

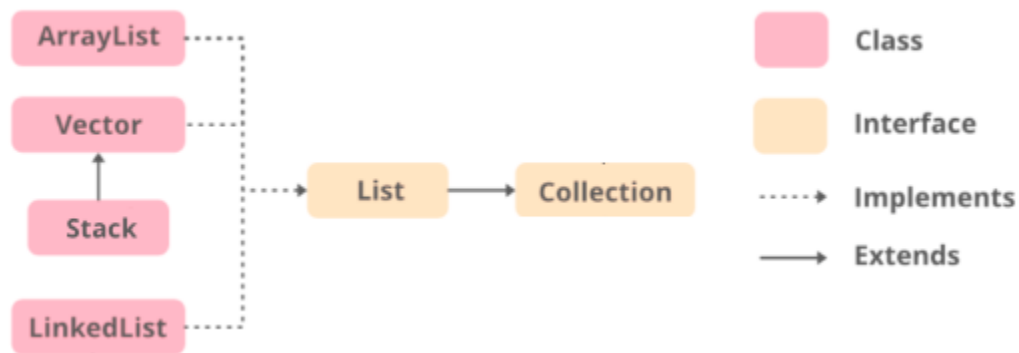
El Framework Collection

Un Framework es un conjunto de clases, clases abstractas e interfaces; las cuales proveen una arquitectura lista para usarse en un área de programación. Java incorpora el Framework **Collection** para manejar una colección de objetos. Cualquier grupo de objetos individuales que se representa como una unidad se conoce como una colección de objetos.

Como se muestra en la siguiente figura el Framework Collection posee una interfaz Collection y esta hereda a la interfaz List.

En Java una interfaz puede heredar a otra y además se permite la herencia múltiple de interfaces. En este caso no se usa implements porque la interfaz List no implementa nada, sólo se hacen extensivas las propiedades de la interfaz Collection.

La interfaz List brinda métodos para administrar una colección ordenada de objetos mediante una lista.



También es posible observar en:

<https://docs.oracle.com/javase/8/docs/api/java/util/List.html>

que la interfaz List hereda de la interfaz Iterable la cual permite iterar a través de la colección de objetos.

Por último, vemos que la clase ArrayList implementa la interfaz List y permite manejar arreglos dinámicos en Java.

12

Ejercicio 12

Basándose en el código del ejercicio 7 en la clase 8, elabore la clase `PoligonoIrreg` la cual representa un polígono irregular en los cuatro cuadrantes del plano cartesiano, donde cada vértice del polígono es un objeto `Coordenada`. Los vértices del polígono se deberán almacenar en un objeto `ArrayList` (véase <https://www.geeksforgeeks.org/list-interface-java-examples/?ref=gcse>).

Se debe disponer de los métodos `anadeVertice` para agregar un nuevo vértice al polígono, es decir un objeto `Coordenada`, y sobrescribir el método `toString()` para imprimir el conjunto de vértices que componen al polígono. En este método debe hacer uso de un bucle `for-each` para imprimir los elementos de `ArrayList` (`for(:){ ... }`).

Pruebe su clase `PoligonoIrreg` al imprimir un objeto recién creado, después agregue tres vértices con valores reales de coordenadas aleatorias comprendidos entre +100.000 y -100.000 y vuelva a imprimir el objeto (la idea es que puedan existir valores positivos y/o negativos con decimales distintos de cero).

13

Ejercicio 13

Incluya en la clase `PoligonoIrreg` el método `ordenaVertices` para ordenar los vértices de menor a mayor en función de su magnitud (distancia de la coordenada al origen) con ayuda de la interfaz `Comparator` (véase <https://www.geeksforgeeks.org/comparator-interface-java/?ref=gcse>). Para resolverlo se debe agregar al objeto `coordenada` un nuevo miembro llamado `magnitud` para facilitar el ordenamiento, así como imprimir la magnitud junto con la coordenada en el método `toString`.

En el método principal instancie un objeto `PoligonoIrreg` al cual posea 10 vértices, imprima los vértices del polígono, mande a llamar el método de ordenamiento y posteriormente vuelva a imprimir los vértices del polígono.