

1. Realiza el siguiente menú para gestionar una lista simplemente enlazada de cadenas:

1. Nueva lista
2. Número de cadenas
3. Añadir cadena
4. Eliminar cadena
5. Contiene cadena
6. Mostrar lista entera
7. Salir

En la opción 1, se limpiará toda la lista dejando la lista vacía.

En la opción 5, se le pedirá una cadena al usuario y se le dirá si está o no en la lista.

2. Rellena una lista simplemente enlazada de números decimales de tipo float.

A continuación, presentar el siguiente menú:

1. Mostrar la lista
2. Mostrar siguiente
3. Eliminar último mostrado
4. Salir

En la opción 2, la primera vez que se ejecute saldrá el primero de la lista. La siguiente vez el segundo, y así sucesivamente. Si el último mostrado ha sido el último y se le vuelve a dar a esta opción, entonces se mostrará el primero.

3. Realiza una clase **Caja** que pueda guardar objetos de cualquier tipo.

Realiza para dicha clase los métodos get y set. Utilizando dicha clase, haz un programa que haga lo siguiente:

- Guardar en una lista 10 objetos de la clase Caja que contenga cadenas.
- Recorrer la lista y mostrar por pantalla el contenido de todas las cajas.
- Guardar en otra lista 10 objetos de la clase Caja que contenga números de tipo long.
- Recorrer la lista mostrando por pantalla el contenido de todas las cajas y dar después un resultado de sumarlas todas.

4. Realiza una interfaz que contenga dos métodos donde T es un genérico.

T primero();

T ultimo();

Realiza una clase **Almacen** que contenga una lista de cajas. Dicha clase debe implementar esta interfaz.

5. En la clase que realizaste para leer valores desde el teclado, unifica los métodos que puedas utilizando genéricos.

6. Realiza el siguiente menú para gestionar una pila de números de tipo byte.

1. Nueva pila
2. Consultar elemento
3. Añadir elemento
4. Eliminar elemento
5. Consultar toda la pila
6. Salir

7. Dados los **empleados comerciales** del boletín anterior, sustituir el atributo array de ventas por un ArrayList.

A continuación, hacer una aplicación donde el usuario podrá elegir entre lo siguiente:

1. Nueva lista de empleados
2. Insertar un empleado nuevo
3. Eliminar un empleado
4. Modificar un empleado
5. Consultar un empleado
6. Consultar lista de empleados
7. Mostrar el número de empleados introducidos.
8. Clonar un empleado
9. Salir

En la opción 1, se limpiará toda la lista dejando la lista vacía.

En la opción 2, aparecerá el siguiente submenú:

- 2.1. Añadir un empleado
- 2.2. Añadir un empleado en una posición
- 2.3. Salir

En la opción 2.1. se añade por el final.

En la opción 3, aparecerá el siguiente submenú:

- 3.1. Eliminar por posición
- 3.2. Eliminar por contenido
- 3.3. Salir

En la opción 3.2, habrá que eliminar todos los empleados que sean iguales. (No hay control de empleados duplicados por lo que el mismo empleado puede estar más de una vez).

En la opción 4, se modificará el empleado de una posición concreta. Habrá que dar un mensaje al usuario mostrando el empleado antiguo y el nuevo. El empleado se podrá modificar utilizando los métodos de la interfaz Trabajar.

En las opciones 5 y 8, se utilizará el empleado de una posición concreta. Además en la opción 8, el clon se añade también a la lista.

8. Utilizando la estructura de empleados del ejercicio anterior, realizar el siguiente menú:

1. Mostrar la lista de empleados
2. Mostrar la lista de empleados al revés
3. Buscar nombre de empleado
4. Añadir empleado
5. Salir

En la opción 2, la lista tendrá que mostrarse desde atrás hacia adelante.

En la opción 3, hay que pedirle al usuario un patrón y mostrarle al usuario todos los empleados cuyo nombre cumple el patrón y dónde se encuentran.

En la opción 4, se añadirá un empleado delante de uno que ya esté en la lista. Si éste se repite varias veces, se mostrarán todas las posiciones al usuario y se le preguntará delante de cuál quiere que se añada. Y si no hay ninguno en la lista, se añade al final.

9. Nos han mandado hacer una ampliación para el **puesto de feria** del boletín anterior.

El dueño quiere regalar sorpresas para captar clientes. Para ello, va a introducir sorpresas en una cola y cada vez que un cliente haga una compra, le regalará una sorpresa de la cola. Para ello, quiere que le hagamos el siguiente menú:

1. Nueva cola de sorpresas
2. Regalar sorpresa

3. Añadir sorpresa
4. Clonar sorpresa
5. Mostrar toda la cola
6. Salir

En la opción 1, se limpiará toda la cola dejando la cola vacía.

En la opción 2, se le regalará la sorpresa al cliente y se quita de la cola. Si el premio que le toca al cliente es una muñeca, hay que probar si el bebé funciona bien, es decir, si implementa bien la interfaz.

En la opción 4, se clonará la primera sorpresa de la cola y el clon se añadirá en la cola.

10. Queremos diseñar una aplicación para gestionar los **vehículos** del boletín anterior.

En dicha aplicación, el usuario podrá elegir entre lo siguiente (utilizar una colección sin duplicados):

1. Insertar un vehículo nuevo, es decir, un coche, una moto o un triciclo.
2. Mostrar los vehículos insertados en el mismo orden en el que fueron introducidos.
3. Mostrar el número de vehículos introducidos.
4. Modificar el estado de un vehículo a motor
5. Mostrar sublista al revés.
6. Clonar un vehículo.
7. Salir.

En la opción 4, hay que modificar el vehículo, es decir, pararlo, arrancarlo o subirle/bajarle la marcha. El usuario introducirá el vehículo que quiere modificar según el orden en el que lo insertó. Es decir, si indica el cuarto, la aplicación deberá modificar el cuarto vehículo que se insertó. Si es un triciclo, hay que indicar que no es un vehículo a motor.

En la opción 5, hay que solicitarle al usuario dos índices, con los cuales sustraer una sublista de vehículos para mostrar al revés. Es decir, si el usuario elige el 2 y el 4, la aplicación mostrará el cuarto vehículo, luego el tercero y por último, el segundo.

En la opción 6, el clon se introducirá en la colección, pero al ser una colección sin duplicados, en la opción 2, solamente tendrá que aparecer el objeto una vez.

11. El **veterinario** del boletín anterior necesita que le hagamos un mapa de animales. La clave de dicho mapa será el código de los animales. Realiza el siguiente menú para gestionar el mapa:

1. Nuevo mapa
2. Añadir animal
3. Eliminar animal
4. Número total de animales
5. Mostrar mapa
6. Sorteo
7. Buscar una mascota y hacerla hablar. Comprobar si está enfadada.
8. Clonar una mascota con sus crías.
9. Salir

En la opción 3, se le pedirá al usuario la clave del animal que desea eliminar.

En la opción 4, se mostrarán por cada tipo de animal, es decir, cuántos perros hay, cuántos gatos, cuántos loros y cuántos periquitos. También se mostrará el número total de crías.

En la opción 5, saldrá el siguiente submenú (hay que mostrar también las crías):

1. Mostrar hacia delante
2. Mostrar hacia atrás
3. Salir

En la opción 6, se elegirá aleatoriamente una clave existente en el mapa. Al animal que le toque, se le bajará un 10% la cuota mensual. Utilizar en esta opción una clase interna a un método.

En las opciones 7 y 8, se le pedirá al usuario la clave del animal. En la opción 8, el animal y su clonado se mostrarán por pantalla para comprobar que son iguales. Luego, se modificará del clonado algo de sus crías y se volverán a mostrar por pantalla para comprobar que el cambio no le ha afectado al animal original.

12. Utilizando los animales del ejercicio anterior, hacer un programa que usando un árbol haga el siguiente menú. Utiliza como criterio de ordenación la fecha de nacimiento del animal.

1. Nuevo árbol
2. Añadir animal
3. Mostrar el más pequeño
4. Mostrar el mayor
5. Mostrar árbol
6. Mostrar subárbol
7. Salir

En la opción 6, se solicitarán dos animales al usuario y se mostrarán todos los animales comprendidos entre ellos en el árbol.

13. Modifica el ejercicio anterior para que el criterio de ordenación sea la clave del animal ordenada alfabéticamente. Hacer tres versiones:

1. Utilizando interfaz Comparator
2. Utilizando interfaz Comparator de manera anónima
3. Utilizando un TreeMap