

En un momento borre el marsRover de los loggers porque me decia que no se usaba y me pasaba los tests. Pero despues me meti en el mensaje de inicializacion y vi que estaba en 3 lineas que parecian hacer falta. Ahora tengo miedo de que por alguna razon me sigan andando los tests cuando no deberian y que cuando suba las modificaciones a vos te dejen de pasar las cosas porque requeriria alguna variable que tengo guardada local o algo asi.

Cambios al proxy:

- Se eliminaron sus subclases. Hacian exactamente lo mismo pero tenian las variables con distinto nombre.
- Se elimino el colaborador rover. En ningun momento se le usaba.
- Se renombro la clase a ObservableProxy
- Se renombro collaboratorValue a value. Se me hizo mas logico, se supone que su uso debe ser lo mas transparente posible. Lamentablemente no le puedo renombrar el mensaje := porque no es un mensaje.
- Se renombro modifyCollaborator a assign. Como no puedo renombrar := y esencialmente es eso lo que hace, decidi ponerle ese nombre.

Cambios a los loggers:

- Se eliminaron los 2 colaboradores que no eran el marsRover en la clase abstracta.

-----

Juro que no intento activamente darte un aneurisma (aunque esta vez me quedo muchisimo mas limpio que lo de la otra vez).

Resulta que andabamos siempre guardando 2 variables y haciamos el seguimiento de tanto el heading como la posicion. El enunciado pide explicitamente que solo se haga el seguimiento de lo que fue configurado. Ademas pide que sea posible crear un nuevo logger reconfigurado en el momento.

Para esto inicialmente se me ocurrio crear varios mensajes de creacion de instancias, pero a medida que crecen los posibles colaboradores del rover, crece exponencialmente la cantidad de mensajes de creacion de instancias. No me parecia que escale.

Como cada logger puede hacer lo que se le cante (nos dieron como 3 formas de formateo del log distintas, mezclando idiomas...) y dijeron que potencialmente en el futuro puedan haber cosas falopa como copiarlo a memoria, mandar mensajes a un microservicio, etc. Use closures para esta parte.

Me costo mucho encontrar una forma de poder mandarlos y que quede bien, no sabia ni como escribir el closure, ademas debian ir en una coleccion de alguna forma porque sino estas limitando la cantidad de variables que puede estar observando. Igual ando contento porque creo que ahora entiendo mejor como funcionan los closures.

AHORA SI, VAMOS AL LISTADO DE CAMBIOS:

- Se eliminaron las subclases del logger. Ahora es muy general (tal vez me pasé, pero creo que era la idea)
- Ahora recibe una coleccion de proxies a los que se va a suscribir y una coleccion con closures que indican el como debe comportarse el observer para cada variable.

(Es importante mantener la relacion entre los indices de las colecciones porque estan mapeados 1-1)

- El logger dejo de ser un logger, ahora es un watcher (Complice aun mas la distincion entre la camara y el que ve la tele, no se cual es el observer, supongo nos enteraremos mañana). La funcion de logger viene exclusivamente por como estan conformados los closures que se le envian.

- Meti adentro de un method object (no estoy seguro de que lo sea, tiene mas de 1 mensaje de computar y no recibe parametros) la funcionalidad de crear los closures para los loggers del rover.

Tiene multiples formas de agregar elementos porque habia distintos formateos y ademas habia veces que debian venir como OrderedCollection.

Si en algun momento te metes adentro de los de colecciones vas a ver que son medio ilegibles, sin embargo no quise hacerles refactor a esos porque tenia miedo de que al agregar mensajes adentro de el objeto ese cambie el scope cuando lleguen al observer. Ando priorizando otras cosas asi que por ahora lo voy a dejar asi, despues si me sobra el tiempo veo si funciona al hacerles algun extract method.

Este es el bicho que estamos usando en los tests, como solo nos dieron cosas que espian a 2 variables como mucho no esta tan picante. Aun asi, logran que los tests queden legibles y, en caso de agregarse nuevos colaboradores y loggers que funcionen de forma similar a alguno de los formateos ya creados, deberia funcionar sin ser necesario agregarle otro tipo de add.

Me parece falopa que las colecciones que manda empiezan siendo closures pero despues a la que dice como es que se hacen las cosas, le vas reemplazando ese closure por el valor posta y de alguna forma el del proxy se mantiene aunque mepa que es una referencia, ni idea como funciona eso.

-----

Estamos medio flojos de tests, en todo esto, solo agregue un tests, recontra colgue. Chequea que, efectivamente, se pueda suscribir a 2 variables. Asumo si lo de los profesores esta bien, entonces nunca va a llamar a algun cambio en alguna variable cuando haya algun error, no se si corresponda testear eso. Seguramente lo pasa de 1.

Te mando tambien una version intermedia por si te llega a servir para visualizar, no me acuerdo en que momento estaba, creo que todavia no habia cambiado los tests. Dudo que esta te sirva pero ya me diras que tan necesaria sea y que tan ilegible haya quedado el modelo.

Lamentablemente voy a levantarme alrededor de las 13, hasta ese momento no creo que vaya a poder responderte consultas, exceptuando un deficit de sueño grande.

Hay algunos nombres que no creo que les vayan a gustar, creo recordar que habian dicho que seria un toque nefasto decir que se suscriben, aunque a mi me parece re facil de entender y que hasta podria ir con la metafora de la camara, los videntes y la television en general. Mepa que si intento mejorarlos ahora voy a trollear, ya paso mi peak de lucidez hace rato.

Exceptuando que tal vez hay algunos nombres no optimos y que tal vez no tenemos el mejor cubrimiento con tests, lo veo muy razonable, principalmente porque el ProxyObservable y el Watcher ni saben que estan laburando con un rover. Tambien lo veo escalable, lo unico que tendrias que agregarle al rover para poder loggear el nuevo colaborador seria: El colaborador, su inicializacion como proxy y el mensaje que te devuelve el proxy. Despues todo el resto es por afuera del rover y lo de meterlo al "method object" es puramente por comodidad, tranquilamente podes crearte tu propia OrderedCollection a mano y funciona igual.