Tarea Numero 2 Semana 4

Yanelis Ayala 4-754-1723

Link https://onlinegdb.com/BJqyEa6A4

```
#Clase llamada nodo

class Persona:
  def __init__(self, nombre, apellido, telefono, direccion):
    self.nombre = nombre
    self.apellido = apellido
    self.telefono = telefono
    self.direccion = direccion
  def getPersona(self):
    return self.nombre + ' '  + self.apellido + ' ' + self.telefono + ' ' + self.direccion


class Nodo:
    #Constructor con los argumentos
    def __init__(self, value=None, izq=None, der=None):
        #Punteros
        self.value = value
        self.izq = izq
        self.der = der

    #Retornar el valor
    def __str__(self):
        return self.value.getPersona()
```

```
    def tieneHijoIzquierdo(self):
        return self.izq != None

    def tieneHijoDerecho(self):
        return self.der != None

#Clase del arbol binario
class aBinarios:
    #metodo constructor y el atributo raiz
    def __init__(self):
        self.raiz = None
    #creamos la funcion agregar
    def agregar(self, elemento):
        if self.raiz == None:
            self.raiz = elemento
        else:
            aux = self.raiz
            padre = None
            while aux != None:
                padre = aux
                if aux.tieneHijoIzquierdo():
                  aux = aux.der
                else:
                  aux = aux.izq

            # print("padre: ", padre)
            # print("aux: ", aux)
            if padre:
```

```
            padre.der = elemento
        else:
            padre.izq = elemento

#se crea metodo mostrar el preorden
def preorden(self, elemento):
    if elemento != None:
        print(elemento)
        # self.preorden(elemento.izq)
        # self.preorden(elemento.der)

#se crea metodo mostrar el posrden (recursividad)
def postorden(self, elemento):
    if elemento != None:
        self.postorden(elemento.izq)
        self.postorden(elemento.der)
        print(elemento)

#se crea metodo mostrar el inorden
def inorden(self, elemento):
    if elemento != None:
        self.inorden(elemento.izq)
        print(elemento)
        self.inorden(elemento.der)

#se crea funcion que permite obtener la reiz
def getRaiz(self):
    return self.raiz
```

```
#creamos un menu
if __name__ == "__main__":
    #llamamos el objeto de arbol binario
    ab = aBinarios()
    while(True):
        #opciones del menu
        print("Arboles_Binarios\n"+
            "1. Agregar\n"+
            "2. Preorden\n"+
            "3. Postorden\n"+
            "4. Inorden\n"+
            "5. Level\n"+
            "6. Salir")

        num = input("ingrese la opcion: ")

        if num == "1":
            nombre = input("Ingrese el nombre: ")
            apellido = input("Ingrese el apellido: ")
            telefono = input("Ingrese el telefono: ")
            direccion = input("Ingrese el direccion: ")
            persona = Persona(nombre, apellido, telefono, direccion)
            nod = Nodo(persona)
            ab.agregar(nod)

        elif num == "2":
            print("Preorden")
            ab.preorden(ab.getRaiz())
        elif num == "3":
```

```
    print("postorden")
    ab.postorden(ab.getRaiz())
elif num == "4":
    print("inorden")
    ab.inorden(ab.getRaiz())
elif num == "5":
    print("encontrar")
    ab.get
elif num == "6":
    exit()
```