19. Ability to undo an action that has been applied

Your software needs to be able to include undo functionality, which, these days, is present in most user interfaces that allow you to edit content.

This is a problem that UI developers have been required to deal with for decades, and, as such, it can be solved by a number of standardized design patterns.

Suitable design patterns

Memento

Memento is a design pattern that was intended specifically to deal with this problem.

Memento is the name of an object that internally stores a sequence of snapshots. But it doesn't reveal those details to the outside world. The only accessible endpoints that the memento objects would have would be related to the functionality that the originator class, such as an editor, would require. These may include passing a new snapshot into the memento, restoring old snapshots, etc.

Whenever a change is made in a specific object, a new snapshot of this object is created and it's sent to the memento object, which appends it to its internal sequence of other snapshots. Then, if needed, the state of the originator can be restored from the history inside the memento.

Why would you use Memento

- Allows to store a full history of changes, so any of the snapshots can be restored easily.
- Object-oriented encapsulation is used well.

Command

The Command design pattern that we had a look at in **chapter 18** is also suitable for undoing and redoing actions. To use it in such a way, you would just need to create a history of Command objects and, for each one of them, add the ability to roll it back.

Unlike Memento, Command doesn't manage the internal states of the objects. So, it won't be suitable when private fields need to get their values changed while undoing an action. However, commands can be changed in such a way that you save the instructions, but don't save a collection of the actual states. This way, you don't restore previous states. You merely revert previous actions by executing the opposite actions. And this allows you to use the memory more efficiently.

Why would you use Command instead of Memento

- More efficient use of memory, as you don't have to store each individual snapshot.
- The actual steps are easier to trace.