Ejercicios Clase 4

Ejercicio 1

Implementar la interfaz Queue vista en clase que permite especificar el TDA Cola.

Ejercicio 2

Implementar la clase ArrayQueue que implementa la interfaz Queue del ejercicio anterior utilizando un arreglo circular.

Ejercicio 3

Implementar la clase LinkedQueue que implementa la interfaz Queue del ejercicio 1 utilizando una estructura de nodos enlazada.

Ejercicio 4

Implemente un método que reciba dos pilas de colas de enteros P1 y P2, y retorne una nueva pila de colas Pout que sea la unión de los elementos de P1 y P2. Cabe aclarar que los elementos de P1 y P2 están ordenados de menor a mayor en función de su tamaño, y que la pila Pout debe quedar ordenada del mismo modo. En el tope de las pilas se encuentra el elemento de mayor tamaño. Resuelva el problema planteado exclusivamente en términos de las operaciones de los TDA Pila y TDACola.

Ejercicio 5

Implementar la interfaz PriorityQueue vista en clase que permite especificar el TDA Cola con Prioridad a través de claves y valores.

Ejercicio 6

Implementar la clase ArrayPriorityQueue que implementa la interfaz PriorityQueue del ejercicio anterior utilizando un arreglo circular.

Ejercicio 7

Implementar la clase LinkedPriorityQueue que implementa la interfaz Queue del ejercicio 5 utilizando una estructura de nodos enlazada.

Ejercicio 8

Implementar un programa de un hospital que permita ingresar pacientes con una prioridad determinada y llamar pacientes para ser atendidos por el doctor según la prioridad con la que ingresaron. Pruebe utilizar una estructura dinámica y una estática, ¿cuál es más apropiada para este problema?