Daily Task Manager Report | Object-Oriented Programming
Michele Stevany Venda Dati | 0806022410021

Program Workflow

1. Program starts.
2. The user selects between Array-Based or Linked List-Based task management.
3. Based on the selection, they can:
   - Array : View, update, complete, and undo tasks.
   - Linked List : Add, remove, view, complete, and undo tasks dynamically.
4. When a task is marked as completed, it is pushed onto the undo stack.
5. Undoing restores the most recent completed task from the stack.
6. The menu loops until the user chooses to exit.
7. Program ends.

Approach to Implementing Each Data Structure

1. Array
   This method uses a fixed-size array to store tasks, making it simple to view and update tasks.
2. Linked List
   This method offers a more flexible way to handle tasks dynamically. With this approach, user can add new task without being by a fixed-size and is able to remove tasks either by position or description.
3. Stack
   To implement the undo feature, a stack (Stack<String>) is used. When a task is marked as completed, its original version is saved in the stack. If the user decides to undo the completion, the task is restored from the stack to its previous state

Challenges and Solution

One of the challenges in the program was that the undo feature was not restoring the most recently completed task correctly. At first, it would restore the first completed task it found instead of the last one. To fix this, the program was adjusted to check tasks in reverse order for arrays and track the last completed task in linked list to ensure the most recent one was stored properly.

Another challenge was there was an issue when users tried to undo a task when no task had been completed yet. Since the undo feature realies on a stack, trying to undo with an empty stack caused the program to crash. This problem was solved by adding a check (error handling) to make sure that there was a tleast one completed task before attempting to undo, so the program can run smoothly without errors.