Batcheador

FFmpeg es una colección de software libre que puede grabar, convertir y hacer streaming de audio y vídeo. FFmpeg suele usarse desde la consola, y tiene cientos de flags para realizar miles de operaciones distintas. El programa es extremadamente potente, pero es muy difícil recordar todas las flags, los tipos de parámetros, el formato de los parámetros, etc.

Por ejemplo, se puede cortar un video con FFmpeg usando el siguiente comando:

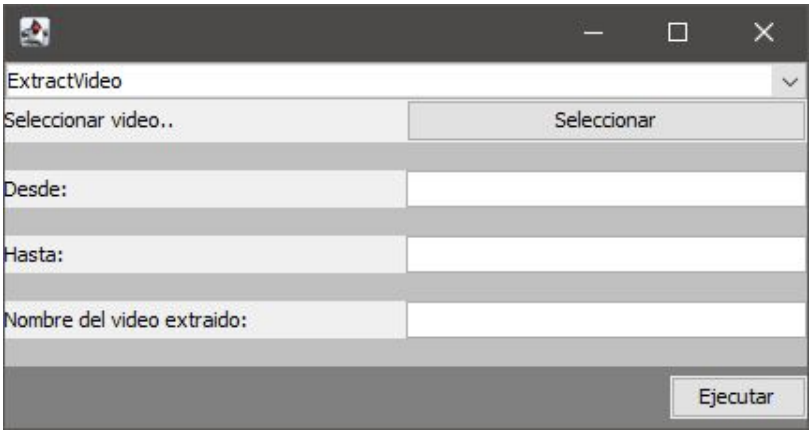
```
ffmpeg -i movie.mp4 -ss 00:00:03 -t 00:00:08 cut.mp4
```

Donde el flag `-i` indica el archivo de entrada, `-ss` el tiempo inicial y `-t` el tiempo final.

Nosotros queremos desarrollar un batcheador, una biblioteca que permite a cualquier programador de Java rápidamente crear interfaces gráficas para realizar comandos preestablecidos con FFmpeg. Queremos que el usuario escriba una clase así:

```java
@Comando()
public class ExtractVideo {
    @Parametro(parametro = "-i", tipo = Parametro.Type.FILEPICKER, nombreCampo = "Seleccionar video..")
    private String pathVideo;
    @Parametro(parametro = "-ss", tipo = Parametro.Type.NUMERIC, nombreCampo = "Desde:")
    private String inicio;
    @Parametro(parametro = "-to", tipo = Parametro.Type.NUMERIC, nombreCampo = "Hasta:")
    private String fin;
    @Parametro(tipo = Parametro.Type.TEXTFIELD, nombreCampo = "Nombre del video extraido:")
    private String nombreNuevoVideo;
}
```

Para generar una interfaz de usuario, con campos y botones, para ejecutar el comando de forma sencilla. Por ejemplo:



Entregas

Entrega 1 2020-09-09

Configuración

- Instalar IDE y proyecto.
- Instalar FFmpeg en su computadora, y agregarlo al path

Requerimientos

- Cuando inicia, el programa debe abrir una ventana de Swing.
- La ventana de Swing debe tener:
    - Un JComboBox (con cualquier opción, para esta entrega no importa).
    - Un JTextField, el cual imprime sus contenidos por consola cada vez que cambia.
    - Un JButton, que al ser presionado genere un JFileChooser.

Entrega 2 2020-09-30

Requerimientos

- Debe existir una clase (batcheador) que se instancia al inicio del programa y pertenece a la ventana de Swing principal.
    - Esta clase recibe en su constructor una lista de Class. Hasta la entrega 4 esta lista se puede generar a mano, es decir, puede estar hard-codeada.
    - Estas clases sólo contienen campos y anotaciones. No extienden o implementan ninguna otra clase o interfaz.
- Debe existir una anotación que permita indicar qué campos generar. Debe incluir:
    - Alguna forma de discernir el tipo del campo (numérico, texto, archivo).
    - Por ahora los campos numéricos son simples TextField.
    - Una descripción o label.
- La lista de Class que recibe el batcheador son Comandos, cuyos campos usan anotaciones para definir qué debe verse en la ventana de Swing.
- La ventana de Swing debe, de alguna forma, pedirle al batcheador los campos de la primera de sus Class y renderizarlos en pantalla.

Entrega 3 2020-10-21

Requerimientos

- La ventana principal de Swing debe tener un ComboBox adicional que permite elegir la pantalla a mostrar. Cada Class del batcheador representa un comando y una pantalla distinta.
- La ventana principal de Swing debe tener un JButton adicional (confirmar) que al ser presionado imprima por consola lo siguiente:
    - Nombre del comando actual.
    - Por cada campo, el nombre del campo y su valor actual.
- Se agrega un nuevo tipo de campo: un ComboBox con una lista de opciones.
    - Como ejemplo, algunos posibles códecs de audio y video son:
    - AUDIO: flac, mp3, wma, aac, libvo_aacenc, copy, amr.
    - VIDEO: libx264, copy, mpeg4, flv, wmv1, libxvid.
- Se agrega un nuevo tipo de campo: un Label y un Button que juntos representan un FilePicker. Al hacer click en el botón se abre un JFileChooser, y al seleccionar un archivo, el path de este queda guardado en el campo.

Entrega 4 2020-11-11

Requerimientos

- Debe agregarse una anotación para indicar aquellas clases que son comandos.
- Debe usarse una biblioteca de terceros para encontrar todas las clases con la anotación y cargarlas al batcheador automáticamente, al inicio del programa, sin pasar ninguna lista a mano. (en el futuro habrá un instructivo sobre esta biblioteca)
- El botón de confirmación debe estar deshabilitado si existen campos vacíos o campos con contenido inválido.
- Al oprimir el botón de confirmación, debe ejecutarse el comando de FFmpeg.
- Debe detectar y notificar la existencia de un error de FFmpeg (ver códigos de error). No hace falta manejarlo ni parsarlo, con mostrar la salida de FFmpeg al usuario es suficiente.
- Deben estar implementados los tres comandos del ejemplo al final del documento.

Repechaje 2020-25-11

- Fecha final de entregas.
- Cualquier corrección pendiente tiene que ser entregada.

Condiciones de aprobación

1. El grupo presentó todas las entregas en tiempo y forma.
2. Las entregas cumplen con la funcionalidad pedida.
3. Si alguna funcionalidad no estuviera presente o tuviera algún error, la entrega fue corregida posteriormente.

Si cumplen estas condiciones, el trabajo práctico estará aprobado. Si existe algún inconveniente para realizar una entrega, es preferible que lo hablen conmigo antes de realizar una entrega incompleta o errónea.

Formato de entrega

Las entregas se realizan mediante un release en el repositorio de GitHub asignado a su grupo. La versión del release debe seguir el siguiente formato: `ve.c`, donde e es el número de entrega y c es el número de corrección. Por ejemplo:

- Primer entrega: v1.0
- Corrección de la primer entrega: v1.1
- Segunda corrección de la tercer entrega: v4.2

El release debe ser creado antes de las 23:59 del día de la entrega.

Ejemplos Comandos FFmpeg

MuteVideo

```
ffmpeg -i video.mp4 -c copy -an muteado.mp4
```

CutVideo

```
ffmpeg -i video.mp4 -ss segundosInicio -t segundosFin cut.mp4
```

AudioToVideo

```
ffmpeg -loop 1 -i img.png -i audio.mp3 -c:a codecAudio -c:v codecVideo
-shortest salida.mp4
```

En cada comando el texto subrayado indica cuales son los campos que el usuario rellena en la interfaz. Por ejemplo, en AudioToVideo el usuario debe poder elegir la ubicación del video de entrada, del audio de entrada, el codec de video y audio y el nombre del archivo de salida. Cada uno de estos debe ser un campo en la clase correspondiente al comando, y debe tener la anotación correspondiente.

Links

Organización GitHub
https://github.com/patrones-2020-2c

Descarga JDK 11
https://www.oracle.com/java/technologies/javase-jdk11-downloads.html

Descarga git
https://git-scm.com/downloads

Descarga Eclipse
https://www.eclipse.org/downloads/

Descarga IntelliJ
https://www.jetbrains.com/idea/download

Instalar FFmpeg
https://www.wikihow.com/Install-FFmpeg-on-Windows