JAVA ZIPPER

Introduction

The Java Zip Utility is a simple desktop application built with vanilla Java and Swing. It allows users to zip and unzip files using Java's built-in ZipInputStream and ZipOutputStream classes. The tool is lightweight, cross-platform, and designed for ease of use with a clean GUI.

Abstract

This project demonstrates how core Java libraries can be used to create a functional ZIP file manager. Users can compress multiple files into a single archive or extract contents from existing ZIP files. The GUI simplifies interaction, making the tool practical for everyday use and educational purposes.

Tools Used

Programming Language:
Java (JDK 8 or higher)

GUI Framework:
Java Swing – used for building the graphical user interface with components like JFrame, JPanel, JButton, JFileChooser, and JLabel.

Compression API:
ZipOutputStream – used to write files into a ZIP archive.
ZipEntry – represents each file entry in the ZIP archive.

Extraction API:
ZipInputStream – used to read and extract files from a ZIP archive.

File Handling:
File, FileInputStream, FileOutputStream – used for reading and writing file data during compression and extraction.

Multithreading (for GUI responsiveness):
SwingWorker – used to perform long-running tasks like zipping/unzipping in the background without freezing the GUI.

Steps Involved in Building the Project

Step 1: Setting Up the GUI

- Created a main JFrame with buttons for "Zip Files" and "Unzip Files".

- Used JFileChooser to allow users to select files and directories.
- Added labels and dialogs for user feedback and status updates.

Step 2: Implementing Zipping Functionality

- Used ZipOutputStream to create a new ZIP archive.
- Iterated through selected files and added each as a ZipEntry.
- Buffered file reads and writes to optimize performance.

Step 3: Implementing Unzipping Functionality

- Used ZipInputStream to read entries from a selected ZIP file.
- Created output files and directories based on entry names.
- Buffered extraction to ensure smooth performance.

Step 4: Handling Errors and Edge Cases

- Added checks for file overwrite conflicts.
- Implemented try-catch blocks for I/O exceptions.
- Used SwingWorker to keep the GUI responsive during long operations.

Step 5: Testing and Optimization

- Verified functionality across different operating systems.
- Tested with large files and nested folders.
- Refined GUI layout and added tooltips for clarity

Conclusion

The Java Zip Utility is a reliable and user-friendly tool for basic file compression tasks. Built entirely in Java, it showcases how standard libraries can be used to develop cross-platform desktop applications. It serves as a solid foundation for future enhancements like drag-and-drop support or password protection.