

Gimenez Dario Sebastian
con 5
09/10/24

PARCIAL 1 TEMA 1 ESTRUCTURAS DE DATOS UNAB 2024

Ejercicios a resolver utilizando el lenguaje de programación Python 3

- 1) **(4 puntos)**. Utilizando TADs se deberá representar y programar los métodos más importantes para representar la información de un autor de libros y sus correspondientes **publicaciones (libros)** de su autoría. Importante a tener en cuenta es programar los métodos **agregar_libro**, **borrar_libro** e **imprimir_libros**, esta última impresión debe estar condicionada por año de publicación.
- 2) **(2 puntos)**. Implementar una función utilizando recursión para determinar la suma de los números pares de una lista dada. Alternativamente pueden utilizar una solución iterativa **(1 punto)**.
- 3) **(4 puntos)**. Asumiendo que cuenta con la implementación de un árbol binario de búsqueda, devolver una lista con los números pares.

class Autor:

```
def __init__(self, nombre, apellido, libros = []):  
    self.nombre = nombre  
    self.apellido = apellido  
    self.libros = libros  
  
def agregar_libro(self, libro):  
    self.libros.append(libro)  
  
def borrar_libro(self, titulo):  
    for libro in self.libros:  
        if libro.titulo == titulo:  
            self.libros.remove(libro)  
  
def imprimir_libro(self, año_publicacion, titulo):  
    print(f'Libros Publicados por {self.nombre}')  
    for libro in self.libros:  
        if libro.año == año_publicacion and libro.titulo == titulo:  
            print(libro)
```

class Libro:

```
def __init__(self, titulo, año, editorial):  
    self.titulo = titulo  
    self.año = año  
    self.editorial = editorial  
  
def __str__(self):  
    return f'Titulo: {self.titulo} - Año Publicacion: {self.año} - ED: {self.editorial}'
```

```
autor1 = Autor('Sebastian', 'Garcia')  
libro1 = Libro('Algoritmos 1', 2018, 'Fenix Ediciones')  
autor1.agregar_libro(libro1)  
autor1.imprimir_libro(2018, 'Algoritmos 1')  
autor1.borrar_libro('Algoritmos 1')
```

12

```
def suma_pares (lista):  
    suma = 0  
    if lista == []:  
        return 0  
    elif lista[0] % 2 == 0:  
        return suma + suma_pares (lista[1:])  
        suma +=  
    else:  
        return suma + suma_pares (lista[1:])
```

13

```
def listar_pares (arbol):  
    pares = []  
    if arbol is not None:  
        listar_pares (arbol.izquierda)  
        if arbol.elemento % 2 == 0:  
            pares.append (arbol.elemento)  
        listar_pares (arbol.derecha)
```