, radio):
        self.radio = radio

    def calcularArea(self):
        return math.pi*(self.radio**2)

>>> a = Circulo(10)
```

El argumento self

- `self` es referencia a la instancia actual
- Cuando se envía un mensaje a un objeto, el algoritmo es el siguiente:
  - El código cliente llama a la instancia del objeto
  - La instancia llama al método de la clase, pasando una referencia a sí misma (`self`)
  - El método usa la referencia pasada para tomar los datos de la instancia

El argumento self

```
class Circulo:
    def __init__(self, radio):
        self.radio = radio

    def calcularArea(self):
        return math.pi*(self.radio**2)

>>> a = Circulo(10)
>>> a.radio
10
>>> a.calcularArea()
314.15926535897933
```

El constructor

- Los objetos son creados por medio de un método llamado el constructor
- En Python, los constructores se llaman `__init__`
- Cada vez que se crea una nueva instancia de la clase, se ejecuta el constructor
- Los parámetros del constructor son usados para inicializar los atributos de la instancia que se está creando

El constructor

```
class Circulo:
    def __init__(self, radio):
        self.radio = radio

    def calcularArea(self):
        return math.pi*(self.radio**2)

>>> a = Circulo(10)
>>> a.radio
10
>>> a.calcularArea()
314.15926535897933
```

Práctica

- Escriba una clase Persona, que tenga los atributos nombre, primerApellido, SegundoApellido, y edad

- Escriba una clase Curso, que tenga de atributo una lista de estudiantes (personas), y que tenga un método para agregar personas al curso, y otro para eliminarlas

- Escriba un método para la clase Curso, que permita imprimir los nombres de los estudiantes, en orden de edad