

Architettura logica

Friday, 31 March 2023

17:21

- E' l'organizzazione su larga scala delle classi software in packages, sottoinsiemi e strati
 - o Ogni strato è dipendente da quelli connessi
 - o Le dipendenze sono fatte dalle frecce
 - o La dipendenza è dall'alto verso il basso
- Si inizia a pensare su larga scala
 - o Codice
 - o Progetto
- Si definisce molto i requisiti non funzionali ed i vincoli
- Abbiamo una architettura a strati, es interfaccia utente, applicativa, servizi, ecc

L'architettura a strati:

- o Gli strati più alti usano servizi strati più bassi
 - NON bisogna implementare qualcosa di altri strati
Questo si chiama **separazione Modello-Vista**
Non permettere al dominio Sale di avere un oggetto JFrame
 - Pattern Observer è un Modello-Vista più rilassato
Dove gli oggetti dominio inviano messaggi agli oggetti UI indirettamente
Per indiretto vuol dire che, il dominio non sa cosa verrà usato dalla UI
- o Strato = gruppo molto grosso di classi, package o sottoinsiemi che hanno responsabilità rispetto ad un aspetto del sistema
E per progettare:
 - Strati inferiori = generali, potrebbero essere usati da più sistemi
 - Strati superiori = specifici, applicazione
- o Esistono 2 tipologie:
 - Stretta
Chi sopra richiama chi subito sotto senza poter chiamare quello 2 strati sotto
 - Rilassata
Chi sopra può chiamare chiunque è sotto di lui
- o Noi ci concentreremo allo strato della logica applicativa (dominio)
- o Vantaggi:
 - + riuso del codice per via degli strati più bassi

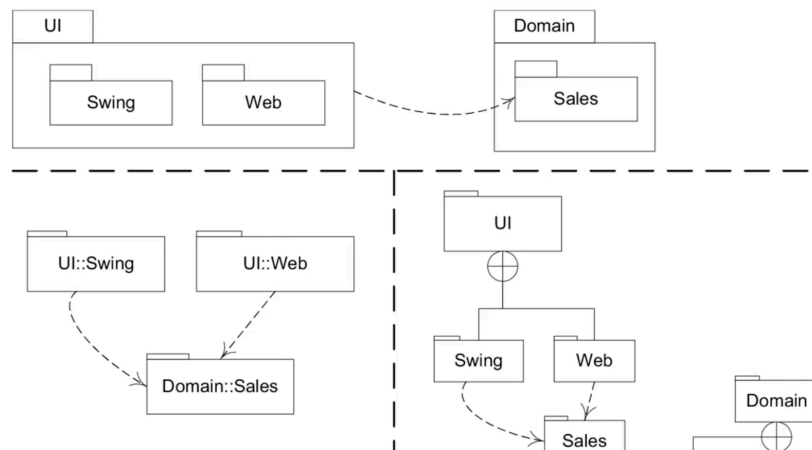
- Più uso del codice per via degli strati più bassi
- Indipendenza degli strati, ci importano input/output
- Meno complessità siccome dividiamo il problema
- Strati distribuiti su determinati strati
- Può essere dimostrata con un package UML
 - Che può raggruppare classi, package, ecc
 - Questo ci permette una maggiore organizzazione del codice

```

/** Strato UI */
com.mycompany.nextgen.ui.swing
com.mycompany.nextgen.ui.web
/** Strato DOMAIN */
// package specifici del progetto NextGen
com.mycompany.nextgen.domain.sales
com.mycompany.nextgen.domain.payments
/** Strato TECHNICAL SERVICES */
// il nostro strato di persistenza (accesso alla base di dati)
com.mycompany.service.persistence
// terze parti
org.apache.log4j
org.apache.soap.rpc
/** Strato FOUNDATION */
// package foundation creati dal nostro team
com.mycompany.util

```

- Un package rappresenta un namespace, quindi è possibile avere 2 package con lo stesso nome se sono su package diversi
- Per mostrare annidamento:



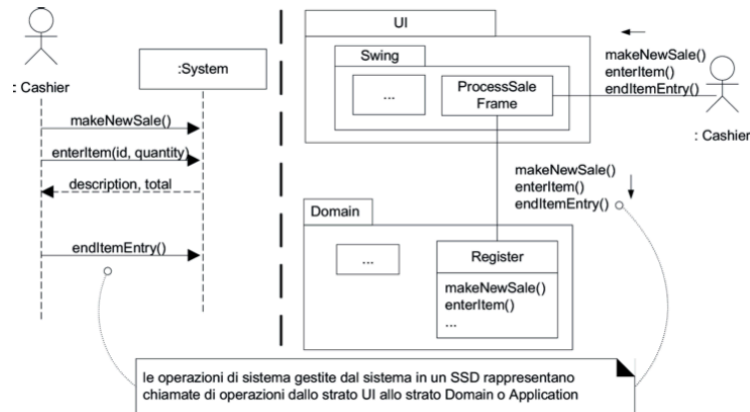
- Bisogna creare oggetti software simili a come li avevamo chiamati nel dominio
 - Chiamato oggetto di dominio
 - Più oggetti del dominio in uno stesso strato sono chiamati **strato del dominio**

Ma quindi, qual è la relazione tra stato del dominio e modello di dominio?

- Modello di dominio guarda le parti più interessanti dei concetti significativi del dominio
- Strato del dominio/Modello di progetto sono parti software con cui sono stati creati prendendo ispirazione dal modello di dominio, è una classe software
Il nome di questo è ispirato dal modello di dominio

- Definizioni:
 - Livello = Nodo fisico di elaborazione

- Strato = Sezione verticale dell'architettura
 - Partizione = Sottoinsieme di uno strato
- Messaggi inviati dallo strato UI (aka utente) allo strato del dominio sono ,essaggi mostrati negli SSD.
A sinistra possiamo vedere quando abbiamo una chiamata diretta, a destra indiretta



- Detto questo definiamo:
- Architettura software**
- Insieme decisioni significative sull'organizzazione di un sistema
 - Scelta elementi strutturali con interfacce
 - Collaborazione tra gli elementi
 - Si crea con elementi di larga scala