Java Collection Framework Project Ideas

Feb 11, 2024

Project ideas that will help students test and deepen their understanding of Java collections through command-line applications

1. Task Manager

Outline:

- **Description**: Develop a command-line task manager that allows users to add, view, mark as completed, and delete tasks.
- **Features**:
  - Utilize a HashMap to store tasks, where the task name serves as the key and the task details (e.g., description, due date) serve as the value.
  - Implement functionality to add tasks, view all tasks, mark tasks as completed, and delete tasks.
  - Provide options to filter tasks based on their status (e.g., completed, pending).
  - Allow users to save and load tasks from a file to maintain persistence across sessions.
- **Implementation Steps**:
  a. Design a Task class to represent individual tasks, including attributes like name, description, due date, and completion status.
  b. Create a TaskManager class to manage the task operations (add, view, mark as completed, delete).
  c. Use a HashMap to store tasks, where the task name is the key and the Task object is the value.
  d. Implement command-line interface (CLI) for user interaction, including options to add, view, mark as completed, and delete tasks.
  e. Integrate file handling to save and load tasks from a file for persistence.

2. Quiz Application

Outline:

- **Description**: Build a command-line quiz application where users can take quizzes on various topics.
- **Features**:
  - Use ArrayList or LinkedList to store quiz questions and answers.
  - Allow users to select a quiz topic and attempt the quiz.
  - Provide feedback on each answer (correct/incorrect) and display the final score at the end.
  - Support multiple-choice questions and true/false questions.
  - Include a timer to limit the time for each question or the entire quiz.
- **Implementation Steps**:
  a. Define classes for QuizQuestion and QuizTopic to represent questions and topics, respectively.
  b. Create an interface for the quiz application, allowing users to select a topic and start the quiz.
  c. Implement methods to load quiz questions and answers from a file into collections (ArrayList/LinkedList).
  d. Develop logic to present questions to the user, receive and validate answers, and calculate the final score.
  e. Incorporate a timer using Java's Timer or ExecutorService to manage time limits for questions or the entire quiz.

3. Contact Management System

Outline:

- **Description**: Develop a command-line contact management system that allows users to add, view, update, and delete contacts.
- **Features**:
  - Utilize TreeSet or TreeMap to store contacts, sorted alphabetically by name.
  - Implement functionalities to add new contacts, view all contacts, search for contacts by name, update contact details, and delete contacts.
  - Ensure uniqueness of contacts based on their name or unique identifier.
  - Provide options to export/import contacts to/from a file for backup and sharing.
- **Implementation Steps**:
  a. Create a Contact class to represent individual contacts, including attributes like name, phone number, email, etc.
  b. Use TreeSet or TreeMap to store contacts, ensuring they are sorted alphabetically by name.
  c. Develop methods to add, view, search, update, and delete contacts within the ContactManager class.
  d. Design a command-line interface for user interaction, including options to perform various contact management operations.
  e. Integrate file handling functionalities to allow users to export/import contacts to/from a file for backup and sharing.