**OBJETOS**

Los objetos en informática definen una representación abstracta de las entidades de un mundo real o virtual, con el objetivo de controlarlos o simularlos.

La abstracción es la acción de aislar los componentes esenciales de un objeto de aquellos que no lo son. En POO, esto nos permitirá reducir la complejidad de nuestro código para hacerlo más mantenible y fácil de entender. Pero sobre todo nos dará la capacidad de reutilizar código.

**PYTHON**

Python es un lenguaje de programación de alto nivel y es un lenguaje interpretado, lo que significa que no se compila, ya que ejecuta las instrucciones una a una, sin hacer una compilación previa. Son mas lentos que los lenguajes compilados ya que traducen el programa a la vez que lo ejecuta.

Este es un lenguaje fuerte y dinámicamente tipado, con conteo de referencias. Un leng es fuertemente tipado si dada una variable de un tipo en concreto, no se puede usar como si fuera de otro tipo a menos que se haga un cast. Es dinámicamente tipado ya que una misma variable puede tomar valores de distintos tipos en distintos momentos.

En Python todos los datos son objetos y son tipo referencia. Una variable tipo referencia guarda su dirección en la pila, pero el objeto se almacena en el heap (memoria dinámica). Además, una clase es tipo referencia, de ahí los objetos instancia de esa clase se almacenan en el montículo.

**VARIABLES Y CONTEO DE REFERENCIA**

En Python todas las variables son objetos instancia de una clase (class”int”,class”bool”,etc…).

El tiempo de vida que pasa desde la instanciación del objeto hasta que es destruido se conoce como tiempo de vida del objeto.

Cuando un programa instancia un objeto, se reserva parte de la memoria para guardar las variables de instancia de esa clase.

Cada objeto tiene un campo interno que se conoce como conteo de referencia y este nos indica cuantas variables diferentes se refieren al mismo objeto.

Si el contador de referencias llega a 0, automáticamente se invoca a la función de liberación de memoria para el objeto especifico.

Para eliminar algo de la memoria, se debe asignar un nuevo valor a una variable o salir de un bloque de código, por ej.: una función, donde aquí ocurre la mayor parte de recolección de basura, por eso es recomendable funciones pequeñas y simples.

Pasaje por Valor: Se crea una copia local de la variable dentro de la función, de manera que cualquier modificación no tendrá efecto sobre la original.

Pasaje por Referencia: Se maneja directamente la variable pasada como parámetro, de forma que los cambios realizados dentro de la función le afectarán también fuera, es decir a la variable original.

**MRO**

Define el orden en el que se buscan las clases base al ejecutar un método. Primero se busca el método o atributo dentro de una clase y luego sigue el orden que especificamos al heredar.

**TESTCASE**

TestCase proporciona un conjunto de métodos que permite compartir valores, configurar pruebas, y limpiar la memoria cuando se haya terminado.

Método setUp() de la clase TestCase, se usa para realizar la inicialización de cada prueba, no acepta argumentos y permite realizar una configuración arbitraria antes de ejecutar cada prueba.

Método tearDown() se utiliza para liberar recursos que se ocuparon durante las pruebas.