JAVA SIMPLE TERMINAL BASED TO DO LIST APP

CODE:

```
import java.util.ArrayList;
import java.util.Scanner;

public class ToDoListApp {

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


    private static void addTask(String task) {
        toDoList.add(task);
        System.out.println("Task added: " + task);
    }


    private static void deleteTask(int index) {
        if (index >= 0 && index < toDoList.size()) {
            String removedTask = toDoList.remove(index);
            System.out.println("Task removed: " + removedTask);
        } else {
            System.out.println("Invalid index. Task not removed.");
        }
    }


    private static void displayTasks() {
        if (toDoList.isEmpty()) {
            System.out.println("No tasks in the to-do list.");
        } else {
            System.out.println("To-Do List:");
            for (int i = 0; i < toDoList.size(); i++) {
                System.out.println((i + 1) + ". " + toDoList.get(i));
            }
        }
    }


    private static void markAsComplete(int index) {
        if (index >= 0 && index < toDoList.size()) {
            String task = toDoList.get(index);
            toDoList.set(index, "[Completed] " + task);
            System.out.println("Task marked as complete: " + task);
        } else {
            System.out.println("Invalid index. Task not marked as complete.");
        }
```

```
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            // Display menu
            System.out.println("\n=== To-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 as Complete");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");

            // Get user input
            choice = scanner.nextInt();
            scanner.nextLine();


            switch (choice) {
                case 1:
                    System.out.print("Enter task to add: ");
                    String taskToAdd = scanner.nextLine();
                    addTask(taskToAdd);
                    break;
                case 2:
                    System.out.print("Enter the index of the task to delete:
");
                    int indexToDelete = scanner.nextInt();
                    deleteTask(indexToDelete - 1);
                    break;
                case 3:
                    displayTasks();
                    break;
                case 4:
                    System.out.print("Enter the index of the task to mark as
complete: ");
                    int indexToComplete = scanner.nextInt();
                    markAsComplete(indexToComplete - 1);
                    break;
                case 0:
                    System.out.println("Exiting the application. Goodbye!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
```

```
        } while (choice != 0);


        scanner.close();
    }
}
```

```
 === To-Do List Application ===
 1. Add Task
 2. Delete Task
 3. Display Tasks
 4. Mark Task as Complete
 0. Exit
 Enter your choice: 1
 Enter task to add: Complete assignment
 Task added: Complete assignment

 === To-Do List Application ===
 1. Add Task
 2. Delete Task
 3. Display Tasks
 4. Mark Task as Complete
 0. Exit
 Enter your choice: 1
 Enter task to add: Submit task
 Task added: Submit task

 === To-Do List Application ===
 1. Add Task
 2. Delete Task
 3. Display Tasks
 4. Mark Task as Complete
 0. Exit
 Enter your choice: 1
 Enter task to add: Netflix & Chill
 Task added: Netflix & Chill

=== To-Do List Application ===
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
0. Exit
Enter your choice: 4
Enter the index of the task to mark as complete: 1
Task marked as complete: Complete assignment

=== To-Do List Application ===
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
0. Exit
Enter your choice: 4
Enter the index of the task to mark as complete: 2
Task marked as complete: Submit task

=== To-Do List Application ===
1. Add Task
2. Delete Task
3. Display Tasks
4. Mark Task as Complete
0. Exit
Enter your choice: 4
Enter the index of the task to mark as complete: 3
Task marked as complete: Netflix & Chill

 === To-Do List Application ===
 1. Add Task
 2. Delete Task
 3. Display Tasks
 4. Mark Task as Complete
 0. Exit
 Enter your choice: 3
 To-Do List:
 1. [Completed] Complete assignment
 2. [Completed] Submit task
 3. [Completed] Netflix & Chill

 === To-Do List Application ===
 1. Add Task
 2. Delete Task
 3. Display Tasks
 4. Mark Task as Complete
 0. Exit
 Enter your choice: 0
 Exiting the application. Goodbye!
```

1. Import Statements

```
import java.util.ArrayList;
import java.util.Scanner;
```

These statements import necessary classes from the **java.util** package. **ArrayList** is used for dynamic arrays, and **Scanner** is used to take user input.

2. Class Definition

```
public class ToDoListApp {
```

Defines the main class for the to-do list application.

3. Private Static ArrayList

```
private static ArrayList<String> toDoList = new ArrayList<>();
```

Declares a private static ArrayList named toDoList to store the tasks.

We are using ArrayList Data Structure to get dynamic array to store ***x*** number of tasks given by user.

4. Methods for To-Do List Operations

4.1. **addTask** Method

```
 private static void addTask(String task) {
     // Adds a task to the to-do list
     toDoList.add(task);
     System.out.println("Task added: " + task);
 }
```

4.2. **deleteTask** Method

```
private static void deleteTask(int index) {
    // Deletes a task from the to-do list based on the provided index
    if (index >= 0 && index < toDoList.size()) {
        String removedTask = toDoList.remove(index);
        System.out.println("Task removed: " + removedTask);
    } else {
        System.out.println("Invalid index. Task not removed.");
    }
}
```

4.3. **displayTasks** Method

```
private static void displayTasks() {
    // Displays the current tasks in the to-do list
    if (toDoList.isEmpty()) {
        System.out.println("No tasks in the to-do list.");
    } else {
        System.out.println("To-Do List:");
        for (int i = 0; i < toDoList.size(); i++) {
            System.out.println((i + 1) + ". " + toDoList.get(i));
        }
    }
}
```

4.4. **markAsComplete** Method

```
private static void markAsComplete(int index) {
    // Marks a task as complete based on the provided index
    if (index >= 0 && index < toDoList.size()) {
        String task = toDoList.get(index);
        toDoList.set(index, "[Completed] " + task);
        System.out.println("Task marked as complete: " + task);
    } else {
        System.out.println("Invalid index. Task not marked as complete.
    }
}
```

5. Main Method

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        // Display menu
        // ...

        // Get user input
        // ...

        // Perform the selected action
        // ...

    } while (choice != 0);

    // Close the scanner
    scanner.close();
}
```

The **main** method serves as the entry point for the application. It contains a loop that displays a menu, takes user input, and calls the corresponding method based on the user's choice.

6. User Interface (Menu)

```
// Display menu
System.out.println("\n=== To-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 as Complete");
System.out.println("0. Exit");
System.out.print("Enter your choice: ");
```

Displays a simple menu for the user to choose from.

7. User Input Handling

```
// Get user input
choice = scanner.nextInt();
scanner.nextLine(); // Consume the newline character
```

Takes the user's choice as input and consumes the newline character.

8. Switch Statement for Menu Options

```
// Perform the selected action
switch (choice) {
    case 1:
        // Add Task
        // ...
        break;
    case 2:
        // Delete Task
        // ...
        break;
    case 3:
        // Display Tasks
        // ...
        break;
    case 4:
        // Mark Task as Complete
        // ...
        break;
    case 0:
        // Exit
        // ...
        break;
    default:
        // Invalid choice
        // ...
}
```

Executes the corresponding action based on the user's choice using a switch statement.

9. Exiting the Application.

```
} while (choice != 0);

// Close the scanner
scanner.close();
```

Exits the loop when the user chooses to exit (entering 0) and closes the scanner.