

Contenidos

- Pilas

Ejercicios

Ejercicio 1

Implemente la clase Pila con los métodos:

- Push
- Pop
- Top
- Empty

Ejercicio 2

Escriba el algoritmo intercambio, el cual toma dos pilas A y B como parámetro y como resultado hace que el contenido de ambas pilas se intercambie, quedando la pila A tal como era la pila B en un principio, y viceversa. Como variable auxiliar sólo puede emplear una única pila.

Ejercicio 3

“Segunda vida” es una plataforma virtual donde las personas pueden pasar sus ratos libres jugando a ser residentes de un mundo virtual. Usted, uno de los programadores del sistema, detecta que en la cocina, cuando se maneja una pila de platos, se pueden sacar y dejar vajilla de modo indistinto (el plato se pone y saca de cualquier parte de la pila de platos) lo que contradice la lógica común de sacar y dejar platos al tope de la pila. Dado este detalle (que resta realidad al juego) es que decide “reimplementar” parte del código y para ello cuenta con las siguientes clases:

```
class Plato:
    def __init__(self, color, diametro):
        self.color = color
        self.diametro = diametro

class PilaDePlatos:
    def __init__(self):
```

```
self.tope = None

def is_empty():
    pass

def push():
    pass

def pop():
    pass

def cantidad_por_color():
    pass
```

Su objetivo es el siguiente:

- Implementar `is_empty()` que retorne true si no hay platos en la Pila de platos.
- Implementar `push()` para que cada plato nuevo sea puesto en el tope de la pila.
- Implementar `pop()` para que cada plato sea sacado del tope de la fila.
- Implementar `cantidad_por_color()` para que, de acuerdo a un color entregado como parámetro, se retorne el número de platos del color elegido.