

## Programación web (curso 2021/2022)

### Práctica 1 – Introducción a Java

*Este documento contiene una propuesta de ejercicios para que el estudiante afiance los conceptos básicos del lenguaje Java. Estos ejercicios se deben realizar de manera individual y no forman parte de la entrega de la primera práctica. No obstante, se recomienda su realización durante la primera semana ya que las clases implementadas servirán de base para las siguientes sesiones y, por tanto, la entrega de la práctica.*

#### **Ejercicio 1: Definición de la clase “Espectáculo”**

Crear una clase Java, de nombre “Espectaculo”, para representar los espectáculos culturales que se pueden programar en un teatro. La clase debe contener los siguientes atributos:

- Título del espectáculo, de tipo *String*.
- Categoría de espectáculo (concierto, obra, monólogo, etc.), utilizando una enumeración.
- Descripción del espectáculo, de tipo *String*.
- Fecha de representación<sup>1</sup>, de tipo *Date*.
- Número de localidades a la venta, de tipo entero.
- Número de localidades vendidas, de tipo entero.

Además, deben definirse los siguientes métodos:

- Constructor vacío (sin parámetros).
- Constructor parametrizado, con tres parámetros: título, categoría y descripción.
- Métodos *get/set* para todos los atributos.
- Método *toString* que imprima toda la información del espectáculo.

Ayuda: Para el manejo de fechas, se recomienda consultar la documentación oficial, en concreto, las clases *Date* y *DateFormat*.

#### **Ejercicio 2: Definición de la clase “Programación”**

Crear una clase Java llamada “Programacion”, que se encargará de almacenar una lista de espectáculos creados. La clase debe contener los siguientes atributos:

- ListaEspectaculos, definido como un *ArrayList* cuyos elementos son objetos de la clase “Espectaculo”.

---

<sup>1</sup> Por simplicidad, se asume una única fecha de representación. Si se desea, puede sustituirse por un vector o lista de fechas para permitir múltiples funciones del espectáculo (por ejemplo, una obra de teatro con varios pases).

A continuación, deberán implementarse los siguientes métodos:

- Constructor vacío, donde se crea una lista de espectáculos vacía.
- Método *programarEspectaculo*, que añada un espectáculo a la programación.
- Método *verEspectaculos*, que devuelva todos los espectáculos que hay guardados en la programación.
- Método *imprimirEventos*, que imprima por consola el contenido de todos los espectáculos de la programación (utilizar el método *toString* de la clase “Espectaculo”).

### Ejercicio 3: Métodos adicionales para la clase “Programación”

Añadir los siguientes métodos a la clase “Programacion”, para realizar otras operaciones y búsquedas de espectáculos:

- Método *verTitulos*, que devuelva una lista con los títulos de los espectáculos que hay registrados en la programación.
- Método *eliminarEspectaculo*, que elimine un espectáculo dado su título. Debe devolver un valor booleano para indicar si se ha borrado con éxito (*true*) o no se ha encontrado el espectáculo (*false*).
- Método *verProximosEspectaculos*, que reciba como parámetro una fecha, y devuelva la lista de los espectáculos programados hasta esa fecha. Debe comprobarse que la fecha introducida es posterior a la actual. En caso contrario, se debe devolver *null*.
- Método *verEspectaculosDisponibles*, que reciba como parámetro una categoría de espectáculo y devuelva los espectáculos de esa categoría para los que aún quedan entradas. Si no hay eventos de esa categoría, se debe devolver una lista vacía.

Ayuda: La fecha actual del sistema se obtiene con *System.currentTimeMillis()*.

### Ejercicio 4: Programa principal

Crear un programa en un fichero aparte para comprobar el funcionamiento de las clases anteriores. Prestar atención a la creación y manejo de fechas, incluyendo la captura de cualquier excepción. El programa debe realizar los siguientes pasos:

1. Crear una programación vacía.
2. Añadir dos espectáculos a la programación, que tengan distinta categoría, utilizando el constructor vacío y los métodos *set*. Incluir un espectáculo con localidades disponibles y otro sin localidades disponibles.
3. Añadir otros dos espectáculos a la programación, utilizando el constructor parametrizado.
4. Invocar al método *verTitulos* e imprimir los títulos de los espectáculos.
5. Invocar al método *verProximosEspectaculos*, primero con una fecha válida y luego con una fecha inválida (anterior a la fecha actual).

6. Invocar al método *verEspectaculosDisponibles* para una categoría de espectáculo considerada en los pasos 2 o 3.
7. Invocar al método *eliminarEspectaculo*, comprobando el resultado obtenido.
8. Invocar al método *imprimirEspectaculo* para ver los espectáculos que quedan en la programación.

Ayuda: Para crear fechas de forma sencilla, consulta el funcionamiento del método *parse* de la clase *SimpleDateFormat*.