Image Viewer and Editor

Overview

The Image Viewer and Editor is a Java-based desktop application that provides a user-friendly graphical interface for viewing and manipulating digital images. It leverages the powerful Java Swing library to create a visually appealing and interactive experience for users. The application supports a wide range of image formats and offers various image manipulation capabilities, including resizing, grayscale conversion, and brightness adjustment.

Features

Image Handling

- **Image Selection**: Users can select image files from their local file system using a file chooser dialog. The application supports a wide range of image formats, including but not limited to JPEG, PNG, BMP, and GIF.
- **Image Viewing**: The selected image can be displayed in its original size and resolution within the application window, allowing for a clear and detailed view.

Image Manipulation

- **Resizing**: The application provides a convenient interface for resizing images. Users can specify new width and height values, and the application will generate a resized version of the image using an interpolation algorithm (SCALE_SMOOTH) to ensure smooth and high-quality results.
- **Grayscale Conversion**: Users can convert the selected image to grayscale with a single button click. This feature can be useful for various purposes, such as creating black-and-white versions of images or preparing them for further processing or analysis.
- **Brightness Adjustment**: The application allows users to adjust the brightness of an image by specifying a brightness factor value. Positive values increase the brightness, while negative values decrease it.

User Interface

- **Separate Windows**: Resized and brightness-adjusted images are displayed in separate windows, allowing users to easily compare the original and modified versions.
- **Back Button**: When viewing a resized or brightness-adjusted image in a separate window, a back button is provided to return to the main panel seamlessly.
- **Close Button**: A dedicated "Close" button allows users to exit the application gracefully.

Implementation Details

Architecture

The application follows an object-oriented design approach and is structured into classes and methods for better organization and extensibility. The main class, `ImgViewer`, extends `JFrame` and implements the `ActionListener` interface to handle user interactions.

Key Classes and Methods

- `ImgViewer`: The main class that represents the application window and handles user interactions.
    - `resizePanel()`: Creates a panel for resizing the image and prompts the user to enter new width and height values.
    - `showResizeImage(int height, int width)`: Displays the resized image in a separate window.
    - `grayScaleImage()`: Converts the selected image to grayscale and displays it in the main panel.
    - `brightnessPanel()`: Creates a panel for adjusting the brightness of the image and prompts the user to enter a brightness factor value.
    - `showBrightnessImage(float factor)`: Displays the brightness-adjusted image in a separate window.
    - `chooseFileImage()`: Allows the user to select an image file from the local file system.
    - `actionPerformed(ActionEvent e)`: Handles user interactions with buttons and performs the corresponding actions.

Technologies and Libraries

- **Java Development Kit (JDK)**: The application is written in Java and requires a compatible JDK version (8 or later) to run.
- **Java Swing**: The application utilizes the Java Swing library for creating the graphical user interface (GUI) components and handling user interactions.

Usage

1. **Prerequisites**: Ensure that you have a compatible Java Development Kit (JDK) installed on your system.
2. **Running the Application**: Compile and run the `ImgViewer` class to launch the application.
3. **Image Selection**: Click the "Select File" button to open a file chooser dialog. Navigate to the desired image file and select it to load it into the application.
4. **Image Viewing**: Once an image is loaded, click the "Show Image" button to view the original image in the main panel.

5. **Resizing**: To resize the image, follow these steps:
    - Click the "Resize" button to open the resize panel.
    - Enter the desired new width and height values in the respective text fields.
    - Click the "Show Resized Image" button.
    - A new window will open, displaying the resized image.
    - Use the back button in the resized image window to return to the main panel.
6. **Grayscale Conversion**: To convert the image to grayscale, click the "Grayscale" button. The grayscale version of the image will be displayed in the main panel.
7. **Brightness Adjustment**: To adjust the brightness of the image, follow these steps:
    - Click the "Brightness" button to open the brightness adjustment panel.
    - Enter the desired brightness factor value in the text field. Positive values increase brightness, and negative values decrease it.
    - Click the "Show Brightness Image" button.
    - A new window will open, displaying the brightness-adjusted image.
    - Use the back button in the brightness-adjusted image window to return to the main panel.
8. **Closing the Application**: To exit the application, click the "Close" button.

Error Handling

The application implements error handling using `JOptionPane` messages for the following cases:

- Attempting to perform an operation (resize, brightness adjustment) without selecting an image first.
- Entering invalid or missing values for width, height, or brightness factor when prompted.

Extensibility

The codebase is designed with modularity and extensibility in mind. Developers can add new features, such as additional image manipulation operations or support for more image formats, by extending the existing classes or creating new ones. The application follows best practices in object-oriented programming, making it easier to maintain and enhance the codebase.

Contributing

Contributions to this project are welcome. If you encounter any issues or have suggestions for improvements, please open an issue or submit a pull request on the project's repository. When contributing, please follow the established coding standards and guidelines.