

LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

Sonar domain pattern observer

Requirements

Si desidera realizzare una versione osservabile (SonarObservable) che soddisfi i seguenti requisiti:

- Il SonarObservable deve inviare informazioni a tutti componenti software interessati alla rilevazione dei valori di distanza;
- Il SonarObservable deve fornire valori di distanza solo quando questi si modificano in modo significativo;
- I componenti interessati ai valori di distanza prodotti dal SonarObservable sono denominati Observer e possono risiedere sullo stesso nodo del SonarObservable (cioè sul RaspberryPi) o su un nodo remoto (ad esempio sul PC);

Requirement analysis

Domande del committente:

- Cosa significa osservare un dispositivo? Significa osservare i valori prodotti dal dispositivo (nel caso del Sonar significa la distanza rilevata)
- Si può dire che il valore della distanza sia una proprietà del dispositivo di misura (cioè del Sonar)? No, indica un valore esterno rilevato.
- Come formalizzare questo approccio nel caso del SonarObservable? Mediante l'utilizzo di modelli, ovvero di un linguaggio comprensibile alla macchina. (Per esempio nel caso di java mediante l'utilizzo di classi e interfacce)

Problem analysis

Un sonar osservabile può essere implementato mediante il pattern observer. Il pattern observer può essere sviluppato in due modi diversi:

- Modo 1: SonarObservable che registra le entità observer che devono essere aggiornate e le notifica con l'aggiornamento del valore mediante il metodo update
- Modo 2: SonarObservable che comunica con una entità terza che a sua volta registra le entità observer che devono essere aggiornate e le notifica con l'aggiornamento del valore mediante il metodo update

Il secondo approccio risulta preferibile avendo un miglior disaccoppiamento dei compiti ed andando quindi a rispettare il "single responsibility principle"

Test plans

Test plan sonar osservabile:

- 1) Viene posto un ostacolo a distanza fissa DTESTING1 davanti al sonar
- 2) Viene poi controllato che tutti gli Observers ricevano il valore DTESTING1.
- 3) Attendo qualche istante
- 4) Modifico la posizione dell'ostacolo a una nuova distanza DTESTING2
- 5) Controllo che gli tutti gli Observers ricevano il valore DTESTING2.

Project

Modo 1:

- 1) Costruisco la classe SonarObservable, implementando l'interfaccia ISonar.
- 2) Costruisco la classe SonarObservableMock e SonarObservableConcrete, estendendo SonarObservable.
- 3) Costruisco la classe SonarObserver, implementando l'interfaccia ISonarObserver.
- 4) Creo la classe di test TestSonarObservableMock

Modo 2:

- 1) Costruisco la classe SonarObservablePubSub, implementando l'interfaccia ISonar.
- 2) Costruisco la classe SonarObservableMockPubSub e SonarObservableConcretePubSub, estendendo SonarObservablePubSub.
- 3) Costruisco la classe SonarObserverBroker
- 4) Costruisco la classe SonarObserver, implementando l'interfaccia ISonarObserver.
- 5) Creo la classe di test TestSonarObservableMockPubSub

Testing

Deployment

Maintenance

By Davide Guidetti email: davide.guidetti3@studio.unibo.it
Git:
https://github.com/ilmada99/issLab2022_DavideGuidetti

