# Project Title: Simple Stopwatch & Countdown Timer

## Overview:

This Java desktop application allows users to use either a stopwatch or a countdown timer with a simple GUI. Users can start, pause, and reset timers easily. The application demonstrates core OOP principles such as inheritance, polymorphism, multithreading, and exception handling, alongside a graphical user interface. It also features a dynamic UI that hides the countdown input field in stopwatch mode and shows it only in countdown mode for better user experience.

## OOP Features:

**GUI:** Implemented using Java Swing components (JFrame, JPanel, JButton, JLabel, JTextField). The countdown input field is dynamically shown or hidden depending on the active mode (stopwatch or countdown).

**Inheritance:** An abstract class AbstractTimer defines shared attributes and methods for timers. Stopwatch and CountdownTimer extend it and provide specific behavior.

**Polymorphism:** The start(), pause(), and reset() methods are overridden in child classes. The GUI interacts with timers via the abstract parent type, demonstrating runtime polymorphism.

**Multithreading:** The timer updates run on a separate thread using Thread and Runnable, preventing the GUI from freezing during long operations.

**Exception Handling:** Handles invalid countdown values by showing an error dialog, and uses try-catch for InterruptedException when stopping the timer thread.

## Technologies Used:

Java 17, Swing

## Conclusion:

The project is small, self-contained, and demonstrates multiple OOP concepts along with a functional, interactive GUI. It also incorporates UI adaptability, ensuring clarity between stopwatch and countdown modes. It is suitable for learning purposes and as a demonstration of Java programming skills.