Java Console

SETTIMANA 1

PATTERN MVC

1° SETTIMANA

Il progetto di base che vi viene fornito è strutturato secondo il pattern MVC (Model View Controller). Questo pattern ci permette di dividere il nostro progetto in 3 livelli connessi tra loro:

- MODEL è il livello in cui sono definite le nostre entità, che devono essere inserite nel package omonimo, e da un altro pattern DAO (Data Access Object) nel quale viene interrogato il nostro database attraverso le query e vengono implementati i metodi per interagire con il DB.
- VIEW è il livello di rappresentazione dei dati all'utente (presentation layer). Vengono utilizzati i DTO (Data Transfert Object) che sono una rappresentazione del nostro modello.
- CONTROLLER in questo livello è confinata la nostra logica di business (Business Logic). L'utente riceve un input dalla view, il controller accede ai dati comunicando con il Model risponde alla view con un output.

Riassumendo le funzionalità dei vari packages:

1. MODEL contiene i business objects.
2. DAO sono le classi che interrogano il database e che mappano ogni business object su una tabella.
3. DTO (Data Transfer Object): è un pattern che utilizziamo per trasportare i dati alla view. Nel nostro caso dato che il progetto è semplice i DTO saranno una copia speculare delle classi del MODEL.
4. CONVERTER hanno il compito di convertire gli oggetti del MODEL in oggetti DTO.
5. SERVICE è un livello che interroga direttamente il livello DAO e richiama i converter.
6. CONTROLLER Utilizzano i services e gestiscono le chiamate alle view. I controller catturano gli input dell'utente e possono fare 2 cose:
    1. Effettuare un redirect ad un'altra view
    2. Chiamare i service per effettuare le operazioni CRUD sul DB
7. VIEW sono chiamate dai Controllers e hanno solo il compito di 'stampare a video' (in questa fase stamperemo i risultati su console in attesa di passare alla tecnologia successiva)
8. MAIN sono presenti le classi e i metodi per far partire l'applicazione.

NB: tutti i parametri di configurazione del DB sono contenuti nel file config.properties

GIT

GIT è un software di controllo di versione che utilizziamo per sviluppare un'applicazione in team. Attraverso i comandi di base `git pull` e `git push` modifichiamo il nostro progetto.

CLONARE IL PROGETTO

- creare una nuova cartella sul desktop con il nome del progetto
- testo destro nella cartella e selezionare "Git Bash Here" per aprire la riga di comando
- `git clone [URLGit]`
- `cd [NomeCartella]` (tasto TAB per la cartella principale)
- `git checkout [NomeBranch]`

PROCEDURA SCAMBIO DATI

Invio modifiche al repository remoto:

- `git status` (per controllare i file modificati e ti segnala i file che vanno in merge)
- `git add .`
- `git commit -m "[MessaggioDiEtichetta]"`
- `git push`

`Aggiornamento del repository locale:`

- `git pull`

MERGE

Per evitare conflitti bisogna lavorare su file diversi e poi fare push. Se c'è la necessità di modificare lo stesso file bisogna aspettare la modifica del tuo collega, acquisire la sua versione pullando e successivamente modificare.

Per risolvere i conflitti utilizzare il comando `git checkout [PercorsoNomeFileInConflitto]` questo comando cancella le tue modifiche e ti permette di fare pull. Ovviamente prima del checkout si consiglia di fare una copia del file su un blocco note per non perdere il lavoro fatto e inserirlo nuovamente su Eclipse dopo aver fatto pull.

REGOLE GENERALI

- Prima di fare push fare sempre il pull

- Fare il push soltanto se nel tuo codice NON sono presenti errori: il progetto deve rimanere funzionante altrimenti gli altri membri del team, quando faranno il pull, avranno anche loro un progetto non funzionante.