



Trabajo Práctico Nº 05

Tema: La clase Cola

1. Mostrar que es lo que se escribe por cada uno de los siguientes segmentos de código, dado que COLA es un objeto de la clase Cola de enteros, PILA es un objeto de la clase Pila de enteros y X, Y y Z son variables enteras:

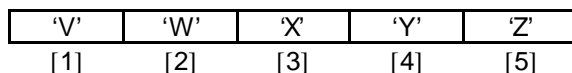
```
a. COLA = new Cola();
   X = 0;
   Y = 1;
   COLA.insertar(X);
   COLA.insertar(Y);
   Y = COLA.borrar();
   Z = Y + 5;
   while (! COLA.estaVacia())
   {
     Z = COLA.borrar();
     System.out.println(Z);
   }
```

```
b. PILA = new Pila();
   COLA = new Cola();
   X = 0;
   Y = 1;
   Z = X + Y;
   while (Z < 10)
   {
     if ((Z % 2) == 0)
       PILA.meter(Z);
     else COLA.insertar(Z);
     X = Y;
     Y = Z;
     Z = X + Y;
   }
   System.out.println("Pila contiene:");
   while (! PILA.estaVacia())
   {
     Z = PILA.sacar();
     System.out.println(Z);
   }

   System.out.println("COLA contiene:");
   while (! COLA.estaVacia())
   {
     Z = COLA.borrar();
     System.out.println(Z);
   }
```

2. En los ejercicios af mostrar el resultado de las operaciones dadas sobre las colas, usando la última implementación vista. Si se presenta desbordamiento positivo o negativo, marcar el correspondiente lugar; caso contrario mostrar los cambios en la cola.

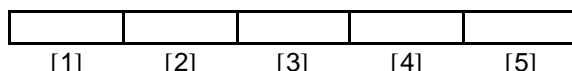
a. elementos



frente = 5
ffinal = 4

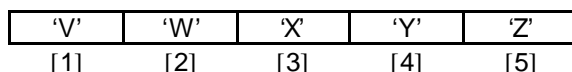
cola.insertar('j'); Desbordamiento?..... Desbordamiento Negativo?.....

elementos



frente =
ffinal =

b. elementos



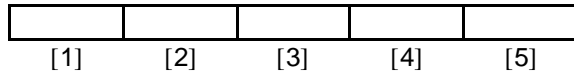
frente = 4
ffinal = 5

cola.insertar('k'); Desbordamiento?..... Desbordamiento Negativo?.....



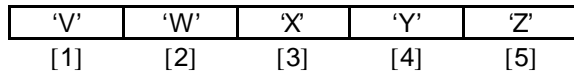
Trabajo Práctico Nº 05

elementos



frente =
 ffinal =

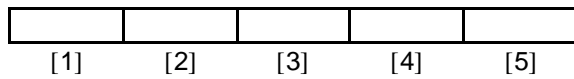
c. elementos



frente = 1
 ffinal = 4

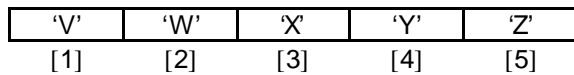
cola.insertar('j'); Desbordamiento?..... Desbordamiento Negativo?.....

elementos



frente =
 ffinal =

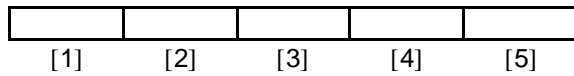
d. elementos



frente = 3
 final = 3

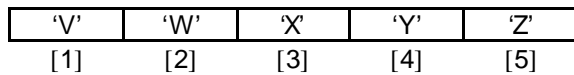
cola.borrar(); Desbordamiento?..... Desbordamiento Negativo?.....

elementos



frente =
 ffinal =

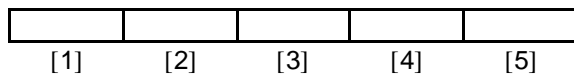
e. elementos



frente = 5
 ffinal = 3

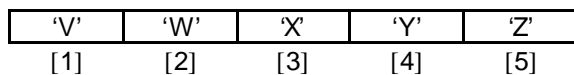
cola.borrar(); Desbordamiento?..... Desbordamiento Negativo?.....

elementos



frente =
 ffinal =

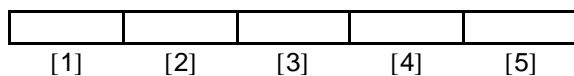
f. cola.elementos



cola.frente = 2
 cola.final = 1

cola.borrar(); Desbordamiento?..... Desbordamiento Negativo?.....

elementos



frente =
 ffinal =

3. Escriba la clase Cola cuya declaración de atributos y métodos correspondan a la implementación Frente Fijo y Final Movable.(tenga en cuenta que al borrar debe correr todos los elementos)

Trabajo Práctico Nº 05

4. En base a la implementación Frente y final movable, escribir un método **contar** que devuelva la cantidad de elementos en la cola.

5. Utilizando las clases Cola y Pila, escribir un algoritmo que invierta los elementos de una cola.

6. Escribir un método que reciba un valor N, saque N elementos de un objeto pila y luego los coloque en una cola. El procedimiento ha de comprobar cualquier posible condición de error y devolver un indicador booleano a Verdad si se encuentra un error. Determinar que otros parámetros se necesitarán para ser recibidos o devueltos y declararlos en la cabecera del método. Puede suponer que las operaciones básicas de la pila y la cola ya han sido codificadas.

7. Escribir un programa que lea una cadena de caracteres, metiendo cada carácter en una pila a medida que se lee y añadiéndolo simultáneamente a una cola. Cuando se encuentre el final de la cadena, utilice las operaciones básicas de pilas y colas para determinar si la cadena es un palíndromo.

8. Un estacionamiento de autos contiene una sola línea, la cual tiene capacidad para 10 autos. Los autos llegan al extremo sur y salen por el norte, si un cliente llega a retirar su auto que no está muy hacia el norte, todos los autos al norte de su auto son retirados, el auto del cliente sale, y los otros autos son colocados de nuevo en el mismo orden en el que estaban originalmente. Cada vez que sale un auto, todos los autos que están al sur son movidos hacia adelante de tal manera que en todo momento todos los espacios vacíos están en la parte sur del estacionamiento.

Escriba un programa que procese un grupo de entradas. Cada entrada contienen una "e" para llegada o "s" para salida y una placa con el número de patente. Se asume que los autos llegan y salen en el orden especificado en la entrada. El programa debe imprimir un mensaje que los autos llegan y salen en el orden especificado en la entrada. El programa debe imprimir un mensaje cada vez que llega un auto, el mensaje debe especificar si hay o no espacio para el auto en el estacionamiento. Si no hay espacio para el auto, el auto espera hasta que haya uno, cuando queda espacio libre, debe imprimirse otro mensaje. Cuando sale un auto, el mensaje debe incluir el número de veces que fue movido el auto para permitir que otros autos salieran.

9. Una variación de la clase Cola es la estructura *Bicola*. Una *bicola* es un conjunto ordinal de elementos en el cual se pueden añadir o quitar elementos de cualquier extremo de la misma. Es, en realidad una cola bidireccional. Los dos extremos de una bicola se llaman *izquierdo* y *derecho*, respectivamente. Las operaciones básicas de una bicola son:

<i>crear(Bq);</i>	inicializa una bicola sin elementos
<i>esVacia(Bq);</i>	devuelve verdadero si la bicola no tiene elementos
<i>insertarIzquierda(x);</i>	añade un elemento por extremo izquierdo
<i>insertarDerecha(x);</i>	añade un elemento por extremo derecho
<i>eliminarIzquierda();</i>	devuelve el elemento izquierdo y lo retira de la bicola
<i>eliminarDerecha();</i>	devuelve el elemento iderecho y lo retira de la bicola

Para representar una bicola se puede elegir una representación estática en la cual se mantienen los elementos de la bicola con un array circular y dos variables índice del extremo izquierdo y derecho, respectivamente.

a) Considere una bicola de caracteres representada por un array circular. El array consta de 9 posiciones. Los extremos actuales y el contenido de la bicola:

$$\text{Izquierdo} = 5 \qquad \text{Derecha} = 7 \qquad \text{Bicola} = A, C, E$$

Escribir los extremos y la bicola según se vayan realizando las siguientes operaciones:

- Se añaden los elementos F, J a la derecha de la bicola.
- Se añaden los elementos R, W, V a la izquierda de la bicola.
- Se añade el elemento M a la derecha de la bicola.
- Se eliminan dos elementos a la izquierda.
- Se añaden los elementos K, L a la derecha de la bicola.
- Se añade el elemento S a la izquierda de la bicola.

Trabajo Práctico Nº 05

- b) Escribir la clase Bicola con los métodos básicas anteriormente detallados. Probar la clase.
- c) Una bicola con restricción de entrada sólo permite inserciones por uno de sus extremos, permitiendo extraer elementos por cualquier extremo. Definir e implementar las operaciones para este TDA.
- d) Una bicola con restricción de salida sólo permite extraer elementos por uno de sus extremos, permitiendo insertar elementos por cualquier extremo. Definir e implementar las operaciones para este TDA.
10. Usar la siguiente información para los ejercicios a, b y c. Un sistema operativo particular almacena los trabajos en unas colas de espera para ejecutarlos de acuerdo con el siguiente esquema:

- Los usuarios del sistema que tengan prioridades relativas basadas en sus números de identificación:

Usuarios 0-99	mayor prioridad (presidente, vicepresidente y gerentes de la empresa)
Usuarios 100-199	siguiente más alta prioridad (subgerentes y asesores)
Usuarios 200-299	siguiente más alta prioridad (jefes de departamentos y secretarios)
-	
-	
Usuarios 800-899	siguiente a la más baja (programadores)
Usuarios 900-999	más baja (trabajos que se ejecutaran solo los sábados a las 3 de la madrugada).
- Dentro de cada grupo de prioridad los trabajos se ejecutan en el orden en que llegan al sistema.
- Si hay un trabajo de prioridad más alta, ejecutarlo antes que otro trabajo, si no, si hay un trabajo de prioridad siguiente a la más alta, ejecutarlo antes de cualquier trabajo de prioridad más baja y así sucesivamente. Esto es, un trabajo de prioridad más baja se ejecutará solo cuando no haya trabajos de prioridad más alta en espera.
- El sistema tiene un array de cola FIFO para almacenar las colas de los distintos niveles de prioridad Cola Trabajos [10]

Para realizar los siguientes ejercicios, puede llamar a cualquiera de las operaciones de cola especificadas en el paquete de cola FIFO.

- a. Escribir un método añadirTrabajo que reciba un número de identificación del usuario y un Token (representando el trabajo que hay que ejecutar) y añada el token a la cola adecuada para el nivel de prioridad de dicho usuario.
- b. Escribir un método obtenerSiguieteTrabajo que devuelva el Token del trabajo que este en la cola de mayor prioridad para ejecución (El Token debe ser quitado de la cola)
- c. Hay que apagar el sistema para mantenimiento. Todos los trabajos que están esperando para ejecutarse se quitaran de las colas de trabajo. Sin embargo, este es un sistema muy amistoso por lo que notifica a los usuarios que sus trabajos serán destruidos y que deben por lo tanto, enviar los trabajos más adelante. El método Notificar, con Token e IdMensaje como entrada, realizará esta notificación. Escribir un método limpiarTrabajos que mande el mensaje a todos los usuarios con trabajos en las colas (llamar al método Notificar para hacer esto) Por supuesto, se envían primero los mensajes a los usuarios de mayor prioridad.

11. Codificar y probar la clase Cola de Object.