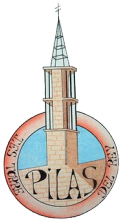
Junta de Andalucía
Consejería de Educación y Deporte



Programación de Servicios y Procesos

2º DAM

PRÁCTICA 1

Toma de contacto con el lenguaje JAVA

Esta práctica pretende servir de toma de contacto con el lenguaje *Java*, que será el empleado en las prácticas de la asignatura, así como con el entorno de trabajo *Netbeans*.

La solución a esta práctica será usada en el futuro para aumentar su funcionalidad.

Se propone la implementación de una **cola circular** usando un vector. Esto implica reutilizar las componentes del vector que contenían elementos ya eliminados, es decir, cuando durante el proceso de añadido lleguemos al final del vector, comenzaremos a llenar de nuevo las componentes iniciales del mismo si se encuentran vacías.

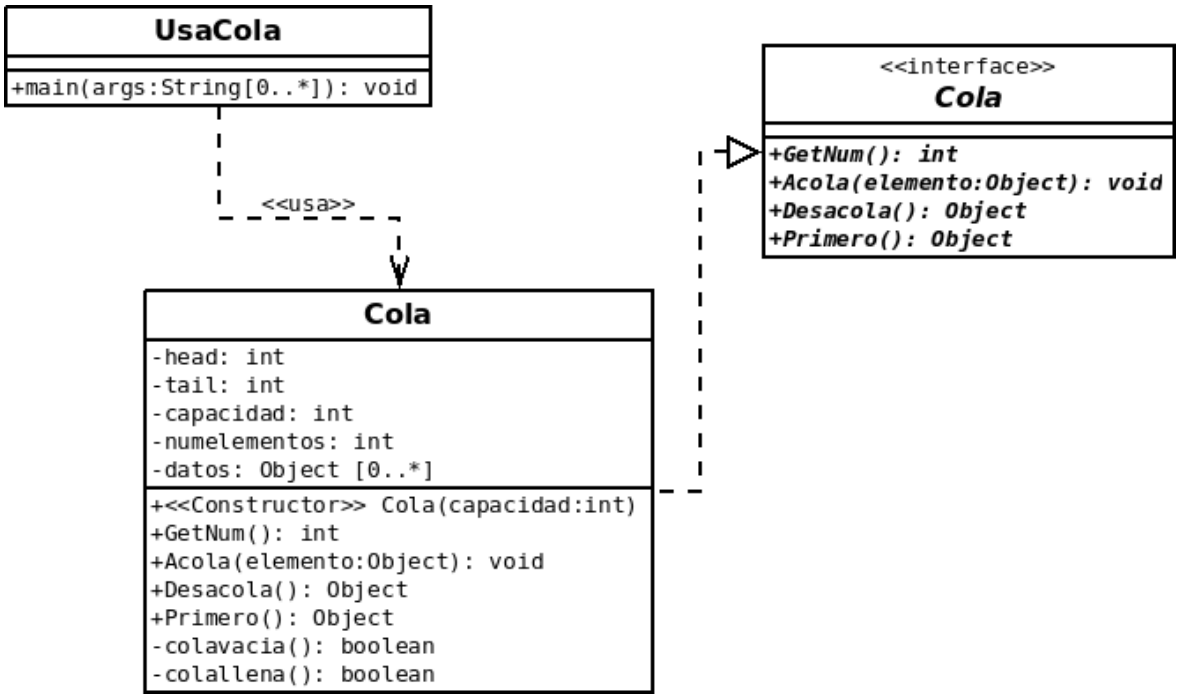
Inicialmente el vector estará vacío y los punteros `head` y `tail` apuntarán a la primera posición del vector.

Cuando se inserta un elemento, si la cola no está llena, se coloca en la posición a la que apunta `head`, y se incrementa su posición.

Cuando se extrae un elemento, si la cola no está vacía, se toma el de la posición a la que apunta `tail` y se incrementa su posición.

Las nuevas posiciones de `head` y `tail`, tras las operaciones de inserción o extracción, se calculan sumando 1 al valor que tienen y calculando el módulo del tamaño del vector, de tal forma que cuando estén en la última posición se coloquen en la primera.

El diagrama de clases es el que sigue:



La descripción de los métodos de la clase Cola son los siguientes:

- **`public Cola(int capacidad)`**
  - Constructor de la clase. Inicializa la cola y los atributos de la clase.
  - Parámetro: `capacidad` - que será la cantidad de elementos con que se inicializa el vector.
- **`public int GetNum()`**
  - Devuelve el número de elementos que hay en la cola
  - Implementa: `GetNum` de `interface ICola`
- **`public void Acola(Object elemento)`**
  - Añade el elemento a la cola si no está llena
  - Implementa `Acola` de `interface Icola`
  - Parámetro: `elemento` - que se acola
  - Throws: `java.lang.Exception` - si la cola está llena
- **`public  Object Desacola()`**

- Extrae el primer elemento de la cola si existe
- Implementa `Desacola` de `interface ICola`
- Devuelve: elemento que se extrae
- Throws: `java.lang.Exception` - si la cola está vacía

- **`public Object Primero()`**
  - Devuelve el primer elemento de la cola sin extraerlo
  - Implementa `Primero` de `interface ICola`
  - Devuelve: elemento que está el primero en la cola
  - Throws: `java.lang.Exception` - si la cola está vacía