

Programación Frontend y Backend

BLOQUE JAVA

Conceptos introducidos en Java 8



Interfaz

Una interfaz en Java es una colección de métodos y propiedades constantes. En las interfaces se especifica qué se debe hacer pero no su implementación. Serán las clases que implementen estas interfaces las que describan la lógica del comportamiento de los métodos.

Interfaz

```
interface Nave {  
    public moverPosicion (int x, int y);  
    public disparar();  
    .....  
}
```

Implementación

```
public class NaveJugador implements Nave {  
    public void moverPosicion (int x, int y) {  
        //Implementación del método  
        posActualx = posActualx - x;  
        posActualy = posActualy - y;  
    }  
  
    public void disparar() {  
        //Implementación del método  
    }  
  
    ...  
}
```

Interfaz

Imaginemos que hemos definido una clase por cada uno de los animales que aparecen en la siguiente imagen: Oso, Jirafa, Elefante, Pato, etc. y cada clase implementará la interfaz Animal.



¿Qué pasa si queremos añadir un nuevo método en la interfaz?

Interfaz

¡SOLUCIÓN!

En Java 8, toda clase que implemente una interfaz debe declarar los distintos métodos que contenga la interfaz salvo aquellos que estén definidos como métodos por defecto. Estos métodos se caracterizan porque son métodos que están declarados en la propia interfaz y pueden ser utilizados directamente en la clase si nos interesa su comportamiento por defecto.

Esto permite al desarrollador añadir nuevos métodos a las interfaces sin romper las implementaciones existentes de esa interfaz. Al ser ya definido en la interfaz no es necesario que se reimplemente en las implementaciones, aunque si se puede hacer con override.

```
public interface Interfaz1 {  
  
    default void saluda() {  
        System.out.println("Hola soy la interfaz 1");  
    }  
}
```

```
public class Ejercicio1 implements Interfaz1{  
  
    public static void main(String[] args) {  
        new Ejercicio1().saluda();  
    }  
}
```

Interfaz

Ejercicio 1 – Multimedia (es.eoi.multimedia.*) **PARTE I**

Definiremos la interfaz Reproducible.java con los siguientes métodos:

- play()
- stop()

Definiremos la clase Pelicula.java que implementará la interfaz Reproducible

- String titulo
- List<String> actores

Definiremos la clase Cancion.java que implementará la interfaz Reproducible.

- título
- List<String> interpretes

Cuando pulsemos play de un objeto reproducible, aparecerá el siguiente mensaje por consola: “Reproduciendo” y cuando pulsemos stop de un objeto reproducible, aparecerá el siguiente mensaje por consola: “Finalizando Reproducción”

Interfaz

Ejercicio 1 – Multimedia (es.eoi.multimedia.*) **PARTE II**

Por último crearemos la clase Multimedia.java que nos permita probar todo lo anterior, para ello:

- Crearemos una lista de objetos reproducibles, en ella introduciremos 5 películas y 5 canciones distintas.
- A continuación recorreremos la lista e iniciaremos la reproducción de cada objeto, después volveremos a recorrer la lista para detener cada reproducción.
- En el caso de que se reproduzca una película producirá la siguiente salida por consola:
“Reproduciendo titulo (primer actor)”

Lambdas

Entre las múltiples novedades que nos brinda Java 8 encontramos las expresiones lambda, estas expresiones lambda son funciones anónimas, es decir, funciones que no necesitan una clase, el operador lambda (->) separa la declaración de parámetros de la declaración del cuerpo de la función, puede clasificarse de la siguiente manera:

```
public interface IText {  
    public String transform (String x);  
}
```

Interfaz Funcional



```
public static void main(String[] args) {
```

```
    IText IUpper = (x) -> {  
        String retorno = x;  
        retorno = retorno.toUpperCase();  
        return retorno;  
    };
```

```
    IText ILower = (x) -> {  
        String retorno = x;  
        retorno = retorno.toLowerCase();  
        return retorno;  
    };
```

```
    System.out.println(IUpper.transform("Hola Soy Javier"));  
    System.out.println(ILower.transform("Hola Soy Javier"));
```

```
}
```

Lambdas

¿Qué es una interfaz funcional?

Una interfaz funcional es una interfaz con uno y solo un método abstracto. La declaración es exactamente igual que las interfaces normales con dos características adicionales:

- Tiene un único método abstracto, como ya hemos dicho.
- De manera opcional puede estar anotada como `@FunctionalInterface`. (Java.util.function)

Lambdas

Ejercicio 2 – ICalculadora (es.eoi.icalc.*)

Haciendo uso de Lambdas vamos a implementar el funcionamiento de una calculadora básica, para ello crearemos una única interfaz funcional ICalculadoraLambda.java que nos permitirá operar con dos valores enteros.

Por último crearemos la clase CalculadoraLambda.java que nos permita probar todo lo anterior, necesitaremos 4 expresiones lambdas para sumar, restar, multiplicar y dividir los dos números enteros, mostraremos por consola ejemplos de las operaciones anteriores.

Adicionalmente, vamos a mostrar por consola las tablas de multiplicar haciendo uso de nuestra nueva calculadora.

Streams

Biblioteca de clases que nos permitirá realizar distintas operaciones funcionales sobre streams pudiendo hacer uso de las expresiones Lambda.

```
List<String> palabras = new ArrayList<String>();  
palabras.add("Ballena");  
palabras.add("Caballo");  
palabras.add("Jirafa");  
palabras.add("Perro");  
palabras.add("Ratón");
```

```
Stream st;
```

```
st = palabras.stream();  
System.out.println(st.findFirst().get());  
st.close();
```

```
st = palabras.stream();  
System.out.println(  
    st.filter(p -> ((String) p).toLowerCase().contains("o")).count());  
st.close();
```

```
st = palabras.stream();  
st.filter(p -> ((String) p).toLowerCase().contains("o")).forEach(System.out::println);  
st.close();
```



Streams

Ejercicio 3 – Tabla de Resultados (es.eoi.tresult.*)

Haciendo uso de Lambdas y Streams, vamos a utilizar la ICalculadora que hicimos en el ejercicio anterior para almacenar las tablas de multiplicar en una lista de Strings, se almacenarán con el siguiente formato: “1 x 1 = 1”, “1 x 2 = 2” ...

Una vez que las hemos almacenado, se debe mostrar por pantalla los siguientes resultados:

- Número de filas almacenadas: XX
- Número de filas que corresponden a la tabla del 3: XX
- Tabla de Multiplicar del número 5:
 $5 \times 1 = 5$
 $5 \times 2 = 10$
- Tabla de Multiplicar del número 5 (Invertida)
 $1 \times 5 = 5$
 $2 \times 5 = 10$