CS 319 - Object-Oriented Software Engineering

Design Patterns Homework

Vedat Eren Arıcan - 22002643

1 Design Patterns Identified

1.1 Decorator Pattern

The options (i.e., `TrackElapsedTime`) that are available for *decorating* a task instance are fit for the use of this pattern. They present a need to be able to add functionality to an existing type. Particularly of note is that these options can be **combined**, which this pattern lets us do.

Implementation: `BaseTodoTaskDecorator`, `TimeTrackingDecorator`, `StatusHistoryDecorator`

1.2 Strategy Pattern

The various ways in which a list can have its contents sorted is fit for the use of this pattern. Note that there should only be one way of sorting for a given list, which is possible with a single strategy.

Implementation: `ITodoTaskSortingStrategy`, `BaseTodoTaskSortingStrategy`, `AlphabeticalSortingStrategy`, `AddOrderSortingStrategy`, `TargetDateSortingStrategy`

1.3 Composite Pattern

Each list can store tasks and other lists within. Since these nested lists may have other objects inside, we can see a clear tree-like structure. This calls for the composite pattern.

Implementation: `ITodoComponent`, `ITodoTask`, `TodoList`

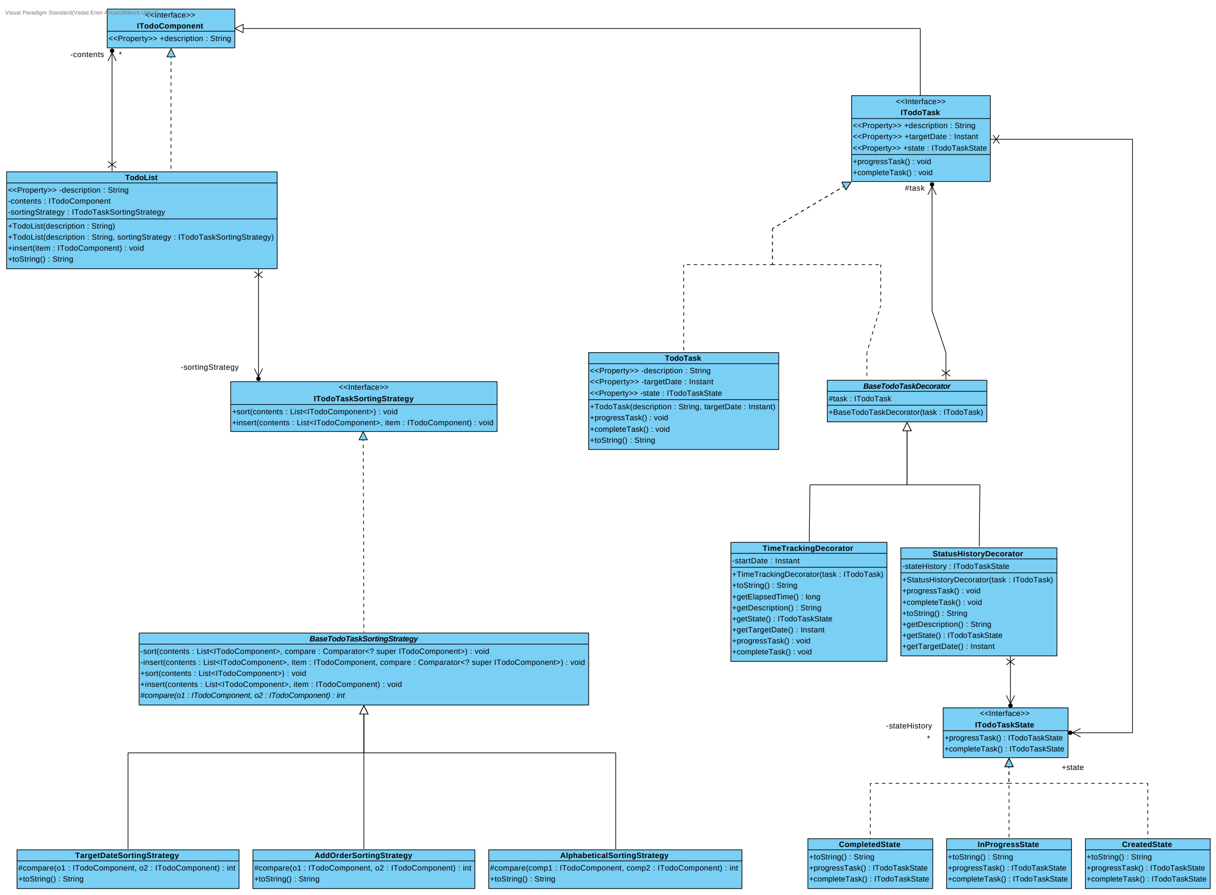
1.4 State Pattern

The various states a task can take are fit for the use of this pattern. We can model the *created*, *in progress*, *completed* states this way.

Implementation: `ITodoTaskState`, `CreatedState`, `InProgressState`, `CompletedState`

2 Class Diagram

Note that the diagram below is in vector format. It can be zoomed into without hurting picture quality.



Visual Paradigm Standard(Vedat Eren Arıcan(Bilkent Univ.))
<<Interface>>
ITodoComponent
<<Property>> +description : String
-contents
*
TodoList
<<Property>> -description : String
-contents : ITodoComponent
-sortingStrategy : ITodoTaskSortingStrategy
+TodoList(description : String)
+TodoList(description : String, sortingStrategy : ITodoTaskSortingStrategy)
+insert(item : ITodoComponent) : void
+toString() : String
-sortingStrategy
<<Interface>>
ITodoTaskSortingStrategy
+sort(contents : List<ITodoComponent>) : void
+insert(contents : List<ITodoComponent>, item : ITodoComponent) : void
BaseTodoTaskSortingStrategy
-sort(contents : List<ITodoComponent>, compare : Comparator<? super ITodoComponent>) : void
-insert(contents : List<ITodoComponent>, item : ITodoComponent, compare : Comparator<? super ITodoComponent>) : void
+sort(contents : List<ITodoComponent>) : void
+insert(contents : List<ITodoComponent>, item : ITodoComponent) : void
#compare(o1 : ITodoComponent, o2 : ITodoComponent) : int
TargetDateSortingStrategy
#compare(o1 : ITodoComponent, o2 : ITodoComponent) : int
+toString() : String
AddOrderSortingStrategy
#compare(o1 : ITodoComponent, o2 : ITodoComponent) : int
+toString() : String
AlphabeticalSortingStrategy
#compare(comp1 : ITodoComponent, comp2 : ITodoComponent) : int
+toString() : String
<<Interface>>
ITodoTask
<<Property>> +description : String
<<Property>> +targetDate : Instant
<<Property>> +state : ITodoTaskState
+progressTask() : void
+completeTask() : void
#task
TodoTask
<<Property>> -description : String
<<Property>> -targetDate : Instant
<<Property>> -state : ITodoTaskState
+TodoTask(description : String, targetDate : Instant)
+progressTask() : void
+completeTask() : void
+toString() : String
BaseTodoTaskDecorator
#task : ITodoTask
+BaseTodoTaskDecorator(task : ITodoTask)
TimeTrackingDecorator
-startDate : Instant
+TimeTrackingDecorator(task : ITodoTask)
+toString() : String
+getElapsedTime() : long
+getDescription() : String
+getState() : ITodoTaskState
+getTargetDate() : Instant
+progressTask() : void
+completeTask() : void
StatusHistoryDecorator
-stateHistory : ITodoTaskState
+StatusHistoryDecorator(task : ITodoTask)
+progressTask() : void
+completeTask() : void
+toString() : String
+getDescription() : String
+getState() : ITodoTaskState
+getTargetDate() : Instant
-stateHistory
*
<<Interface>>
ITodoTaskState
+progressTask() : ITodoTaskState
+completeTask() : ITodoTaskState
+state
CompletedState
+toString() : String
+progressTask() : ITodoTaskState
+completeTask() : ITodoTaskState
InProgressState
+toString() : String
+progressTask() : ITodoTaskState
+completeTask() : ITodoTaskState
CreatedState
+toString() : String
+progressTask() : ITodoTaskState
+completeTask() : ITodoTaskState