



Trabajo practico N.º 1

Programación orientada a objetos en Python

1- Escribir un programa en Python que cree una clase representando un círculo. Emplear radio como atributo e incluir métodos para calcular su área y perímetro.

2- Realizar un programa en Python que cree una clase que represente un triángulo.

- Atributos: ladoA, ladoB y ladoC.
- Métodos:
 - perímetro(): devuelve perimetro del triángulo.
 - esEquilatero(): retorna True si los tres lados son iguales.
 - esIsosceles(): retorna True si dos lados son iguales y uno distinto.
 - esEscaleno(): retorna True si los tres lados iguale son distintos.

3- Realizar una clase Contador con un atributo llamado cuenta. Implementar los siguientes métodos:

- Constructor con cuenta opcional iniciada en cero.
- mostrar(): Mostrar en pantalla la cuenta actual
- incrementar(valor): Incrementar la cuenta en valor o por defecto en 1 si no se especifica valor.
- decrementar(valor): Decrementar la cuenta en valor o por defecto en 1 si no se especifica valor.
- reiniciar(): Reinicar la cuenta a cero.

4- Desarrollar una clase Cafetera con:

- Atributos:
 - capacidadMaxima: Cantidad máxima de café que puede contener la cafetera en CC
 - cantidadActual: Cantidad actual de café que hay en la cafetera en CC
- Métodos:
 - Constructor: Considerar por defecto capacidad máxima de 1000 CC y cantidad actual en cero
 - llenar(): Llena cafetera a la capacidad máxima
 - servir(cantidad): sirve una cantidad de café (Verificar que cantidad sea positivo y no servir mas del actual)
 - vaciar(): Vaciar cafetera
 - agregar(cantidad): añade una cantidad de café (Verificar que cantidad sea positivo y no supere el máximo)

5- Escribir una clase Persona con los siguientes atributos y métodos:

- Atributos: nombre, apellido, edad en años, altura en metros y peso en Kg.
- Métodos:
 - constructor,
 - nombreCompleto(): Retorna apellido y nombre con formato: 'apellido, nombre'
 - esMayor(): Retorna True si la persona tiene 18 años o más.
 - imc(): Retorna indice de masa corporal de la persona
 - __str__(): Retorna todos los atributos separados por coma.



6- Realizar una clase CuentaBancaria con los siguientes atributos y metodos:

- Atributos: nombre del titular, apellido del titular, numero de cuenta, saldo de la cuenta.
- Métodos:
 - constructor: con saldo inicial en cero.
 - __str__(): devolver todos los atributos.
 - saldo(): mostrar el saldo de la cuenta
 - ingresar(monto): ingresar un monto a la cuenta verificando monto positivo
 - retirar(monto): retirar un monto a la cuenta verificando monto positivo

7- Incorporar a la clase anterior una lista con el historial de ingresos y retiros de la cuenta. Agregar también el método verHistorial para que muestre los movimientos realizados.

8- Realizar una clase que represente una estructura de datos de tipo 'Pila' con una lista como atributo. Incluir los siguientes métodos:

- constructor: inicializa lista vacía.
- agregar(): agrega un elementos al final de la lista.
- quitar(): quita el último elemento de la lista y lo devuelve.
- estaVacia(): devuelve True si la lista esta vacía.
- cantidad(): devuelve cantidad de elementos ingresados

9- Realizar una clase que represente una estructura de datos de tipo 'Cola' con una lista como atributo. Incluir los siguientes métodos:

- constructor: inicializa lista vacía.
- agregar(): agrega un elementos al final de la lista.
- quitar(): quita el primer elemento de la lista y lo devuelve.
- estaVacia(): devuelve True si la lista esta vacía.
- cantidad(): devuelve cantidad de elementos ingresados