

Se desea extender la clase *ToDoItem* definida en los trabajos prácticos. La misma representaba una tarea con (i) un nombre, (ii) prioridad y (iii) una marca de cumplido-no cumplido. La extensión se desea realizar en dos aspectos:

(I) Por un lado, se pretende definir una serie de pasos en lugar de una simple marca: cumplido-no cumplido. Los pasos por los que debe atravesar son: (i) nuevo, (ii) comenzado y (iii) finalizado. La clase *ToDoItem* debe reconocer el siguiente protocolo:

ToDoItem class>> new

"instancia un *ToDoItem* nuevo en estado pending"

ToDoItem>> start

"pasa el *ToDoItem* a started (siempre y cuando su estado actual sea pending, si se encuentra en otro estado, no hace nada)"

ToDoItem>> finish

"pasa el *ToDoItem* a finished (siempre y cuando su estado actual sea started, si se encuentra en otro estado, no hace nada)"

Tenga presente que la secuencia de estados debe ser pending, started y finished. Si se envían mensajes en otro orden no se cambia de estado.

(II) Por otro lado, se desea monitorear cuando un *ToDoItem* cambia de estado. Tenga presente que puede haber objetos de diferentes clases interesados en ser notificados del cambio de estado. Su solución debe permitir a cualquier objeto interesado en ser notificado del cambio. Para poder probar como proporciona esta funcionalidad, considere que existe una clase *ToDoItemView* que muestra en una ventana un *ToDoItem* con diferentes colores de acuerdo al estado en que se encuentra. La clase *ToDoItemView* debe implementar

ToDoItemView class>> new: aToDoItem
"Instancia a inicializa un *ToDoItemView*"

ToDoItemView>> initialize: aToDoItem
"Debe hacer las definiciones necesarias para que el *ToDoItemView* queda atento a los cambios en los pasos de aToDoItem"

ToDoItemView>> "defina el nombre del metodo"
"Este método es invocado cuando el *ToDoItem* cambia de estado.
No se preocupe por implementar como se cambia de color."

Debe hacer los ajustes necesario en los métodos necesarios de *ToDoItem*, para que cuando cambie de estado, se desencadene el mecanismo que hace que se invoque el método previo.

Tareas:

- 1.- Realice el diagrama de clases.
- 2.- Implemente los mensajes indicados.
- 3.- Escriba en un workspace las instrucciones necesarias para:
 - Instanciar un *ToDoItem*
 - Instanciar un *ToDoItemView* atento al *ToDoItem* previo
 - Cambiar el *ToDoItem* a started
 - Cambiar el *ToDoItem* a finished