**Documentation for Text Processing and Data Management Tool**

**Overview**

This project is a text processing and data management tool developed in Java. It enables users to perform complex operations on text data using regular expressions and efficiently manipulate collections of data using Java collections. The user interface is designed with JavaFX for a seamless and interactive experience.

**Key Features**

**1. Regular Expressions (Regex) Module**

* **Functionality:**
  + Search, match, and replace text using regex.
  + Support for advanced regex features such as sets, ranges, alternations, shorthand character classes, and quantifiers.
* **User Interaction:**
  + Input regex patterns and text data through a user-friendly JavaFX interface.
  + Display matched text and replacement results.

**2. Text Processing Module**

* **Functionality:**
  + Search for patterns and replace characters in text using regex.
  + Utilize Java's Pattern and Matcher classes for efficient operations.
* **Output:**
  + Results of search and replace operations are displayed in the user interface.

**3. Data Management Module**

* **Functionality:**
  + Manage collections using Java's built-in data structures (e.g., ArrayList, Set, Map).
  + Allow users to create, update, and delete entries in collections.
* **Custom Implementations:**
  + Override hashCode() and equals() methods in custom objects to ensure proper behaviour in collections.

**4. User Interface (UI)**

* **Technology:**
  + Developed using JavaFX for an intuitive and visually appealing interface.
* **Capabilities:**
  + Input fields for text data and regex patterns.
  + Buttons and options to trigger regex and collection operations.
  + Dynamic display of results.

**Implementation Details**

**Regex Module**

* **Classes and Methods:**
  + RegexProcessor: Contains methods for performing regex operations (match, replace).
  + Usage of Pattern.compile() and Matcher.find() for processing.
* **Error Handling:**
  + Validate regex patterns before processing.
  + Catch invalid pattern exceptions and display error messages to the user.

**Text Processing Module**

* **Integration:**
  + Built on top of the Regex Module.
  + Designed to chain operations like searching and replacing within a single workflow.

**Data Management Module**

* **Core Components:**
  + Use of Java collections for efficient data handling.
  + Support for CRUD (Create, Read, Update, Delete) operations on collections.
* **Custom Objects:**
  + Implementation of hashCode() and equals() for data consistency.

**User Interface**

* **Design Elements:**
  + Text fields for data input.
  + Buttons for triggering regex and data operations.
  + Output areas for displaying results.
* **Error Display:**
  + Use alert dialogs for error notifications.

**Advanced Features**

* Dynamic preview of regex matches in real-time.
* History of operations for undo/redo functionality.
* Export results to a file (e.g., text or JSON format).

**Code Quality**

* Adherence to Java coding conventions.
* Modular code with comprehensive documentation.
* Reusable components to facilitate extension.

**Error Handling**

* Use of try-catch blocks to prevent crashes.
* Custom error messages for regex compilation and runtime errors.
* Notifications via the UI for user guidance.

**Future Enhancements**

* Add support for more advanced regex constructs (e.g., lookaheads, lookbehinds).
* Implement visualization for collection data structures.
* Enable import/export of data to/from external files.

**Conclusion**

This tool serves as a comprehensive solution for text processing and data management using Java. The combination of regex functionalities, collection management, and a robust UI provides users with a powerful and interactive experience.