

# Radio Station

Se desea implementar un sistema de reproducción de elementos de audio para la automatización de una emisora de radio. En este sistema se dispone de elementos reproducibles (**AudioElement**) que tienen un nombre, se pueden reproducir y que generan un efecto económico según su naturaleza.

Existen diversos tipos de elementos reproducibles: canciones (**Song**), cuñas publicitarias (**Commercial**), audios propios (**OwnAudio**) y secuencias de reproducción (**PlaybackSequence**). La reproducción de las canciones generan un coste, la reproducción de cuñas publicitarias producen un beneficio y los audios propios no generan beneficios ni costes; las secuencias de reproducción son secuencias de elementos reproducibles (**AudioElement**) con un orden de emisión preestablecido. Para implementar esta aplicación se ha ideado el siguiente conjunto de clases:



Clase **AudioElement** abstracta:

- Constructor al que se le pasa el nombre del elemento (String).
- getName: devuelve el nombre (String).
- play: reproduce el elemento.
- duration: devuelve el tiempo (int) en segundos que tarda en reproducirse el elemento
- profit: devuelve el beneficio (int) o coste (en negativo) de reproducirlo
- toString: devuelve el nombre seguido de un espacio y su duración en segundos entre paréntesis.

Clase **Song**: Métodos disponibles, además de los heredados:

- Constructor al que se le pasa el nombre (String), el fichero (String) y el coste de reproducción (int).

Clase **Commercial**: Métodos disponibles, además de los heredados:

- Constructor al que se le pasa el nombre del anuncio (String), el fichero (String) y el beneficio de reproducción (int).

Clase **OwnAudio**: Métodos disponibles, además de los heredados:

- Constructor al que se le pasa el nombre del audio propio (String) y el fichero (String).

Clase `PlaybackSequence`: las secuencias soportan un máximo de 100 elementos reproducibles. Las operaciones de `play`, `duration` y `profit` son consecuentes con la reproducción de los elementos que componen la secuencia. Los elementos de la secuencia están numerados de 0 al número de elementos-1. Los métodos disponibles, además de los heredados son:

- Constructor al que se le pasa el nombre (`String`).
- `add`: añade un nuevo elemento de reproducción al final de la secuencia (no hace nada si no puede añadirlo)
- `set`: se le pasa una posición y un elemento reproducible, sustituyendo el situado en la posición pasada por el nuevo elemento (no hace nada si no puede realizar la operación)
- `remove`: se le pasa una posición y elimina el elemento en dicha posición, desplazando los que se encuentran en posiciones superiores (no hace nada si no puede realizar la operación)
- `insert`: se le pasa una posición y un elemento reproducible, añadiéndolo a la secuencia en la posición pasada y desplazando el que se encontraba en esa posición y superiores. Sólo se admite insertar en posiciones ocupadas y una más de la última (no hace nada si no puede realizar la operación)
- `toString`: devuelve, además del nombre del elemento seguido de un espacio y su duración en segundos entre paréntesis, un espacio y entre llaves y separado por “,”, el resultado de `toString` de cada elemento en la secuencia.

Se dispone de una clase de apoyo ya implementada llamada `AudioTools` que suministra los siguientes métodos estáticos:

- `getDuration`: a la que se le pasa un nombre de fichero de audio (`String`) y devuelve la duración (`int`) en segundos de su reproducción (número aleatorio que representa segundos)
- `play`: a la que se le pasa un nombre de fichero de audio (`String`) y lo “reproduce” (muestra por pantalla “Reproduciendo: <fichero>”)

Crea una clase `RadioStation` y prueba los métodos públicos de las clases además de los propuestos en la siguiente figura.

```

public class RadioStation {
    public static void main(String[] args) {

        PlaybackSequence list40 = new PlaybackSequence("List 40");
        PlaybackSequence salsa = new PlaybackSequence("Salsa");
        PlaybackSequence program = new PlaybackSequence("Program");

        list40.add(new Song("Despacito", "despacito.mp3",10));
        list40.add(new Commercial("Movistar", "movistar.mp3",30));
        list40.add(new Song("Black summer", "black_summer.mp3",8));
        list40.add(new Commercial("Campofrio", "campofrio.mp3",100));
        list40.add(new Song("De perreo", "de_perreo.mp3",7));
        list40.add(new OwnAudio("Noticia local", "noticia_local001.mp3"));

        salsa.add(new Song("Gitana", "gitana.mp3",5));
        salsa.add(new Commercial("Telepizza", "telepizza.mp3",100));
        salsa.add(new OwnAudio("Noticia internacional", "noticia_internacional001.mp3"));
        salsa.add(new Song("Amores como el nuestro", "amores_como_el_nuestro.mp3",5));

        program.add(list40);
        program.add(new Commercial("Coca Cola", "coca_cola.mp3",120));
        program.add(salsa);

        System.out.println(program);
        program.play();
        System.out.println("Beneficio: " + program.beneficio());
    }
}

```