

EJERCICIOS TEMA 7

1. Realiza un programa que introduzca 20 valores aleatorios (entre 0 y 100) en un ArrayList y que luego calcule la suma, la media, el máximo y el mínimo de esos números. Crear una clase Entero con un atributo de tipo int.
2. Escribe un método que ordene dicha colección de números enteros.
3. Realiza un programa equivalente al ejercicio 1 pero en esta ocasión, el programa debe manipular palabras en lugar de números. Añadir la opción de ordenar.
4. Construye un método calculaInversa que devuelva invertida una lista pasada como parámetro. Añade esta opción al ejercicio 3.
5. Construye un método insertaEnOrden, que añade a una lista **ordenada** pasada como parámetro, un nuevo elemento, que también se pasa como parámetro. De forma que la lista siga estando ordenada.
6. Construye un método estaOrdenada que dada una lista como parámetro de entrada, devuelva verdadero si la lista está en orden y falso si la lista está desordenada.
7. Utilizando la clase ArrayList de la interfaz List, construir una clase que implemente una matriz dinámica. Para ello definir una clase Matriz:
 - a. Un atributo m que es un array de ArrayList, de forma que su primer elemento (coincidiría con m[0]) contendrá un ArrayList con la primera fila de la matriz; su segundo elemento contendrá un ArrayList con la segunda fila de la matriz; y así sucesivamente.
 - b. Dos atributos más con el número de filas y el número de columnas de la matriz.
 - c. Un constructor que reciba dos parámetros: el número de filas y el número de columnas, e inicialice m con elementos de tipo Elemento de valor cero.
 - d. Otro constructor que reciba tres parámetros: el número de filas, el número de columnas y un array bidimensional de tipo int, que contendrá los valores iniciales de la matriz.
 - e. Un método set que reciba una fila, una columna y un Elemento y modifique el valor correspondiente a esa fila y columna con el argumento Elemento. Utilice para ello el método set de la interfaz List.
 - f. Un método get que reciba dos argumentos de tipo int y devuelva el Elemento que está en la posición indicada.
 - g. Un método denominado espejo de tal manera que el objeto que invoca se modifique, devolviendo una matriz con su imagen especular:

-1 2 3 7	7 3 2 -1
7 8 8 1	1 8 8 7

- h. Un método denominado trasponer que hace que el objeto que invoca se modifique, devolviendo una matriz traspuesta: cambiar filas por columnas.
8. Dada la siguiente interface:

```
public interface ILibro{  
    public String getTitulo();  
    public int getNumPaginas();
```

```

        public String[] getAutores();
        public String getEditorial();
        public void setTitulo(String t);
        public void setNumPaginas(int nP);
        public void setAutores(String[] a);
        public void setEditorial(String e);
    }

```

Implementar dos funciones en el main:

- a. func1: dada una lista de libros, devuelva el número de libros que tienen al menos 300 páginas y a lo sumo 4 autores: `int func1(List)`
 - b. func2: dada una lista de libros, devuelva verdadero si existe al menos un libro anónimo: `boolean func2(List)`. La clase Libro implementa ILibro. Crear una lista de libros en el main, y probar ambas funciones.
9. La máquina Eurocoin genera una moneda de curso legal cada vez que se pulsa un botón siguiendo la siguiente pauta: o bien coincide el valor con la moneda anteriormente generada - 1 céntimo, 2 céntimos, 5 céntimos, 10 céntimos, 25 céntimos, 50 céntimos, 1 euro o 2 euros - o bien coincide la posición – cara o cruz. Simula, mediante un programa, la generación de 6 monedas aleatorias siguiendo la pauta correcta. Cada moneda generada debe ser una instancia de la clase Moneda y la secuencia se debe ir almacenando en una lista.

Ejemplo:

2 céntimos – cara
 2 céntimos – cruz
 50 céntimos – cruz
 1 euro – cruz
 1 euro – cara
 10 céntimos – cara

10. Realiza un programa que escoja al azar 10 cartas de la baraja española (10 objetos de la clase Carta). Emplea un objeto de la clase ArrayList para almacenarlas y asegúrate de que no se repite ninguna. La clase Carta tendrá los siguientes atributos:

```

private static String[] n = {"as", "dos", "tres", "cuatro", "cinco",
"seis", "siete", "sota", "caballo", "rey"};
private static String[] p = {"bastos", "copas", "espadas", "oros"};
private String numero;
private String palo;

```

Y un constructor vacío, donde se generará aleatoriamente el valor de sus atributos numero y palo.

11. Modifica el programa anterior de tal forma que las cartas se muestren ordenadas. Primero se ordenarán por palo: bastos, copas, espadas, oros. Cuando coincida el palo, se ordenará por número: as, 2, 3, 4, 5, 6, 7, sota, caballo, rey.
12. Eliminar elementos repetidos de una lista de nodos de tipo Entero.
13. Dividir una lista de elementos de tipo Entero en dos sublistas, en una los positivos y en la otra los negativos.

14. Dadas dos listas de elementos de tipo Entero, obtener una sólo con todos los elementos que están en ambas sin repetir.

15. Dadas dos listas de elementos de tipo Entero, obtener una sólo con todos los elementos no comunes a ambas.

16. ENUNCIADO PROYECTO DISCO

Menú de opciones:

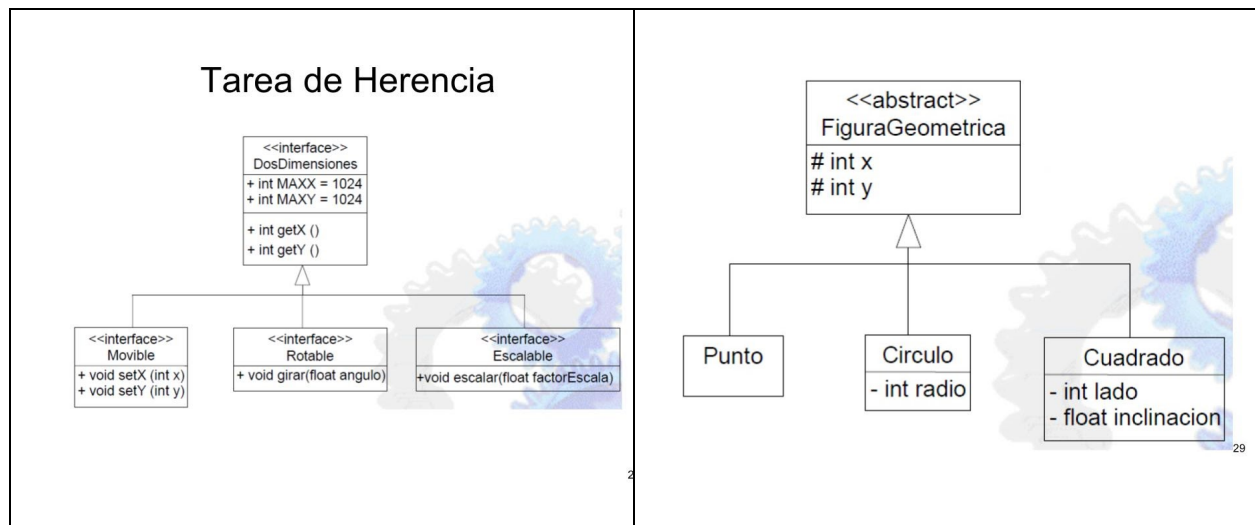
1. Insertar nuevo disco, según su codigo, que es único
2. Listado ordenado según autor
3. Listado agrupado por género, mostrando el número de discos de cada género
4. Listado por duración
5. Eliminar un disco según su código
6. Salir

Clases:

- Disco
- LDisco: insertaDisco, eliminaDisco, listadoAutor, listadoDuracion, listadoGenero.
- PruebaDisco
- GeneroComparator
- DuracionComparator

17. PROYECTO CILINDRO:

- A. Clase Punto (atributos int: x e y; void trasladar(double,double) ;double distancia())
- B. Clase Circulo (atributos Punto centro y double radio:)
- C. Clase Cuadrado (atributos Punto puntoInicial, int lado y float inclinación)
- D. Clase Cilindro (atributos Circulo base y double altura)



La clase abstracta FiguraGeometrica implementa la interfaz DosDimensiones, ya que cualquier figura tiene una posición en el plano.

Punto hereda de FiguraGeometrica, con lo que implícitamente implementa DosDimensiones, pero además implementa Movable.

Circulo implementa Movable y Escalable, pero no Rotable.

Cuadrado implementa Movable, Escalable y Rotable.

ROTAR RESPECTO AL ORIGEN: 90° (-y,x) 180° (-x,-y) 270° (y,-x) 360° (x,y)

18. PROYECTO MEDICAMENTO:

A. Clase Medicamento:

```
public interface Medicamento extends Comparable<Medicamento>{  
    // Métodos consultores  
    Integer getCodigo();  
    String getNombre();  
    Double getPrecio();  
    Boolean getReceta();  
    // métodos modificadores  
    void setPrecio(Double np) throws ExcepcionPrecioMenorCero;  
    void setReceta(Boolean nr);  
    void incrementaPrecio(Integer p);  
}
```

B. Clase MedicamentoImpl: establecer que dos medicamentos no son iguales si tienen códigos distintos (redefinir equals)

ATRIBUTOS:

```
private Integer codigo;  
private String nombre;  
private Double precio;  
private Boolean receta;
```

CONSTRUCTOR:

```
public MedicamentoImpl(Integer c, String n, Double p, Boolean r) throws  
ExcepcionPrecioMenorCero{  
  
    if (p<0){  
        throw new ExcepcionPrecioMenorCero(" Error intento de  
crear un medicamento con precio negativo");  
    }  
    codigo=c;  
    nombre=n;  
    // comprobar que p es positivo  
    precio=p;  
    receta=r;  
}
```

C. Clase ExcepcionPrecioMenorCero: al dar de alta un medicamento, saltará dicha excepción si el precio es negativo.

D. Clase ListaMedicamentos

E. PruebaMedicamento: menú de opciones:

- a. INSERTAR MEDICAMENTO (con su código, nombre, precio y si tiene o no receta)

- b. LISTAR MEDICAMENTOS (ordenados por nombre y en caso de igualdad por su precio)
 - c. INCREMENTAR PRECIO %
 - d. DECIR SI DOS MEDICAMENTOS SON IGUALES
 - e. SALIR
19. Implementar una lista de Elemento, que funcione como una pila de palabras. Menú de operaciones: insertar un elemento (apilar), eliminar una palabra (desapilar), listar. El funcionamiento de una pila consiste en insertar y eliminar el elemento que está en la cima.
20. Implementar una lista de Cadena, que funcione como una cola de palabras. Menú de operaciones: insertar un elemento (encolar), eliminar una palabra (desencolar), listar. El funcionamiento de una cola consiste en insertar por un extremo de la lista y eliminar por el otro extremo de la lista.