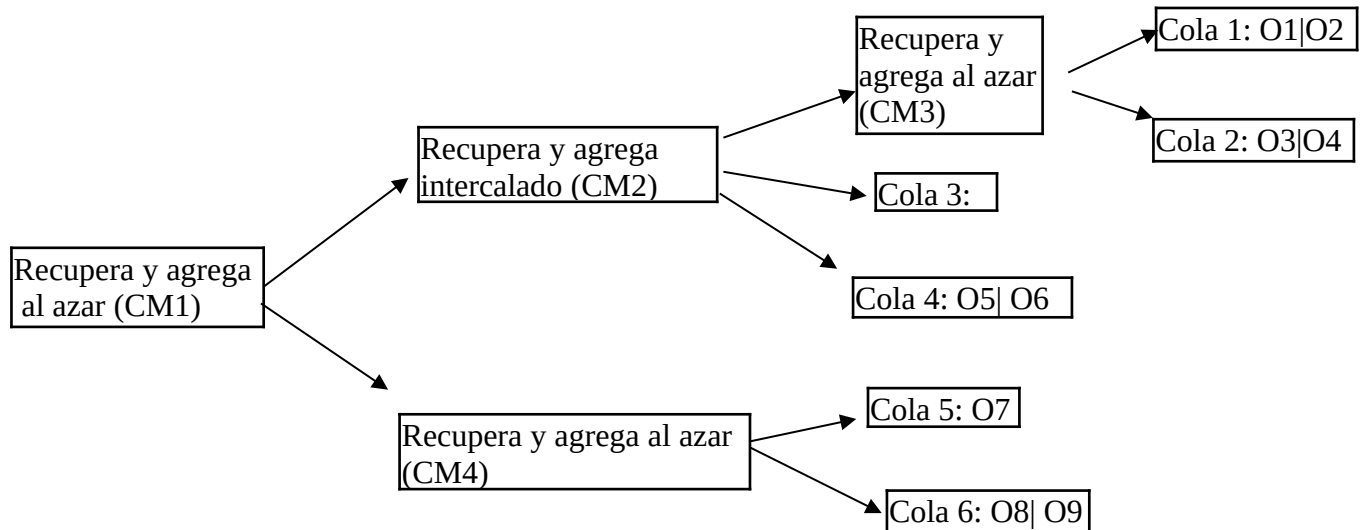


# Programación Orientada a Objetos - Práctica Especial

**Defina las clases (nombre, superclase, atributos y métodos) para implementar una solución orientada a objetos para el siguiente problema e implemente en Java.**

**El ejercicio se debe resolver proveyendo soluciones para cada uno de los puntos y respetando la particularidad de cada uno de ellos**

1) Se debe implementar una estructura de almacenamiento de múltiples colas de **objetos** con múltiples niveles. La misma debe permitir almacenar y recuperar los objetos que se guardan en esta. Por ejemplo (**es sólo un posible ejemplo**):



La estructura puede ser vista como una cola a la cual se le solicita que recupere o guarde un elemento. Los elementos se almacenan en colas terminales FIFO (primer elemento en entrar será también el primero en salir). Una cola múltiple (CM) recupera y almacena un elemento de alguna de las sub-colas y de variadas formas:

- Agrega y recupera al azar
- Agrega y recupera intercalado

Por ejemplo si se **agrega** el objeto **O10** a la cola **CM1** podría darse este camino de ejecución:

1. Se quiere agregar **O10** a la cola **CM1**
2. La cola **CM1** al azar selecciona la cola **CM4** para agregar **O10**
3. Se quiere agregar **O10** a la cola **CM4**
4. La cola **CM4** al azar selecciona la **Cola 5** para agregar **O10**
5. Se agrega a la **Cola 5** el **O10** quedando la **Cola 5** con los objetos: **O7 | O10**

**Cola 5: O7 | O10**

## Programación Orientada a Objetos - Práctica Especial

Por ejemplo si se quiere **recuperar** un objeto de la cola **CM1** podría darse este camino de ejecución:

1. Se quiere recuperar un **objeto** de la cola **CM1**
2. La cola **CM1** al azar selecciona la cola **CM2** para recuperar
3. Se quiere recuperar un **objeto** de la cola **CM2**
4. La cola **CM2**, suponiendo que en la operación anterior (agregar o recuperar) utilizó la sub-cola **Cola-3**, utilizará la siguiente cola para recuperar (**Cola 4**).
5. Se recupera y se retorna el **objeto O5** de la **Cola 4** quedando esta con el objeto **O6**:

Cola 4: O6

6. La cola **CM2** retorna el objeto **O5**
7. La cola **CM1** retorna el objeto **O5**

Además debe permitir calcular:

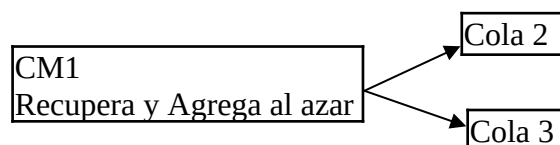
- \* la cantidad **total** de elementos de la estructura
- \*devolver la cola **terminal** que tiene más elementos
- \*determinar la cantidad de colas **terminales** que contiene una cola (para el ejemplo, la primer cola retorna 6 )

**La solución que defina no debe permitir variar la forma de agregar y recuperar de una cola y siempre debe ser definida estáticamente.**

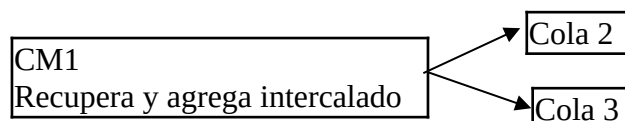
2) Extienda la solución anterior para poder almacenar el **tipo primitivo int** en las colas. Justifique la solución.

3) Como seria ahora la solución si se **podiera variar en tiempo de ejecución la forma de agregar y recuperar de una cola**.

Por ejemplo la Cola CM1 en un momento dado **recupera y agrega** al **azar**:



pero luego se decide cambiar la forma en que agrega y recupera, y ese **misma cola** ahora **recupera y agrega intercalado**:



## Programación Orientada a Objetos - Práctica Especial

4) Como sería ahora la solución si se pudiera **variar en tiempo de ejecución la forma de agregar y recuperar de una cola (punto 3)**, y además las formas de **agregar y recuperar se pueden combinar**, por ejemplo **agrego al azar** y **recupero intercalado** en una misma cola.

Por ejemplo la Cola CM1 en un momento dado **recupera al azar** y **agrega intercalado**:



pero luego se decide cambiar la forma en que agrega y recupera, y ese **misma cola** ahora **recupera intercalado** y **agrega al azar**:



5) Incorporar al punto 4 nuevas formas **de agregar y recuperar**:

- agregue o recupera de la cola con **más** elementos
- agregue o recupera de la cola con **menos** elementos

**Hay que implementar los 5 incisos. También ejemplos de uso de cada uno de ellos.**