Nuevas Tecnologías de Programación. CESDE 2020
Carlos Steeven Jaramillo Ibargüen.
ACTIVIDADES. 20 abril a 10 de mayo

MATERIAL DE ESTUDIO:

20 abril a 10 de mayo - LINKS:
Carpeta 8. Funciones
https://drive.google.com/drive/u/0/folders/11xdSDzH0dTmuz2dCeU8NXiNI3TGl9nP5

Carpeta 9. Funciones
https://drive.google.com/drive/u/0/folders/1hsM8TyOjrW1i2s9uphnwwPMTHFnDR4Pl

Carpeta 9. Manejo de exepciones
https://drive.google.com/drive/u/0/folders/1n1CVrMlfPNARKbn0b-GgyFCi6LU5qXYi

Conceptos:
https://unipython.com/programacion-orientada-objetos-python/

TALLER A REALIZAR

Para estas dos semanas (incluida la clase de evaluación) se debe realizar una serie de entregables sencillos con conceptos.

Temas a tener en consideración:

- Lenguaje: Python
- Tecnología a usar: POO
- Herramienta: Visual Studio Code, Python env,
- Entregable: Repositorio en github con las clases.
- Equipos: Máximo 3 personas.
- Entregar un documento en word (sólo una portada con integrantes y el contenido) con las conclusiones del trabajo y observaciones.

EJERCICIOS PROPUESTOS
Recuerden que todos los ejercicios deben tener su estructura, pero tambien el código para ejecutarlos. Es decir, deben entregar las clases y tambien las instancias y ejecución de métodos. Adicional a ello, las funciones deben manejar errores de forma básica.

1. Crea una clase Contador con los métodos para incrementar y decrementar el contador. La clase contendrá un constructor por defecto, un constructor con parámetros, un constructor copia y los métodos getters y setters.

Nuevas Tecnologías de Programación. CESDE 2020
Carlos Steeven Jaramillo Ibargüen.
ACTIVIDADES. 20 abril a 10 de mayo

2. Crea una clase llamada Cuenta que tendrá los siguientes atributos: cedula del titular, nombre del titular, fecha de apertura y cantidad (puede tener decimales).

   Los datos del titular serán obligatorios y la cantidad es opcional. Crea dos constructores que cumpla lo anterior.

   Crea sus métodos get, set y toString.

   Tendrá dos métodos especiales:
   - ingresar(double cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
   - retirar(double cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0 y debe decir cuanto fue lo que pudo retirar.

3. Haz una clase llamada Persona que siga las siguientes condiciones:

   Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.

   Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.).

   Sexo sera hombre por defecto, usa una “constante” para ello.

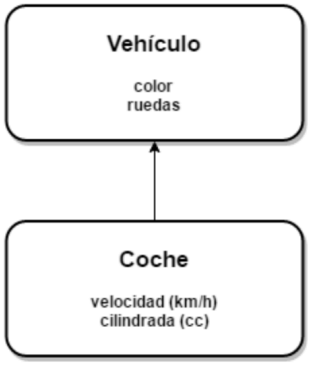
   Se implantaran varios constructores:
   - Un constructor por defecto.
   - Un constructor con el nombre, edad y sexo, el resto por defecto.
   - Un constructor con todos los atributos como parámetro.

   Los métodos que se implementaran son:
   - calcularIMC(): calculara si la persona esta en su peso ideal (peso en kg/(altura^2 en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que esta por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses “constantes” para devolver estos valores.

- esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.
- comprobarSexo(char sexo): comprueba que el sexo introducido es correcto. Si no es correcto, sera H. No sera visible al exterior.
- toString(): devuelve toda la información del objeto como String en un formato elegido por usted.
- generaDNI(): genera un número aleatorio de 8 cifras. Este método sera invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
- Métodos set de cada parámetro, excepto de DNI.

4. Con base en este ejemplo:

Nuevas Tecnologías de Programación. CESDE 2020
Carlos Steeven Jaramillo Ibargüen.
ACTIVIDADES. 20 abril a 10 de mayo

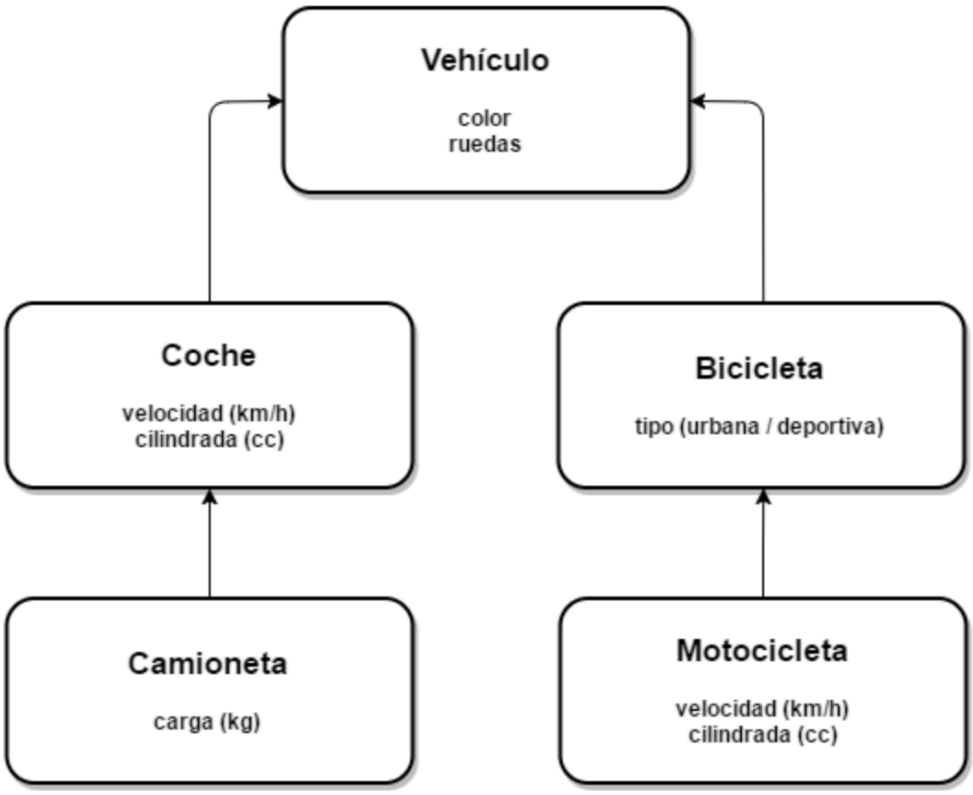


Ejercicio Resultado

```
class Vehiculo():

    def __init__(self, color, ruedas):
        self.color = color
        self.ruedas = ruedas

    def __str__(self):
        return "Color {}, {} ruedas".format( self.color, self.ruedas )

class Coche(Vehiculo):

    def __init__(self, color, ruedas, velocidad, cilindrada):
        self.color = color
        self.ruedas = ruedas
        self.velocidad = velocidad
        self.cilindrada = cilindrada

    def __str__(self):
        return "color {}, {} km/h, {} ruedas, {} cc".format( self.color, self.v


coche = Coche("azul", 150, 4, 1200)
print(coche)
```

Realizar el siguiente (recuerde la herencia)

Nuevas Tecnologías de Programación. CESDE 2020
Carlos Steeven Jaramillo Ibargüen.
ACTIVIDADES. 20 abril a 10 de mayo



- Crea al menos un objeto de cada subclase y añádelos a una lista llamada vehiculos.
- Crear una función catalogar() por fuera de las clases para que reciba una lista de vehículos y un argumento opcional “ruedas” con un valor por defecto de 4. Debe recorrer la lista y mostrar únicamente los vehículos que su número de ruedas concuerde con el valor del argumento. También debe mostrar un mensaje "Se han encontrado {} vehículos con {} ruedas:".