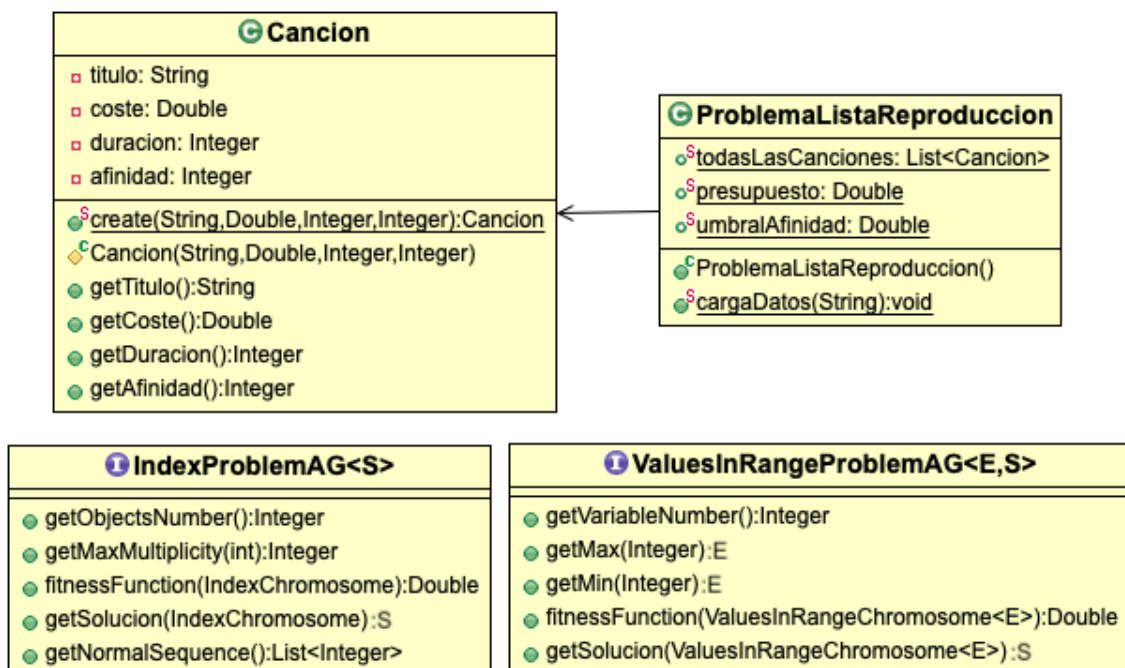


Ejercicio 3

Se tiene una colección de canciones y se desea elaborar una lista de reproducción con un subconjunto de ellas. Cada canción tiene las propiedades: duración (en segundos), afinidad (número entero del 0 al 5) y coste (en euros). Para elaborar la lista de reproducción resultante se cuenta con un presupuesto limitado P, y el objetivo es obtener la lista de mayor duración (en segundos) cuya media de afinidad no baje de un umbral A.

SE PIDE: Resolver el problema mediante **Algoritmos Genéticos**.

- Indique, razonadamente, el tipo de cromosoma seleccionado.
- Realice la implementación de la clase *ProblemaListaReproduccionAG*.



Solución

public class ProblemaListaReproduccionAG **implements**

IndexProblemAG<SolucionListaReproduccion>{

```
    public ProblemaListaReproduccionAG(String fichero) {  
        ProblemaListaReproduccion.cargaDatos(fichero);  
    }
```

```
    @Override
```

```
    public Integer getObjectsNumber() {  
        return ProblemaListaReproduccion.todasLasCanciones.size();  
    }
```

```
    @Override
```

```
    public Double fitnessFunction(IndexChromosome cr) {  
        SolucionListaReproduccion sol = getSolucion(cr);  
  
        Double c = (ProblemaListaReproduccion.presupuesto-sol.getCoste()) >= 0 ?  
            0 : sol.getCoste()-ProblemaListaReproduccion.presupuesto;  
        Double a = sol.getAfinidadMedia() >= ProblemaListaReproduccion.umbraAfinidad ?  
            0 : sol.getAfinidadMedia() -  
                ProblemaListaReproduccion.umbraAfinidad;  
        Integer duracion = sol.getDuracion();  
  
        Double fitness = duracion - 1000000*getObjectsNumber()*(c+a);  
  
        return fitness;  
    }
```

```
    @Override
```

```
    public SolucionListaReproduccion getSolucion(IndexChromosome cr) {  
        List<Integer> ls = cr.decode();  
        return SolucionListaReproduccion.create(ls);  
    }
```

```
}
```