

TALLER 3

Estructura de Datos y Laboratorio

Docente: Ricardo Franco Ceballos

Instituto Tecnológico Metropolitano

Instrucciones: Lea atentamente las instrucciones. El correcto seguimiento de éstas también hace parte de la calificación.

- Si el código presenta **errores de ejecución**, automáticamente la nota será de **0,0**.
- Igualmente, se penalizará la **copia** de códigos. Sean honestos con ustedes mismos.
- El momento evaluativo se puede presentar en grupos de hasta **3 estudiantes**.
- **TODOS** los códigos utilizados deben ir debidamente marcados en comentarios en la parte superior. Indicando los integrantes y el grupo al cual pertenecen.
- Envíe los códigos utilizados en un **comprimido** al correo ricardofranco9295@correo.itm.edu.co. Con el asunto **TALLER 3– Grupo X**, indique en el cuerpo del correo los nombres de todos los integrantes.
- El límite para la recepción de la solución es el 22 de abril de 2022 a las 10 p.m.
- Los métodos implementados únicamente pueden hacer uso de estructuras de Pilas y Colas. **NO** utilizar: arreglos, listas de Python, listas simples, etc.

Implemente las clases QueueCollection y StackCollection teniendo en cuenta los siguientes diagramas de clase y en módulos separados.

StackCollection
-data:Stack
+ StackCollection() +add(Object e) +remove(Object e):Object e +find(Object e):Object e +printStack()

QueueCollection
-data:Queue
+ QueueCollection() +add(Object e) +remove(Object e):Object e +find(Object e):Object e +printQueue()

StackCollection

- data: Corresponde a una Pila implementada con listas simples
- add (**1.0**): Método encargado de insertar un nuevo objeto en la pila de forma tal que ésta quede organizada de mayor a menor, es decir en el tope de la pila debe quedar el valor mínimo.
- Remove (**0.5**): Método encargado de eliminar un objeto de la pila.
- Find (**0.5**): Método encargado de buscar un objeto en la pila.
- printStack (**0.5**): Método que se encarga de imprimir todos los valores de la pila.

QueueCollection

- data: Corresponde a una Cola implementada con listas simples

- **add (1.0):** Método encargado de insertar un nuevo objeto en la cola de forma tal que ésta quede organizada de mayor a menor, es decir el primero de la cola debe ser el valor máximo.
- **remove (0.5):** Método encargado de eliminar un objeto de la cola.
- **find (0.5):** Método encargado de buscar un objeto en la cola.
- **printQueue (0.5):** Método que se encarga de imprimir todos los valores de la cola.

Nota: Todos los métodos implementados DEBEN garantizar la integridad de los datos. Es decir, se debe garantizar que las estructuras queden organizadas de acuerdo con la regla establecida por los métodos add.

Se incluye el método “main.py”. Este **NO** debe ser modificado.

main.py	
	<pre> import QueueCollection as qc import StackCollection as sc # Queue myQueue=qc.QueueCollection() myQueue.add(3) myQueue.add(4) myQueue.add(7) myQueue.add(9) myQueue.add(1) myQueue.printQueue() print(myQueue.find(5)) myQueue.printQueue() print(myQueue.remove(3)) myQueue.printQueue() # Stack myStack=sc.StackCollection() myStack.add(3) myStack.add(4) myStack.add(7) myStack.add(9) myStack.add(1) myStack.printStack() print(myStack.remove(3)) myStack.printStack() print(myQueue.find(3)) myStack.printStack() </pre>
Output	
	<pre> 9 7 4 3 1 None 9 7 4 3 1 3 9 7 4 1 1 3 4 7 9 3 1 4 7 9 None 1 4 7 9 </pre>

