D. SIVAKARTHIKEYAN

Project overview:

Objective: Develop a simple to-do list application using java with an emphasis on functions and data structures.

key components:

1. Funtions(methods): In java, functions are referred to as method. You'll be implementing various methods to handle different aspects of the to-do list application. Methods are modular blocks of code that perform specific tasks, making your more organized and easier to understand.

Method to add a task

Method to delete a task

Method to display the list of tasks

Method to mark a task as complete

2. Data Structures: Utilize appropriate data structures to store and manage the to-do-list. Common choices in Java include

ArrayList, LinkedList, or HashMap, but you can explore other options based on your creativity and understanding.

PROGRAM:

import java.util.ArrayList;

import java.util.Scanner;

public class ToDoList {

    private static ArrayList<String> tasks = new ArrayList<>();

    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {

        int choice;

        do {

            System.out.println("\nTo-Do List Application");

            System.out.println("1. Add Task");

            System.out.println("2. Delete Task");

            System.out.println("3. Display Tasks");

            System.out.println("4. Mark Task Complete");

            System.out.println("5. Exit");

            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();

            scanner.nextLine(); // Consume remaining newline character

            switch (choice) {

                case 1:

                    addTask();

                    break;

                case 2:

                    deleteTask();

                    break;

                case 3:

                    displayTasks();

                    break;

                case 4:

                    markTaskComplete();

                    break;

                case 5:

                    System.out.println("Exiting application...");

                    break;

                default:

                    System.out.println("Invalid choice!");

            }

        } while (choice != 5);

        scanner.close();

    }

    private static void addTask() {

        System.out.print("Enter task description: ");

        String task = scanner.nextLine();

        tasks.add(task);

        System.out.println("Task added successfully!");

    }

    private static void deleteTask() {

        if (tasks.isEmpty()) {

            System.out.println("No tasks to delete!");

            return;

        }

        displayTasks();

        System.out.print("Enter the index of the task to delete (1-" + tasks.size() + "): ");

        int index = scanner.nextInt() - 1; // Adjust for zero-based indexing

        if (index >= 0 && index < tasks.size()) {

            tasks.remove(index);

            System.out.println("Task deleted successfully!");

        } else {

            System.out.println("Invalid index!");

        }

    }

    private static void displayTasks() {

        if (tasks.isEmpty()) {

            System.out.println("No tasks in the list!");

            return;

        }

        System.out.println("\nYour Tasks:");

        for (int i = 0; i < tasks.size(); i++) {

            System.out.println((i + 1) + ". " + tasks.get(i));

        }

    }

    private static void markTaskComplete() {

        if (tasks.isEmpty()) {

            System.out.println("No tasks to mark complete!");

            return;

        }

        displayTasks();

        System.out.print("Enter the index of the task to mark complete (1-" + tasks.size() + "): ");

        int index = scanner.nextInt() - 1; // Adjust for zero-based indexing

        if (index >= 0 && index < tasks.size()) {

            // Simple implementation: Add "(completed)" to the task description

            tasks.set(index, tasks.get(index) + " (completed)");

            System.out.println("Task marked complete!");

        } else {

            System.out.println("Invalid index!");

        }

    }

}

OUTPUT:



