8.11. Crea la clase Campana que hereda de Instrumento (definida en la Actividad resuelta 8.4).

8.12. Las empresas de transporte, para evitar daños en los paquetes, embalan todas sus mercancías en cajas con el tamaño adecuado. Una caja se crea expresamente con un ancho, un alto y un fondo y, una vez creada, se mantiene inmutable. Cada caja lleva pegada una etiqueta, de un máximo de 30 caracteres, con información útil como el nombre del destinatario, dirección, etc. Implementa la clase Caja con los siguientes métodos:

- Caja (int ancho, int alto, int fondo, Unidad unidad): que construye una caja con las dimensiones especificadas, que pueden encontrarse en «cm» (centímetros) o «m» (metros).
- double getVolumen ( ): que devuelve el volumen de la caja en metros cúbicos.
- void setEtiqueta (String etiqueta): que modifica el valor de la etiqueta de la caja.
- String toString ( ): que devuelve una cadena con la representación de la caja.

8.13. La empresa de mensajería BiciExpress, que reparte en bicicleta, para disminuir el peso que transportan sus empleados solo utiliza cajas de cartón. El volumen de estas se calcula como el 80% del volumen real, con el fin de evitar que se deformen y se rompan. Otra característica de las cajas de cartón es que sus medidas siempre están en centímetros. Por último, la empresa, para controlar costes, necesita saber cuál es la superficie total de cartón utilizado para construir todas las cajas. Escribe la clase CajaCarton heredando de la clase Caja.

8.14. Reimplementa la clase Lista de la Actividad resuelta 7.11, sustituyendo el método mostrar ( ) por el método toString ( ). **Hecho**

8.15. Escribe en la clase Lista un método equals ( ) para compararlas. Dos listas se consideran iguales si tienen los mismos elementos (incluidas las repeticiones) en el mismo orden.

8.16. Diseña la clase Pila heredando de Lista (ver Actividad resuelta 7.13).

8.17. Escribe la clase cola heredando de Lista (ver Actividad final 7.18).

8.18. Diseña la clase ColaDoble, que hereda de cola para enteros, añadiendo los siguientes métodos:

- void encolarPrincipio( ), que encola un elemento al principio de la cola.
- Integer desencolarFinal( ) , que desencola un elemento del final de la cola.

8.19. Un conjunto es un objeto similar a las listas, capaz de guardar valores de un tipo determinado, con la diferencia de que sus elementos no pueden estar repetidos. Escribe la clase Conjunto para enteros heredando de Lista y reimplementando los métodos de inserción para evitar las repeticiones.

8.20. Implementa el método equals ( ) en la clase conjunto. Dos conjuntos se consideran iguales si tienen los mismos elementos, no importa en qué orden.

8.21. Implementa los siguientes métodos:

- static boolean esNumero (Object ob), que nos dice si su parámetro de entrada es de tipo numérico (Integer, Double, Long, Float . . . ).
- boolean sumar (Object a, Object b), que muestra por consola la concatenación de los parámetros de entrada, si ambos son cadenas, o muestra su suma convertida al tipo Double,

si ambos son de tipo numérico. En cualquier otro caso, muestra el mensaje «No sumables».

8.22. La clase Object dispone del método finalize( ) , que se ejecuta justo antes de que el recolector de basura destruya un objeto. Escribe un programa que, mediante la creación masiva de objetos no referenciados y el overriding del método finalize( ), compruebe el funcionamiento del recolector de basura.

8.23. Implementa la clase abstracta Poligono, con los atributos base y altura, de tipo double y el método abstracto double area( ).

8.24. Heredando de Poligono, implementa las clases no abstractas Triangulo y Rectangulo.