

1 INTRODUCCIÓN A LA TERMINOLOGÍA DE OBJETOS.

Objeto: colección de datos y funciones que operan en esos datos.

Clase: implementación en software de un objeto.

Encapsulado: es el empaque conjunto de datos y funciones y es básico en la POO. El acceso a los datos de un objeto solo puede realizarse a través de sus propios métodos y en algunos lenguajes se conoce como ocultación de datos.

Mensajes: son la forma en que los objetos se comunican entre sí, un objeto pasa un mensaje a otro y éste contesta ejecutando alguna de sus operaciones.

Atributos: también llamados datos o estructura.

Métodos: son las operaciones permitidas por el objeto.

Instancias: son casos particulares de un objeto. Tienen sus propios datos y pueden utilizar los métodos del objeto del cual han sido definidos.

2 DEFINICIÓN DE CLASES EN PYTHON.

Es posible acceder a los elementos que integran una clase usando el operador punto, ejemplo `math.pi`, `objeto.dato`, `objeto.funcion()`

Si la primera instrucción del cuerpo de la clase es una cadena de texto, ésta se usa como cadena de documentación de la clase.

La definición de un objeto se realiza por medio de una clase, tal como se presenta a continuación:

```
class <nombre>:
    <definición de variables de clase>
    <definición de métodos>
```

Como ejemplo supongamos se desea definir una clase para un objeto cuadrado. Este objeto va a tener un atributo, el lado y va a tener dos métodos para el cálculo del área del cuadrado.

```
class Cuadrado:
    def __init__(self, lado):
        self.lado = lado
    def getLado(self):
        return self.lado
    def setLado(self, nuevo):
        self.lado = nuevo
    def calculeArea(self):
        return self.lado ** 2
    def calculePerimetro(self):
        return self.lado * 4
```

Para crear una instancia del objeto cuadrado se puede asignar a una variable o almacenarlos en alguna otra estructura como una lista o un diccionario.

```
>>> a = Cuadrado(10)
```

En este caso se crea una instancia del objeto Cuadrado, con un largo de 10.

```
>>> a.getLado()
```

```
10
```

```
>>> a.calculaArea()
```

```
100
```