

Java Unit Converter Tool Documentation

This document provides an overview of the Java Unit Converter project, a command-line application that converts between various units of measurement.

1. Overview

The Java Unit Converter is a simple command-line tool that enables users to convert values between various units of measurement, including temperature, distance, and weight. It supports common metric and imperial units, provides a persistent history of conversions, and includes user-friendly commands for interaction.

The application is designed with an object-oriented approach, separating concerns into distinct packages for unit representation, conversion logic, and file handling.

2. Features

- **Multi-Category Conversions:** Supports three different categories of measurement.
 - Temperature: Celsius (c), Fahrenheit (f), and Kelvin (k).
 - Distance: Millimeter (mm), Centimeter (cm), Meter (m), Kilometer (km), and Miles (mi).
 - Weight: Gram (g), Kilogram (kg), and Pounds (lbs).
 - **Interactive Command-Line Interface (CLI):** A straightforward CLI for easy user interaction.
 - **Input Validation:** Checks user input for correct formatting (e.g., 10 kg), valid numbers, and supported units.
 - **Conversion History:** Automatically saves a log of all successful conversions to a text file (history/output.txt).
 - **History Management:** Allows users to view (history) or delete (delete) the entire conversion history.
-

3. How to Use

1. Compile and run the converterTool.java file.
2. The program will display a welcome message and a list of available commands.
3. Enter a value and its unit, separated by a space.
4. When prompted, enter the target unit you wish to convert to.
5. The result will be displayed on the screen and saved to the history log.

Example Session:

Input (ensure a space between the value and its unit "5 kg"):

25 km

Convert to:

mi

15.534323188828234 mi

Available Commands:

- `help`: Displays the list of supported units and available commands.
 - `history`: Prints the entire conversion history to the console.
 - `delete`: Clears all entries from the conversion history file.
 - `exit`: Terminates the application.
-

4. Project Structure

The project is organized into several packages, each with a specific responsibility.

- `com.converterTool.Main`
 - `converterTool.java`: The main entry point of the application. It contains the main method, handles the main application loop, processes user input, and coordinates calls to the converters and file handlers.
- `com.converterTool.Units`
 - `NumberWithUnits.java`: A base class that encapsulates a numeric value and its unitSign.
 - `Temperature.java`, `Distance.java`, `Weight.java`: Concrete classes that extend `NumberWithUnits` to represent specific types of measurements.
- `com.converterTool.Converter`
 - `ConverterInterface.java`: An interface that defines the contract for all converter classes, ensuring they have a `convertValues` method.
 - `TemperatureConverter.java`: Implements the `ConverterInterface` and contains the logic for converting between Kelvin, Celsius, and Fahrenheit.
 - `MetricConverter.java`: Implements the `ConverterInterface` and contains the logic for all distance and weight conversions.
- `com.converterTool.SaveFunctions`
 - `ReadAndWrite.java`: A utility class with static methods for handling all file I/O operations, including reading, writing, appending, and deleting the `history/output.txt` file.