



TAREA PARA PROG09: Utilización avanzada de clases Curso 2023-24

ENUNCIADO

- A lo largo de esta unidad has terminado de familiarizarte con el resto de conceptos relacionados con la **Programación Orientada a Objetos** que faltaban por ver de una manera más formal y con ejemplos explícitos: **composición; herencia; clases y métodos abstractos; sobrescritura de métodos; interfaces; polimorfismo; ligadura dinámica**, etc.
- Has experimentando con todos estos conceptos y los has utilizado en pequeñas aplicaciones para comprobar su funcionamiento y su utilidad.
- Una vez finalizada la unidad se puede decir que tienes ya un dominio adecuado del lenguaje Java como un lenguaje que permite aplicar todas las posibilidades de la **Programación Orientada a Objetos**. Dado ese supuesto, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunas de las construcciones que has aprendido a utilizar.
- Se trata de desarrollar un proyecto NetBeans en Java :
Tarea_Prog09_Multimedia_NombreApellido1Apellido2: donde almacenaremos las clases necesarias de una aplicación de manejo de polígonos regulares
- Habrá que realizar un documento **PDF** llamado: **Tarea_Prog09_NombreApellido1Apellido2.pdf**, en el que se pondrá una cabecera con el nº de la tarea, vuestro nombre y el nombre del módulo . Se adjuntarán las **capturas de pantallas** en la que aparezca como **fondo** vuestra conexión **al aula virtual**, los resultados de la ejecución del ejercicio (tal y como se vería con Netbeans), para así mostrar que funcionan (debe aparecer una prueba ejecutada del ejercicio).
- **La tarea no se dará por entregada si el proyecto tiene errores de sintaxis, no compila, no se ejecuta correctamente, si lo entregado no se corresponde con lo pedido o si no se ha realizado un mínimo de la tarea, es decir, si está vacía o casi.**
- En el proyecto **Tarea_Prog09_Multimedia_NombreApellido1Apellido2**, crearemos los siguientes paquetes:
 - **modelos**
 - **controlador**
 - **app**
- Dentro del paquete **modelos** realizaremos las clases:
Formato, Anuncio , Multimedia, Pelicula, CapituloSerieInfantil
- Dentro del paquete **controlador** realizaremos la clase **Utilidades**.
- Dentro del paquete **app** realizaremos la clase **TVApp**

Clase Utilidades:

- En esta clase realizaremos los siguientes métodos estáticos:
 - public static void **visualizaMensaje**(String mensaje):
 - public static int **leerOpcion**(int limite)
Permite leer una opción entre 1 y limite hasta que sea correcta.
 - public static void **pulsarTecla**()
Pulsa una Tecla para seguir

- public static String **leerString**(String mensaje)
lee por teclado un String
- public static int **leerEntero**(String mensaje)
lee por teclado un entero hasta que sea correcto
- public static float **leerFloat**(String mensaje)
lee por teclado un float hasta que sea correcto

Clase Formato:

- La clase **Formato** no hereda de ninguna clase y los objetos tendrán la siguiente información
 - **Atributos (privados):**
 - **codigoFormato**: entero con el código de formato.
 - **arrayFormato**: array de String con el nombre de los formatos. Los posibles valores del array será uno de los siguientes: wav, mp3, midi, mp4, avi, mov, mpg, cdAudio y dvd. El codigoFormato será el índice al array arrayFormato.
 - **Métodos (públicos):**
 - **constructores**:
 - uno asignándole al codigoFormato a 0
 - otro asignándole al codigoFormato un valor que se le pase como parámetro.
 - **getNombre**: que devuelva el nombre del formato del objeto, devuelve un String con el nombre del formato del códigoFormato del objeto de arrayFormato[codigoFormato].
 - **getCodigoFormato** y **setCodigoFormato**

Clase Anuncio

- La clase **Anuncio** no hereda de ninguna clase, representa el anuncio de un producto.
 - **Atributos (privados):**
 - **duración**
 - **nombreProducto**
 - **mensajeAnuncio**
 - **Métodos (públicos):**
 - **Constructores**:
 - Uno con los valores a nulo y cero
 - Otro con todos los atributos.
 - Implementar los **getters y setters** de todos los atributos (duración, nombreProducto y mensajeAnuncio)
 - Sobreescribe el método **toString**: devuelve concatenado el nombre del producto y la duración
 - Sobreescribe el método **equals**: devuelve true cuando el nombre y la duración del producto sean iguales y false cuando no lo son.
 - No es necesario crear ningún método más

Clase Multimedia

- La clase **Multimedia** se utiliza para poder crear objetos multimedia (películas, canciones, cuentos, documentales, ...).
- Los objetos de esta clase tendrán la siguiente información:
 - **Atributos (privados):**
 - **título, autor y duración (en segundos).**
 - **formato:** será un objeto de la **clase Formato**
 - **Métodos (públicos):**
 - **Constructores:**
 - uno con todos los valores a nulos y ceros
 - otro pasándole todos los valores.
 - Implementar los **getters y setters** de todos los atributos (título, autor, duración y formato)
 - Implementa el método **reproducir**, que reproducirá los segundos de objeto multimedia.
La implementación de este método solo escribirá en consola la secuencia de segundos transcurridos hasta que finalice, según la duración.
Por ejemplo, si dura 70 segundos (1 minuto 10 segundos), escribirá:
00:00:01—00:00:02—00:00:03—00:00:04 --...-00:00:59—00:01:00 ... 00:01:10 [Fin]
Cuando llegue a 60 segundos, la hora será una mas y comenzará a contar de nuevo los minutos.
 - Sobreescribe el método **toString**. Devolverá una cadena con la descripción del objeto: concatenando: el título, el autor, la duración y el formato.
 - Sobreescribe método **equals**: dos objetos multimedia son iguales cuando el titulo, el autor y la duración son iguales.

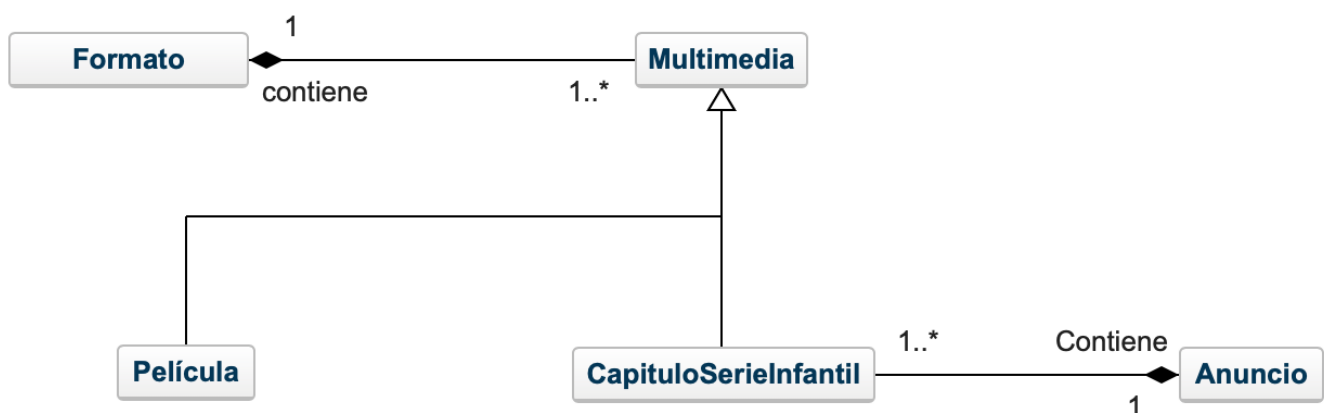
Clase Película

- La clase **Película** hereda de la clase **Multimedia**.
- La clase **Película** tiene, además de los atributos heredados sus propios atributos y métodos:
 - **Atributos (privados):**
 - **actorPrincipal**
 - **actrizPrincipal**
 - **Métodos (públicos):**
 - **Constructores:**
 - uno con todos los valores a nulo y ceros
 - otro constructor pándole como entrada: un objeto Multimedia, actorPrincipal y actrizPrincipal.
 - Implementar los **getters y setters** de todos los atributos (actorPrincipal y actrizPrincipal)
 - Sobreescribe el método **toString** para que devuelva, además de la información general (heredada), la nueva información (actor y actriz principal). Utiliza una llamada al toString de la clase padre.
 - Piensa si deberías sobrecribir el método **equals** en esta clase.

Clase CapituloSerieInfantil

- La clase **CapituloSerieInfantil** hereda de la clase Multimedia, representa un capítulo de una serie de dibujos animados, compuesta de varios capítulos.
- Un capítulo tendrá siempre un **anuncio** que se emitirá justo a mitad del capítulo (ver clase anuncio).
- La clase **CapituloSerieInfantil** tiene, además de los atributos heredados sus propios atributos y métodos:
 - **Atributos:**
 - **numeroCapítulo**
 - **canalTV** : canal donde está disponible el capítulo de la serie
 - **anuncio**: objeto de la **clase Anuncio**.
 - **Métodos:**
 - **Constructores:**
Uno con todos los valores a nulo o cero
Otro pasándole como parámetro: un objeto de tipo Multimedia, el nº de capítulo, el canal de TV y un **objeto de la clase Anuncio**.
 - Método **reproducir** un capítulo de serie infantil es diferente a reproducir una película. Durante la reproducción del capítulo, se visualizará el mensaje del anuncio. Implementa este método de modo que escriba en consola la secuencia de segundos transcurridos del capítulo, y el corte del anuncio con el mensaje del anuncio.
Por ejemplo, si el capítulo dura 2 minutos, el anuncio se verá en el minuto 6. Por tanto, escribirá en consola:
00:00:01--00:00:02-- ... --00:00:59
<Anuncio: "**MENSAJE**">
00:01:00--00:01:01--00:01:02-- ... --00:01:59
[Fin]
- Sobreescribe el método **toString** de igual manera que en la clase Película.
- Piensa si debes sobrecribir el método **equals**.

Esquema en el que se muestran las principales relaciones entre las distintas clases utilizadas:



Aplicación TVApp

- Escribe una clase ejecutable en la que tendremos un **ArrayList de objetos multimedia** y mediante un menú podamos introducir objetos en ese array:
 1. **Introducir Películas:** llamando al método **introducirPelículas**
 2. **Introducir Capítulo serie infantil:** llamando al método **introducirCapitulos**
 3. **Emitir Todos**
 4. **Comparar elementos**
 5. **Fin**
- Una vez introducidos todos los objetos multimedia que deseemos podremos lo siguiente:
 1. Realizar una llamada al método **emitirTodos** que emitirá todos los objetos del array de objetos multimedia.
 2. Comparar entre sí dos objetos dándole como entrada al método **comparar** los elementos del array que se deseen comparar, introduciendo por teclado la posición de los dos elementos que deseemos comparar .
- **Método: emitirTodos** que reciba como parámetro el ArrayList de objetos Multimedia y los reproduzca en el orden en el que se han introducido.

Este método, debe visualizar en consola la siguiente información:

 - título, autor, duración y formato
 - si es una película, el actor y actriz principal.
 - si es un capítulo de una serie: el nº de capítulo y el canal de TV
 - Una vez visualizada la información anterior, se reproducirá el objeto.
- **Método: comparar** que dándole como entrada dos objetos multimedia compare ambos objetos, si son iguales visualizará un mensaje "OBJETOS SI SON IGUALES" y "OBJETOS NO SON IGUALES".
- **Método: introducirPelículas :**

Realizaremos lo siguiente:

 - Crear un método **introducirMultimedia**, que lo utilizaremos tanto en películas como en capítulos. En este método, introduciremos por teclado: **título, autor, duración (en segundos), formato:** introduciremos el código de formato (visualizaremos los códigos y el nombre de cada formato antes de teclear, para saber cuáles son). Este método devolverá un objeto de tipo **Multimedia**.
 - Introducir por teclado los atributos propios de películas: **actor principal y actriz principal**.
 - Introduciremos películas hasta que deseemos.
- **Método: introducirCapitulos:**

Realizaremos lo siguiente:

 - Llamar al método **introducirMultimedia**, que nos devolverá un objeto Multimedia.
 - Introducir por teclado los atributos propios del capítulo: **numeroCapítulo, canalTV y anuncio: duración, nombreProducto y mensajeAnuncio**.
 - Introduciremos capítulos hasta que deseemos.

CRITERIOS DE PUNTUACIÓN. TOTAL 10 PUNTOS.

- Aplicación **Multimedia**: 10 puntos

QUÉ HAY QUE ENTREGAR

- Se deberá entregar lo siguiente:
 - **Proyecto NetBeans: Tarea_Prog09_Multimedia_NombreApellido1Apellido2**

- Fichero **PDF** con la descripción de todos los métodos, las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea y las capturas de ejecución de ambas aplicaciones, teniendo como fondo vuestra conexión al Papas:

Tarea_Prog09_NombreApellido1Apellido2.pdf.

- Se los dos archivos se deben **comprimir en uno solo:**
Tarea_Prog09_NombreApellido1Apellido2.rar
- En caso de que haya que realizar un 2º envío el fichero comprimido se llamará:
Tarea_Prog09_NombreApellido1Apellido2_ENVIO2.rar

RECURSOS NECESARIOS PARA REALIZAR LA TAREA.

- Ordenador personal.
- JDK y JRE de Java SE.
- Entorno de desarrollo NetBeans .
- Ejercicios resueltos subidos por el profesor: **RECURSOS APORTADOS POR EL PROFESOR: PROG09. EJEMPLOS-ENUNCIADOS.**