|  |
| --- |
|  |

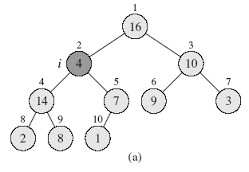
|  |
| --- |
| **Colección Priority Queue**  **Integrantes**: Esteban Mella  Isais escobar  Marcos caniu |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**INTRODUCCIÓN**

La PriorityQueue<E> es una colección utilizada para desarrollar un sistema en el que se pueda ingresar más de un elemento que ayuda y permite al programar ir ordenando los datos según las necesidades , es por ende que viene su nombre de colas de prioridad. Esta cola de prioridad a medida que los datos van siendo ingresados va comparando cada una de los objetos almacenados en el sistema y va asignando los puestos según sea su comparación.

Una colección PriorityQueue es dinámica y no acepta valores nulos.

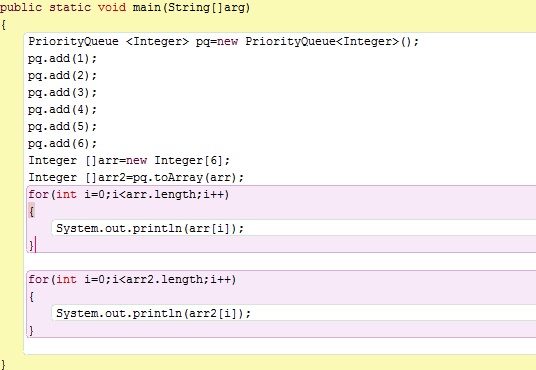
Cada vez que un dato es ingresado se van agregando 2 nodos más que nos permitirá seguir ingresando hasta 1024 elementos a la cola para comparar como un árbol binario que consta de 11 módulos de ordenamiento.



Los tres métodos más importantes de esta colección son:

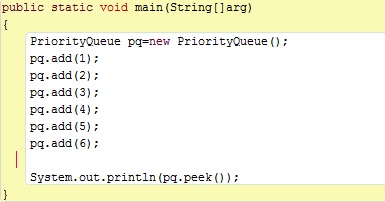
**1.-**toArray () = Devuelve una matriz que contiene todos los elementos en esta cola.

Ejemplo:



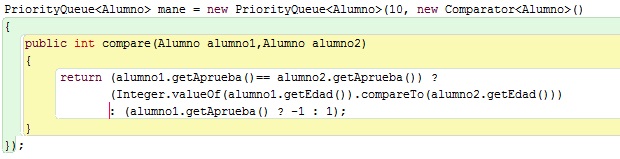
**2.-**peek () = Recupera, pero no elimina, el jefe de esta cola, o vuelve nula si esta cola está vacía.

Ejemplo:



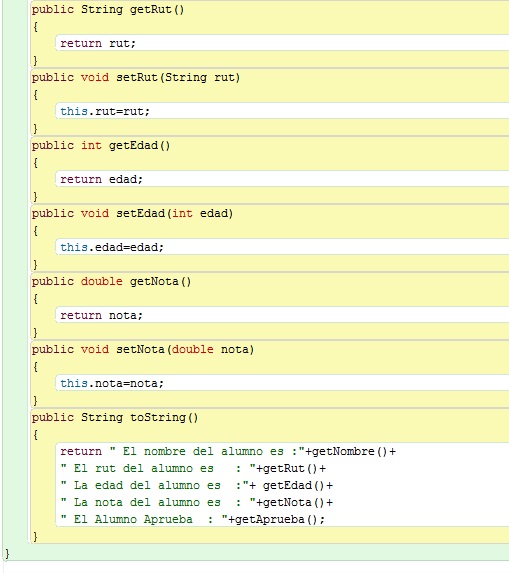
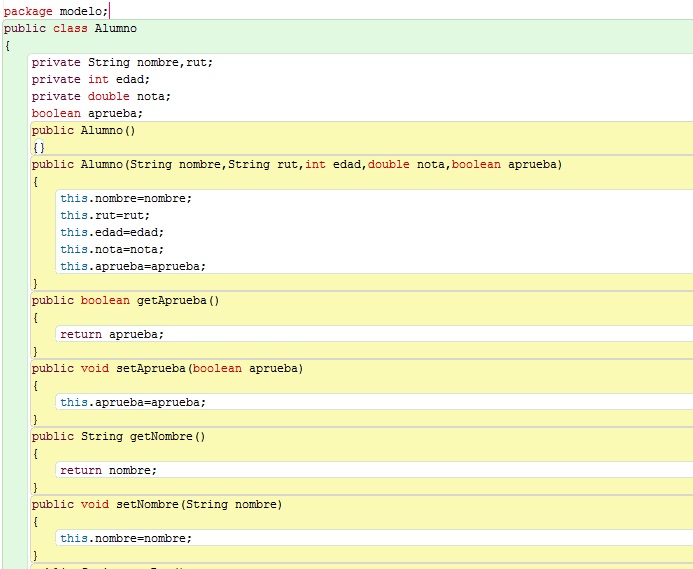
**3.-**Comparator () = Devuelve el comparador utilizado para ordenar los elementos en esta cola, o nulo si esta cola esta ordenada de acuerdo con el orden natural de sus elementos.

Ejemplo:

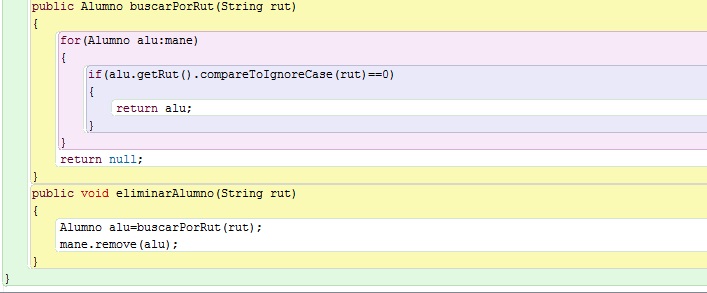
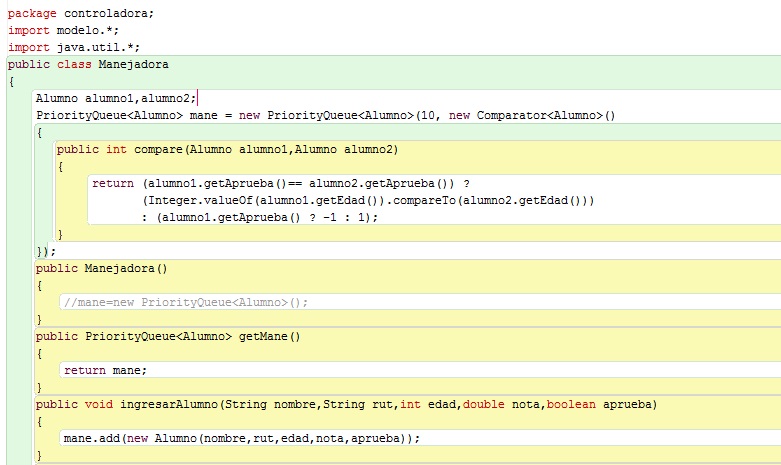


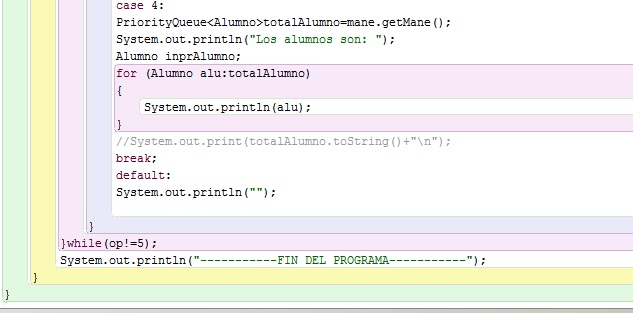
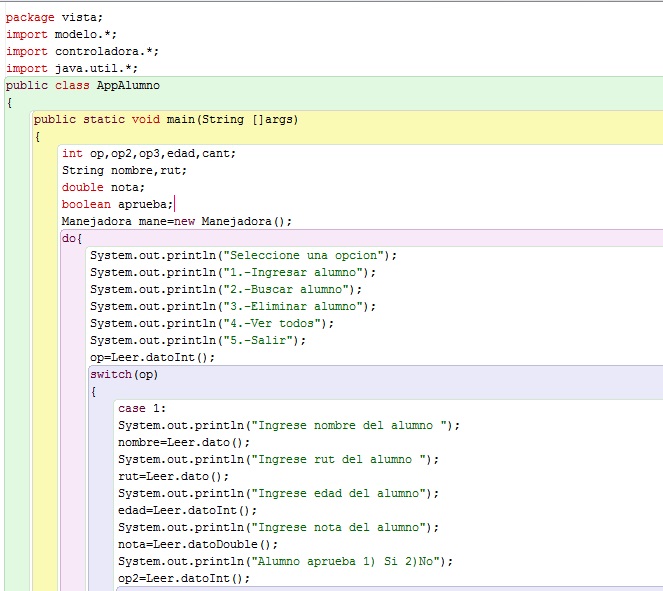
**Proyecto mvc Priority Queue**

**Paquete Modelo**

****

**Paquete Controladora**

****

**PaqueteVista**

**Conclusión**

Esta colección tiene la posibilidad de archivar hasta 1024 elementos donde cada uno de ellos son divididos en nodos de a 2 que se van comparando cada vez que se van ingresando datos y así ir ordenando de mayor a menor por tipo de prioridad, según esta comparación que se construye en el paquete de la controladora.

Los métodos más importantes a modo de resumen fue el método peek () que muestra el dato de mayor prioridad, el toArray () que devuelve una matriz con todos los elementos ingresados a esta cola de prioridad y el Comparator() como método más importante ya que compara los elementos para ordenarlos en el sistema.