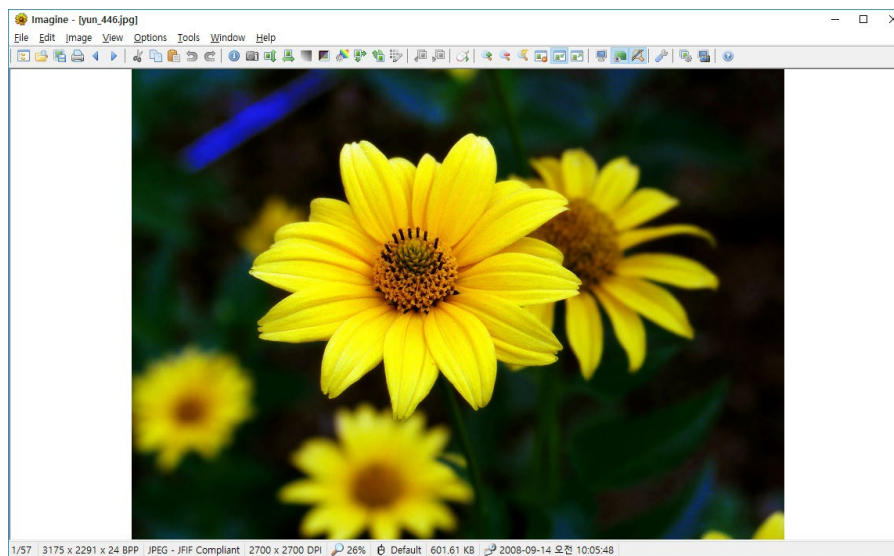
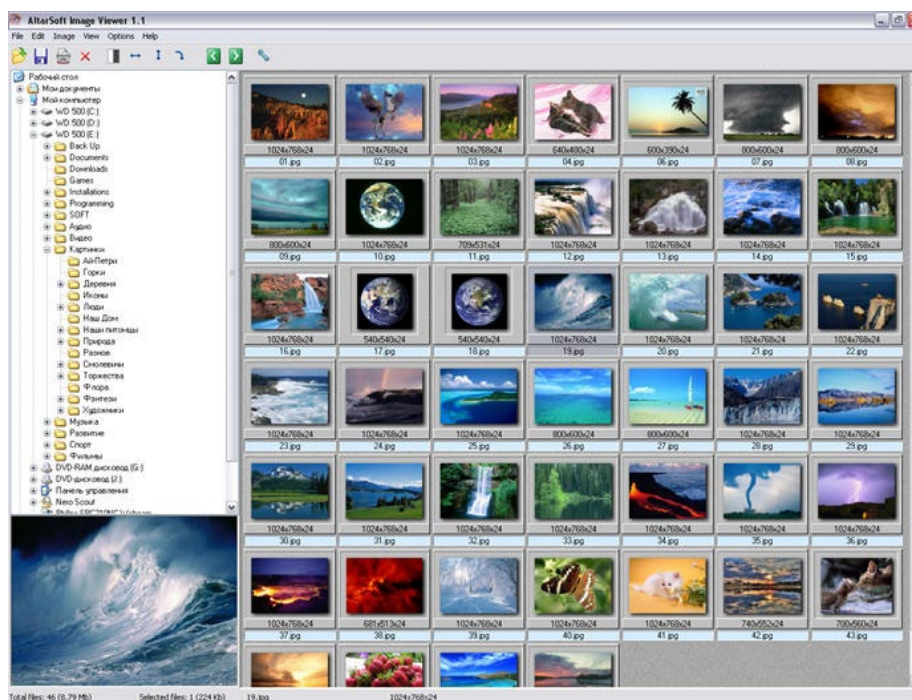


Práctica Individual: Biblioteca de Imágenes

El objetivo de la práctica es realizar un programa en Java que sirva como biblioteca de imágenes, mostrando datos básicos de un conjunto de imágenes y permitiendo visualizar las imágenes. El conjunto de funcionalidades a implementar será un subconjunto de las de las bibliotecas de imágenes existentes, que permita mostrar los conocimientos adquiridos a lo largo de las diferentes unidades de la asignatura.



Imagine: <https://www.nyam.pe.kr/dev/imagine/>



Altarsoft Image Viewer: http://www.altarsoft.com/altarsoft_image_viewer.shtml



Requisitos:

- 1. Crear colección de imágenes (~3 puntos):** Tu proyecto debe ser capaz de generar una serie de carpetas que contengan imágenes, que simulen la situación de partida, una colección de imágenes desordenada que permita evaluar tu proyecto de forma óptima. Para ello:
 - 1.1. (Folder structure)** Debes tener una función que genere una jerarquía de carpetas aleatoria, con diferentes niveles de profundidad y anchura. Esto debe simular el paso de los años acumulando imágenes sin ningún tipo de orden. La función debería poder configurarse en parámetros como por ejemplo (número de carpetas máximo por nivel, número de niveles máximo, nombres de las carpetas, etc).
 - 1.2. (Create images)** Esta función se encargará de "crear imágenes" explotando la librería de [Graphics2D](#) de Java. Esta librería permite crear imágenes (o dibujos más bien) a partir de primitivas (dibuja una línea, dibuja un círculo, dibuja un cuadrado). También tiene la capacidad de dibujar una imagen completa o un fragmento de una imagen, por lo que podréis crear composiciones a partir de otras imágenes (reales). Esto permitirá que las imágenes creadas sean diferentes y se pueden llegar a hacer cosas bastante creativas. El objetivo debería ser que la función sea capaz de crear una imagen diferente cada vez que es llamada, por lo que debe poder configurarse de alguna manera (número de primitivas a utilizar, colores a utilizar, tipo de imagen a crear, dimensiones, extensión, etc). El resultado final será la creación de un archivo de imagen en un path determinado.
 - 1.3. (Edit metadata)** Esta función se encargará de manipular los metadatos de una imagen utilizando la librería [Apache commons-imaging](#) para poblarlos con valores interesantes que nos permitan después realizar análisis sobre la biblioteca de imágenes. Modifica parámetros como la fecha de captura, la posición GPS, etc.
- 2. Análisis de imágenes (~3 puntos):** Tu proyecto debe ser capaz de realizar un análisis de las imágenes contenidas en las carpetas a partir de un path dado, recolectando información de las diferentes imágenes:
 - 2.1. (Formato)** Diseña una clase/clases adecuadas para almacenar la información relevante de las imágenes. Decide que información quieres guardar de cada imagen, y en que colección/colecciones guardas las imágenes.
 - 2.2. (Analizador)** Haz un algoritmo que recorra la carpeta dada (y sus carpetas anidadas) recopilando los datos de cada imagen y almacenándolos adecuadamente.
 - 2.3. (Ordenaciones):** Implementa funciones para ordenar tu colección de imágenes por el criterio dado (por ejemplo, por fecha ascendente/descendente; por ancho de imagen, etc).



- 2.4. (Filtros):** Implementa funciones para filtrar tu colección por el criterio dado, (por ejemplo, imágenes de 500 pixels de ancho o más; imágenes creadas en una fecha concreta, etc.).
- 3. Graphical User Interface (~3 puntos)** Tu proyecto debe mostrar las imágenes y la información de estas mediante un interfaz gráfico. El interfaz también permitirá acceder a las distintas funcionalidades implementadas (en los puntos anteriores):
- 3.1. (TableView):** que muestre los datos de cada una de las imágenes, utilizando columnas para cada uno de los campos y exponiendo la funcionalidad de ordenación (con los nombres de las columnas).
 - 3.2. (ImageView):** que muestre la imagen seleccionada en cada momento, respetando adecuadamente los distintos tamaños y ratios de imagen.
 - 3.3. (Menús y diálogos):** que permitan acceder al resto de funcionalidad (como creación de carpetas e imágenes, análisis, filtros, etc).
- 4. Otros (~1 puntos)** Tu proyecto debe ser único, aquí puedes incluir cualquier funcionalidad que encaje con la temática, algunas ideas:
- 4.1. Sincronización de la información hacia/desde base de datos.
 - 4.2. Edición de imágenes (recortar, redimensionar, cambios de color/brillo).
 - 4.3. Miniaturas / carrousel de imágenes / matriz de imagenes.
 - 4.4. Operaciones en masa (renombrar / modificar metadatos / cambio de formatos / añadir marcos).
 - 4.5. Creación de imágenes desde el interfaz (un Paint).
 - 4.6. Análisis Multihilo / batch processing de las imágenes.



Criterios de Evaluación:

- Las prácticas son individuales, la línea entre compartir ideas / compartir soluciones / compartir código / copiar puede ser delgada en ocasiones. Ante la duda documentar siempre la fuente de las partes de código que incorporéis al proyecto.
- Los puntos indicados en el documento son estimaciones (ya que los elementos no son totalmente independientes).
- El enunciado de la práctica es deliberadamente abierto para que cada alumno pueda orientarla hacia lo que más le interese (y todas las prácticas sean lo suficientemente diferentes).
- Se creará un foro en la PDU para las clarificaciones o dudas respecto a la práctica.
- No hay puntos extra por cosas que no se detallen en el documento y que no sean previamente "aprobadas" por el profesor, si tenéis sugerencias hacérselas llegar (email o en clase).
- El criterio de evaluación principal es el cumplimiento de los requisitos, pero además se aplicará un multiplicador de 0 a 1 teniendo en cuenta:
 - Uso adecuado de la programación orientada objetos y de las librerías de Java.
 - Calidad del interfaz gráfico generado, atendiendo a criterios de usabilidad y ergonomía.
 - Documentación del código (Javadoc / tests)
 - Calidad del código, incluyendo estructura, decisiones de diseño, comentarios, nombres de clases/variables y métodos, etc.
 - Presentación de la práctica y respuesta a las preguntas.

Entrega:

Presentación: Como parte de la entrega del proyecto debéis presentarlo de forma adecuada (diapositivas + demo) explicando las decisiones tomadas, el funcionamiento de la práctica, como se ha llegado a ese diseño y cualquier cosa que queráis destacar. Asimismo, tendréis que contestar a preguntas relacionadas sobre el proyecto y su funcionamiento. 15 min. diapos+demo; 10 min preguntas.

Formato: Proyecto de Java con el código fuente comprimido + archivo.jar listo para lanzar el proyecto (+ instrucciones de uso si no es evidente como manipularlo).

Deadline: La presentación se deberá realizar antes de que concluya el periodo de evaluación de Junio. El código deberá entregarse con al menos 24 horas de antelación a la presentación. Cuando se acerquen las fechas el profesor publicará un calendario para las presentaciones.