PYTHON ORIENTADO A OBJETOS

Subtemas:

- Clases
- Herencia
- Polimorfismo

Palabras clave: class, self, __init__, __str__

Clase

- En python se define una clase de la siguiente forma:

```
Class NombreClase:

<método 1>
…
…
<método N>
```

Ejemplo

- 01_Fecha: el código muestra la creación de una clase, la creación de dos objetos, la obtención de los atributos y la ejecución de sus métodos.

Uso de self

- El uso de la palabra self es para hacer referencia al objeto que se está creando.
- Se utiliza para acceder al espacio de nombres del objeto.
- Es el primer parámetro de todos los métodos, siempre debemos incluirlo en la definición.
- Al llamar a la función no se coloca, se agrega automáticamente.

Ejemplo

- 02_self: este ejemplo hace uso de las variables de clase y las variables de referencia

Inicialización

- Para poder inicializar los atributos de una instancia existe un método especial que permite actuar sobre la inicialización del objeto.

- El método se llama __init__ y permite cualquier número de parámetros, siendo el primero self que hace referencia al mismo objeto.

- Este método se ejecuta siempre que se crea una referencia u objeto de la clase.

Atributos y métodos privados

- Por defectos los atributos y métodos son públicos.

- Para hacer atributos privados se deben de agregar dos guiones bajos antes del nombre de un atributo o método.

Ejemplo

- 03_Humano: Este ejemplo muestra el uso del constructor de objetos, de atributos y métodos privados.

Herencia

- La herencia se aplica cunado a creamos una clase (clase hija) a partir de otra clase (clase base o padre).

- La clase hija conserva las propiedades y funcionalidades de la clase padre, pero se puede hacer más especializada.

- Cuando se hace herencia, la clase hija puede modificar el comportamiento de uno de los métodos de la clase padre, a lo que se le llama sobreescritura.

Ejemplo

- 04_Herencia: En este ejemplo se muestra el uso de la herencia.

Multiherencia

- Una característica de python es que soporta multiherencia, es decir, una clase puede tener más de dos clases base.

Ejemplo

- 05_Multiherencia: En este ejemplo se muestran el uso de la herencia múltiple.

Polimorfismo

- Aplica cuando un objeto tiene la misma funcionalidad que otro, pero la desempeñan de manera diferente.

Método str

- El método str se hereda de la súper clase object y, por defecto, imprime la localidad de memoria donde se guarda el objeto.

- Es posible sobre-escribir el comportamiento el método str para imprimir, por ejemplo, los atributos del objeto.

Ejemplo

- 06_str: En este ejemplo se muestran el uso de método str que se hereda de object.

Ejemplo

- 07_Polimorfismo: En este ejemplo se muestra el uso sobreescritura de métodos.

Parámetros por defecto.

- Si al crear un objeto se omite algún parámetro del método init, la ejecución del programa producirá un error.

- Python permite inicializar los valores de un objeto cuando éstos no son pasados como parámetros al crear el mismo.

Ejemplo

- 08_parámetros: En este ejemplo se muestra el uso los parámetros por defecto al crear una instancia.

Archivos

- Los archivos en python se consideran apuntadores de tipo file.
- Los archivos se pueden abrir de diferentes modos:

‘r’: lectura
‘w’: escritura
‘a’: añadir
‘b’: lectura en binario

Ejemplo

- 9_archivos: Este código muestra las funciones necesarias para abrir, leer y escribir en archivos.

Ejercicio

- Ahora que ya sabe manejar archivos modificar el programa de tablas de multiplicar hecho en el ejercicio 13 de tuplas para que ahora los resultados sean mandados a un archivo de texto.