Java. Práctica 3

Ejercicio 1. Implementar excepciones en las clases `Lista`, `Pila` y `Cola`. Cada estructura debe proporcionar sus propias excepciones. Así, una `Lista` puede producir la excepción `EmptyListaException`, una `Pila`, `EmptyPilaException` y una `Cola` `EmptyColaException`.
Crear una aplicación que utiliza esas excepciones.

NOTA-1. A partir de ahora se supone que vamos a utilizar el marco de colecciones de Java
NOTA-2. Para los ejercicios en que sea necesario, los separadores de palabras se consideran que son: el espacio, el punto, la coma, el punto y coma, y los dos puntos.

Ejercicio 2. Utilizando una lista para almacenar 100 número aleatorios entre 1 y 100. Usar un iterador para mostrar todos los números. Mostrar también los elementos de la lista sin repetir. Por último, mostrar los elementos de la lista sin repetir y en orden ascendente.

Ejercicio 3. Crear la clase `CuentaW` que en el constructor tome como argumento una cadena de caracteres y sea capaz de mantener información sobre el número de letras y palabras que hay en dicha cadena, así como el número de veces que aparece cada signo (sin diferenciar mayúsculas de minúsculas). Crear los métodos necesarios para poder consultar esa información. Crear una clase que prueba a la anterior con varios ejemplos.

Ejercicio 4. Crear la clase `WUR` que tome una frase en el constructor y almacene por un lado las palabras que aparecen sólo una vez y por otro las que estén duplicadas. Dar métodos que devuelvan estos conjuntos. Crear una aplicación que la utilice.

Ejercicio 5. Crear la clase `PaLinea` que mantenga información sobre las letras que aparecen en una frase (dada en el constructor) junto con una lista ordenada que indica el orden de las palabras en las que aparece. Dar ejemplos de uso.

Ejercicio 6. Crear la clase `Palabra` que mantenga información sobre las palabras que se le agreguen por medio del método `add(String)`. Al final, debe permitirnos conocer el conjunto de palabras de una determinada longitud. Este conjunto deberá estar ordenado alfabéticamente. Crear una aplicación que muestre toda la información que disponga la clase `Palabra`.

Ejercicio 7. Crear la clase `Pila` y la clase `Cola` basada en las colecciones de Java.

Ejercicio 8. Crear la clase `ColaPrior` que representa una cola de prioridades. Para ello, las clases de elementos que pueden insertarse en esta cola deben implementar la interface

```
public interface Prioridad {
     public int prioridad();
}
```

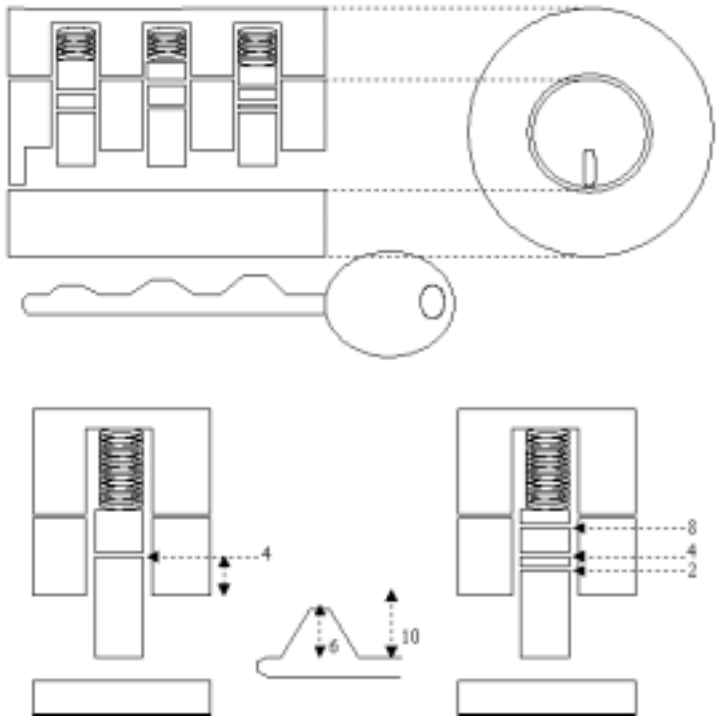
y los elementos en esta cola quedarán ordenados según su prioridad, de mayor a menor. Si se inserta un elemento de una misma prioridad que otro ya existente, se insertará detrás. Crear un par de clase que implementen la interface anterior y realizar pruebas.

Ejercicio 9. (ver figura) Una llave esta formada por un número determinado de “pines”. Cada pin inicialmente tiene una altura de 10 unidades pudiendo limarse hasta que la altura quede en 0.

Una cerradura tiene un número determinado de “bombines”. Cada bombín puede llevar hasta 3 cortes. Los cortes están numerados de 0 a 10 desde la base del bombín.

Una llave abre una cerradura si el número de pines coincide con el número de bombines y además, cada pin empuja el émbolo del bombín de manera que nivela uno de los cortes con la zona de giro. En definitiva, si existe un corte cuyo número sumado con el del pin correspondiente sea 10, ese bombín se abre.

Construir las clases `Llave`, `Cerradura` y una aplicación que prueba si una cerradura se abre con varias llaves dadas.



Ejercicio 10. Crear la clase `Carta` que representa a una carta de la baraja española. Crear la clase `Baraja` que inicialmente contenga a todas las cartas en un monto. Esta clase tendrá un método `void baraja()` que permite barajar el monto de cartas. Otro método, `Carta getCarta()` nos proporciona la carta que está situada en la parte superior del monto retirándola del mismo. Crear una aplicación que utilice esta baraja.