

ENUNCIADO DEL EXAMEN – PRÁCTICAS LTP – 2/11/2021

Resuelve los 4 ejercicios de este enunciado de examen en el proyecto BlueJ facilitado.

IMPORTANTE. Se valorará, cuando sean aplicables, el uso adecuado de los mecanismos de **herencia**, **sobrecarga**, **polimorfismo** o **genericidad**.

Contenido del proyecto BlueJ.

Se facilita el siguiente conjunto de clases e interfaces:

- **Coleccion<T>**: Interfaz que modela las operaciones sobre una colección de elementos.
 - **Vector<T>**: Interfaz que extiende a **Coleccion<T>**, añadiendo funcionalidades para acceder aleatoriamente a los elementos de una colección.
 - **VectorList<T>**: Clase que implementa la interfaz **Vector**.
 - **Figure**. Clase que guarda información sobre figuras geométricas (concretamente, su posición).
 - **ColeccionUse**: Clase con un método **main** para probar el resto de clases. Esta clase NO se debe modificar. Esta clase compilará cuando se resuelvan los ejercicios.
-

EJERCICIO 1 (1,75 puntos).

Crea una nueva interfaz, llamada **Cola<T>**, que extienda a **Coleccion<T>** definiendo dos nuevas operaciones:

T primero()

T desencolar()

La funcionalidad de estos métodos:

- **primero()** es un método consultor que devuelve el valor del primero de la cola. El primero es el elemento que fue insertado antes que cualquier otro que esté en la cola.
 - **desencolar()** es un método modificador que borra el primero de la cola y devuelve su valor.
-

EJERCICIO 2 (2,75 puntos).

Crea una nueva clase, llamada **ColaList<T>**, que implemente la interfaz **Cola<T>** con las siguientes características:

- Extenderá la clase **VectorList<T>**.
 - No dispondrá de atributos (solamente los recibidos por herencia).
 - El primero de la cola será el primer elemento de **laLista** (atributo heredado de **VectorList**).
-

EJERCICIO 3 (2,75 puntos).

Añade un nuevo método a la interfaz **Coleccion<T>**:

int apariciones(T e)

Este método devuelve cuántas veces un elemento está en una colección.

Implementa este método una sola vez, en el lugar adecuado para que esté disponible para las clases **VectorList** y **ColaList**.

EJERCICIO 4 (2,75 puntos).

Crea una nueva clase, llamada **FiguresCola<T>**, que extienda la clase **ColaList<T>**, pero con restricción de la genericidad a la clase **Figure**. **FiguresCola** no dispondrá de atributos (solamente los recibidos por herencia).

Añade, e implementa, en **FiguresCola** el siguiente método:

double getYprimero()

Este método devolverá el valor del atributo **y** de la figura que esté en la posición primera de la cola de figuras.

TEST DE LOS EJERCICIOS.

Si se resuelven correctamente los ejercicios, la clase **ColeccionUse** compilará y la ejecución de su método **main** generará la siguiente salida:

```
Cuantos en el Vector? (8 : 1) (99 : 0)
ColaList cL : [20.0, 19.0, 18.0, 17.0, 16.0, 15.0, 99.0, 99.0]
primero : 20.0
desencolar : 20.0
primero : 19.0
ColaList pL : [19.0, 18.0, 17.0, 16.0, 15.0, 99.0, 99.0]
Cuantos en la Cola? (8.0 : 0) (99.0 : 2)
FiguresCola cF : [Position: (4.0, 6.5), Position: (3.5, 8.2)]
y del primero de cF: 6.5
```

Obtener esta salida es indicio, pero no garantía, de la resolución correcta de los ejercicios.
