12.11.- Utilizando la clase Contenedor definida en la Actividad resuelta 12.2, implementa una aplicación donde se guardan 30 enteros aleatorios entre 1 y 10 y luego se ordenan de menor a mayor. La aplicación debe mostrar el contenedor antes y después de ordenar.

12.12. Añade a la clase Contenedor el método *void ordenar(Comparator<T> c)*, que ordena los elementos del contenedor según el criterio de c .

12.13. Repite la Actividad de aplicación 12.11 ordenando los números de mayor a menor.

12.14. Añade a la clase Contenedor el método T get(int intIndice), que devuelve el elemento que ocupa el lugar in intIndice dentro del contenedor.

12.15. Implementa un método genérico al que se le pasa una lista de valores de la clase genérica T y devuelve otra donde se han eliminado las repeticiones.

12.16. Implementa una aplicación que gestione los socios de un club usando la clase socio implementada en la Actividad resuelta 12.11. En particular, se deberán ofrecer las opciones de alta, baja y modificación de los datos de un socio. Además, se listarán los socios por nombre o por antigüedad en el club.

12.17. Implementa la clase Cola genérica utilizando un objeto ArrayList para guardar los elementos.

12.18. Implementa la clase Pila genérica utilizando un objeto ArrayList para guardar los elementos.

12.19. Escribe un programa donde se introduzca por consola una frase que conste exclusivamente de palabras separadas por espacios. Las palabras de la frase se almacenarán en una lista. Finalmente, se mostrarán por pantalla las palabras que estén repetidas y, a continuación, las que no lo estén.

12.20. Utilizando colecciones, implementa la clase SuperCola, que tiene como atributos dos colas para enteros, en las que se encola y desencola por separado. Sin embargo, si una de las colas queda vacía, al llamar a su método desencolar( ) , se desencola de la otra mientras tenga elementos. Solo cuando las dos colas estén vacías, cualquier llamada a desencolar( ) devolverá nuil. Escribe un programa con el menú:

1 Encolar en colal .
2. Encolar en cola2 .
3. Desencolar de colal.
4. Desencolar de cola2 .
5. Salir.

Después de cada operación se mostrará el estado de las dos colas para seguir su evolución.

12.21. Implementa una aplicación donde se insertan 20 números enteros aleatorios distintos, menores que 100, en una colección. Deberán guardarse por orden decreciente a medida que se vayan generando, y se mostrará la colección resultante por pantalla.

12.22. Introduce por teclado, hasta que se introduzca «fin», una serie de nombres, que se insertarán en una colección, de forma que se conserve el orden de inserción y que no puedan repetirse. Al final, la colección se mostrará por pantalla.

12.23. Repite la Actividad de aplicación 12.22 de forma que se inserten los nombres manteniendo el orden alfabético.

12.24. Implementa una función a la que se le pasen dos listas de enteros ordenadas en sentido creciente y nos devuelva una única lista, también ordenada, fusión de las dos anteriores. Desarrolla el algoritmo de forma no destructiva, es decir, que las listas utilizadas como parámetros de entrada se mantengan intactas.

12.25. Implementa una aplicación que gestione un club donde se identifica a los socios por un apodo personal y único. De cada socio, además del apodo, se guarda el nombre y su fecha de ingreso en el club. Utiliza un mapa donde las claves serán los apodos y los valores, objetos de la clase Socio. Los datos se guardarán en un fichero llamado «club.dat», de donde se leerá el mapa al arrancar y donde se volverá a guardar actualizado al salir. Las operaciones se mostrarán en un menú que tendrá las siguientes opciones:

1 Alta socio.
2. Baja socio.
3. Modificación socio.
4. Listar socios por apodo.
5. Listar socios por antigüedad.
6. Listar los socios con alta anterior a un año determinado.
7. Salir.

12.26. Un centro educativo necesita distribuir de forma aleatoria a los alumnos de un curso entre los grupos disponibles para ese curso. Diseña la función List<List<String>> repartoAlumnos(List<String> lista, int numGrupos) que devuelve una lista de listas, cada una de las cuales corresponde a un grupo.
Cada nombre de la lista de alumnos se asigna a uno de los grupos.

12.27. Genera un programa que guarde una lista con los verbos irregulares en inglés (verbo, pasado simple). Muestra uno por uno los verbos y cuenta los aciertos que haga el usuario al insertar el pasado simple. Muestra las estadísticas de aciertos y fallos. Repite lo mismo pero sacando verbos de manera aleatoria hasta que el usuario nos indique que no quiere sacar más

12.28. Escribe un programa que genere una secuencia de 5 cartas de la baraja española y que sume los puntos según el juego de la brisca. El valor de las cartas se debe guardar en una estructura *HashMap* que debe contener parejas (figura, valor), por ejemplo ("caballo", 3).
La secuencia de cartas debe ser una estructura de la clase *ArrayList* que contiene objetos de la clase *Carta*. El valor de las cartas es el siguiente: as → 11, tres →10, sota → 2, caballo → 3, rey → 4; el resto de cartas no vale nada.

Ejemplo:

as de oros
cinco de bastos
caballo de espadas
sota de copas
tres de oros

Tienes 26 puntos