TECNICHE DI PROGRAMMAZIONE

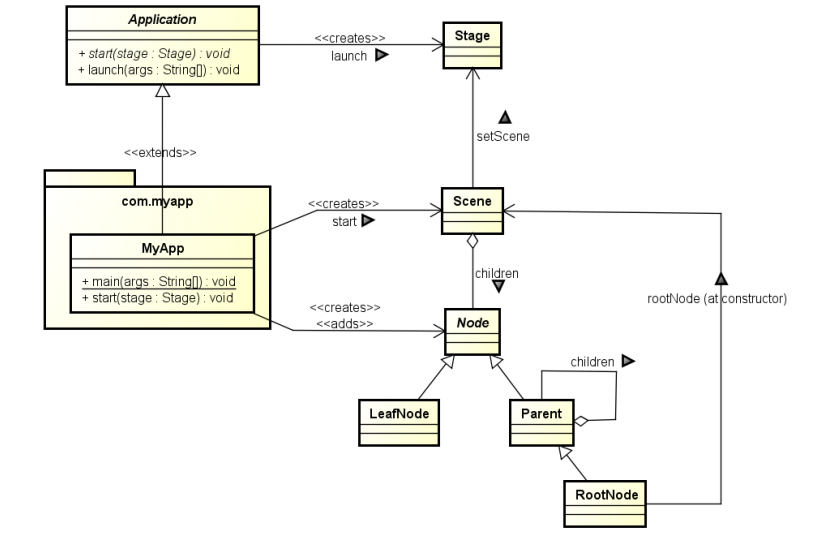
12/03/19

Nuovo pattern per avere le strutture dei nostri programmi più robuste.

Il controller dei nostri programmi precedenti (vedi ‘indovina numero’ ) contenevano sia la struttura fisica del nostro programma che la struttura logica. Per programmi più grandi diventa molto difficile gestire tutto.

Dobbiamo trovare quindi una nuova struttura. Tutto parte dall’application. Abbiamo un metodo main che contiene il ‘launch’ che inizializza tante cose e crea un oggetto stage e lo passa al metodo start. Questo oggetto stage lo usiamo nel metodo start per agganciarci la nostra scena.

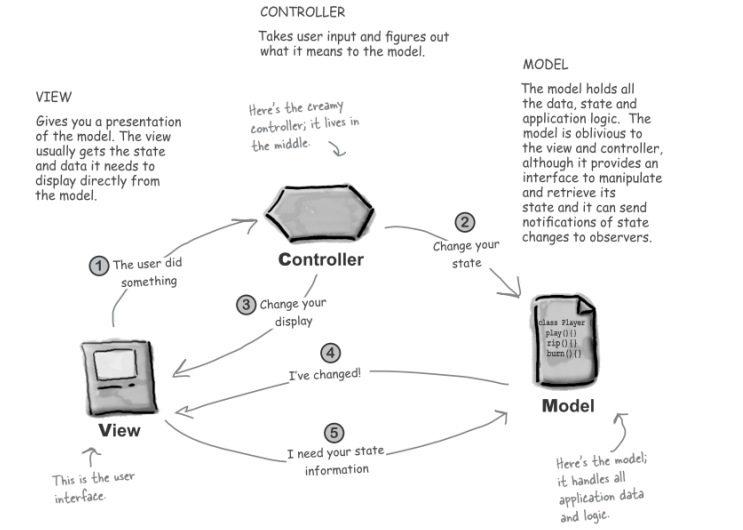
Una scena è un contenitore di nodi (TextArea, Label…). Un nodo può contenere altri nodi. Il nodo foglia non contiene altri nodi. Il nostro nodo router è il BorderPain. Facendo stage.show(); la nostra scena parte. Tutto spiegato nella figura qui sotto.



Ogni nodo ha delle “Properties”. Ci sono dei comandi che ci aiutano molto durante il programma e che vedremo in seguito.

Un Design Pattern è una soluzione standard che si è decisa di usare in determinate situazioni. E’ molto difficile programmare seguendo un Design Pattern, ma una volta imparato diventa molto più facile capirlo.

MVC Pattern: in ‘Indovina il Numero’ noi avevamo solo la parte View e Controller. Adesso noi aggiungiamo il modello che ha il compito di salvare tutti i dati della logica applicativa.



Nel caso in cui l’utente agisce sull’interfaccia modificando qualche dato del programma, la View comunica con il Model e non con il controller. La vista comunica al modello che è cambiata e il modello agisce di conseguenza.

Noi però passeremo ancora dal controller. Per capire se abbiamo fatto tutto bene, dobbiamo immaginare di dividere il modello dal controller e chiederci se il modello potremmo usarlo anche su altri programmi. Modello e controller devono essere separati tra loro, non devono avere cose in comune che li collegano.

Prima di fare il clone dobbiamo fare il fork.

Esercizio con Patter MVC.

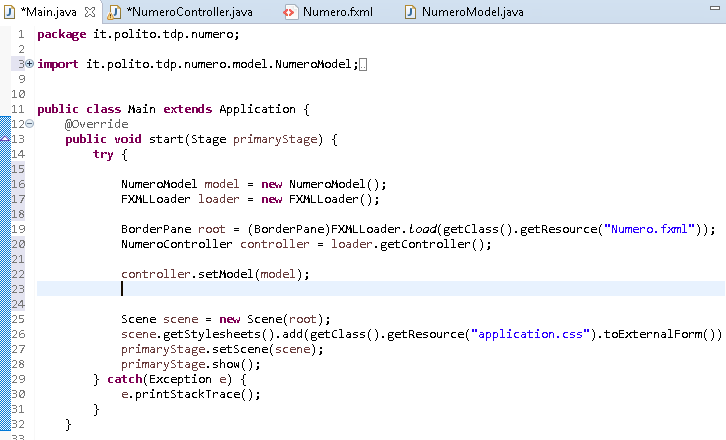
I branch possono servire quando voglio modificare un programma ma non sono sicuro se siano giuste o no. Nel caso in cui non siano giuste, posso fare ‘Re-emarge’ e ritornare al punto di partenza.

Chi crea il Controller? Lui viene creato automaticamente dalla Vista. Se apriamo il file FXML con SceneBuilder, in basso a sinistra c’è proprio un campo del menù a tendina che si chiama Controller.

Il Model lo creiamo nella classe Application. Aggiungo un metodo SET MODEL nel Controller che gli comunica che quello è il modello che deve utilizzare. Il modello può essere composto da piu classi.

Creo un Package nuovo dal src e dentro al package ci metto una nuova classe.

FXMLLoader ci permette di creare una scena a partire da un file FXML.



Modificare il main nel seguente modo (sarà sempre cosi da adesso in poi). Adesso noi andiamo a togliere della roba dal controller e la mettiamo nel model (NumeroModel).

Andiamo ad analizzare nel testo, tutte le variabili che servono nel modello ( che fanno parte del gioco, della sua logica). Prendiamo queste variabili e le spostiamo dal controller al model.