

▼ Clase 21: Trabajo dirigido con objetos

Problema.






Queremos leer las líneas de un archivo y grabarlas al revés (en orden inverso) (1º la última, 2º la penúltima, ..., última la 1ª).

Indicación.

- Use un **Stack (Pila)**
- **Stack:** es un contenedor que opera con el principio LIFO (Last In First Out), es decir, el último valor que ingresa es el primero que sale.

Para comprender la abstracción sirve imaginar una pila de libros, en que apilamos los libros uno sobre otro. Luego al sacar los libros, iremos sacando el último que pusimos primero. El funcionamiento de una pila se puede implementar fácilmente con una lista.

Una clase Stack tendrá las siguientes operaciones:

Operación	significado	explicación
s=Stack()		crear stack vacío
s.push(x)		Poner x en stack s
s.pop()		Sacar y entregar un valor de stack s
s.reset()		vaciar stack s
s.empty()	¿  ?	True si stack s está vacío

Revise en detalle la implementación de Stack presentada a continuación, preocúpese de entenderla y ejecutarla con un archivo de prueba creado por Ud. Luego inspeccione el archivo de salida.

▼ Clase Stack

```
#__L: list(any)
class Stack:

    #constructor
    def __init__(self):
        self.__L = []

    # empty: None -> bool
    # retorna True si la lista esta vacia
    def empty(self):
```

```

def empty(self):

    return len(self.__L)==0

# reset: None -> None
# efecto: deja la pila vacia
def reset(self):

    self.__L = []

# push: any -> None
# efecto: agrega x al final de __L
def push(self,x):

    self.__L.append(x)

# pop: None -> any
# efecto: saca el ultimo elemento de __L
def pop(self):

    return self.__L.pop()

# Tests
s=Stack()
assert s.empty()
s.push("a")
assert not s.empty()
s.push("b")
s.push("c")
assert s.pop()=="c"
s.reset()
assert s.empty()

```

Ahora probemos que funcione! (cree un archivo de prueba y entreguelo al programa)

```

#leer líneas y ponerlas en stack
s=Stack()
archivo=input("archivo? ")
a=open(archivo)
for linea in a:
    s.push(linea)
a.close()

#sacar y grabar líneas de stack
a=open(archivo,"w");
while not s.empty():
    a.write(s.pop())
a.close()

```

Ejercicio 11 (entregar vía u-cursos)

Una **Queue** (Cola) es un contenedor que opera con el principio **FIFO (First In First Out)**, es decir, el primer valor que ingresa es el primero que sale.

Para entender esta abstracción sirve pensar en una cola de gente esperando entrar al cine. El primero que llega la cola es el primero en entrar al cine.

Ejercicio: Escribir la clase Queue e incluir tests.

Operación	significado
q=Queue()	_____
q.enqueue(x)	_____ ← x
q.dequeue()	← _____
q.reset()	_____
q.empty()	¿_____?