

Tecnologie di Sviluppo per il Web

Programmazione su Basi di Dati: Aspetti Metodologici Parte a

versione 3.1

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

G. Mecca – Università della Basilicata – mecca@unibas.it



Programmazione su BD: Aspetti Metodologici >> Sommario



Sommario

- Il Processo di Sviluppo
- Persistenza e DAO
 - ⇒ Operazioni CRUD
 - ⇒ Creazione degli Identificatori
 - ⇒ Gestione delle Connessioni



Il Processo di Sviluppo

- Processo di sviluppo visto finora
 - ⇒ I passo: analisi delle specifiche >> casi d'uso
 - ⇒ II passo: modello concettuale
 - ⇒ III passo: scelta dei componenti >> attribuzione delle responsabilità
 - ⇒ IV passo: sviluppo dei componenti (scrittura dei metodi)



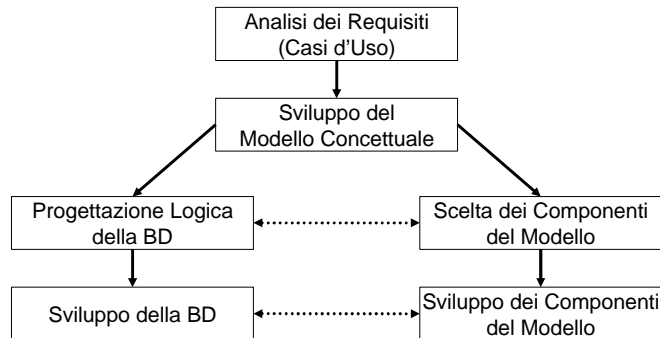
Il Processo di Sviluppo

- In un'applicazione client-server
 - ⇒ la differenza fondamentale è legata al cambiamento nella tecnologia della persistenza
- In particolare
 - ⇒ è necessario sviluppare la base di dati
 - ⇒ cambia lo strato di persistenza
 - ⇒ i componenti del modello devono riflettere i dati persistenti nella base di dati



Il Processo di Sviluppo

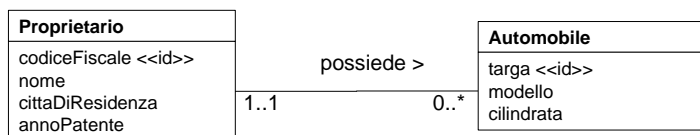
○ Come cambia il processo di sviluppo



Il Processo di Sviluppo

○ Nel nostro esempio

⇒ il modello concettuale porta ad individuare due classi in associazione





Il Processo di Sviluppo

o La progettazione logica

- ⇒ genera la seguente base di dati
- ⇒ l'associazione è tradotta dalla chiave esterna "proprietario" nella tabella automobili

```

create table proprietari (
  codiceFiscale char(16)
    not null primary key,
  nome varchar(50) not null,
  cittaDiResidenza varchar(50),
  annoPatente integer
)

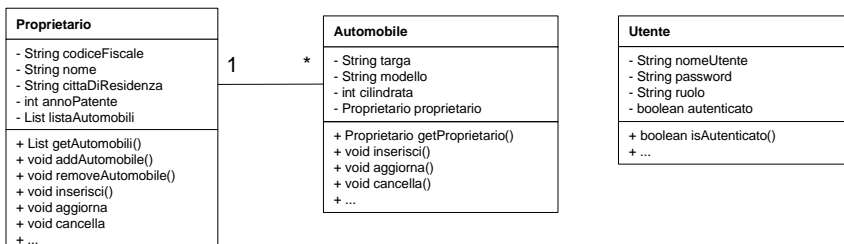
create table automobili (
  targa char(7) not null
    primary key,
  modello varchar(50),
  cilindrata integer,
  proprietario char(16)
    not null references
    Proprietari(codiceFiscale)
)
    
```



Il Processo di Sviluppo

o La scelta dei componenti

- ⇒ due classi con una associazione bidirezionale che discendono dai concetti
- ⇒ la classe Utente per l'autenticazione





Il Processo di Sviluppo

ATTENZIONE
alla novità

- **Attenzione alla novità nella persistenza**
 - ⇒ rispetto alle tecnologie di persistenza viste finora (file a formato libero ed XML)
 - ⇒ il processo era quello tipico dell'elaborazione orientata ai file
 - ⇒ caricamento dell'intero archivio in memoria
 - ⇒ elaborazione in memoria centrale
 - ⇒ salvataggio dell'intero archivio su disco



Il Processo di Sviluppo

- **Viceversa**
 - ⇒ nel caso del DBMS non è ragionevole caricare l'intero archivio in memoria centrale (operazione da evitare per le dimensioni)
 - ⇒ di conseguenza c'è un fenomeno di duplicazione dei dati
 - ⇒ i dati persistenti sono nella base di dati
 - ⇒ gli oggetti in memoria rappresentano di volta in volta una copia di una porzione della bd



Il Processo di Sviluppo

- Componenti del modello
 - ⇒ possiamo definirli come “una vista orientata agli oggetti” delle ennuple della base di dati”
- In cui, grossolanamente
 - ⇒ ogni classe è associabile ad una tabella
 - ⇒ ed ogni oggetto della classe rappresenta una ennupla della tabella



Il Processo di Sviluppo

- Il flusso tipico di lavoro in un caso d'uso
 - ⇒ alcune ennuple vengono caricate dalla base di dati
 - ⇒ vengono creati oggetti per rappresentarle
 - ⇒ vengono inizializzati opportunamente i riferimenti tra questi oggetti
 - ⇒ gli oggetti vengono manipolati dal controllo
 - ⇒ le ennuple vengono eventualmente aggiornate nella base di dati per riflettere lo stato degli oggetti



Il Processo di Sviluppo

○ Terminologia

- ⇒ i componenti della persistenza sono detti “Data Access Object” (DAO) – hanno il compito di effettuare tutti gli accessi alla bd
- ⇒ i componenti del modello sono detti “Data Transfer Object” (DTO) – hanno il compito di “trasportare” i dati caricati dalla base di dati fino al controllo e poi di nuovo fino alla base di dati



Persistenza e DAO

○ Package della persistenza

- ⇒ un DAO per ogni tabella della base di dati
- ⇒ eventuali DAO aggiuntivi per effettuare operazioni non specifiche di una tabella particolare (es: interrogazioni o statistiche che coinvolgono più tabelle)
- ⇒ componenti specializzati (DataSource e DAOException)



Persistenza e DAO

○ DAO

- ⇒ componente statico specializzato in operazioni sulla persistenza
- ⇒ tipicamente associato ad una tabella

○ Nel caso di una base di dati relazionale

- ⇒ ciascun DAO implementa le quattro operazioni fondamentali sulla tabella
- ⇒ cosiddette operazioni “CRUD”



Operazioni CRUD

○ Operazioni CRUD su una tabella

- ⇒ “Create”: inserimento di una ennupla
- ⇒ “Update”: aggiornamento di una ennupla
- ⇒ “Delete”: eliminazione di una ennupla
- ⇒ “Retrieve”: ricerca di ennuple nella tabella sulla base di un criterio di selezione
- ⇒ le operazioni di tipo “retrieve” possono essere varie; es: ricerca per codice fiscale, ricerca per nome



Operazioni CRUD

○ Esempio: DAOProprietario

- ⇒ public void doInsert(Proprietario p)
- ⇒ public void doUpdate(Proprietario p)
- ⇒ public void doDelete(Proprietario p)
- ⇒ public Proprietario doSelectCodF(String cf)
- ⇒ public List doSelectNome(String nome)

>> DAOProprietario
>> DAOAutomobile



Operazioni CRUD

○ Struttura tipica del caso d'uso

- ⇒ esempio: inserimento di un proprietario
- ⇒ il controllo acquisisce attraverso la vista i dati di un nuovo proprietario
- ⇒ il controllo crea un oggetto della classe Proprietario e ne imposta le proprietà
- ⇒ il controllo chiede al DAOProprietario di effettuare l'inserimento



Operazioni CRUD

>> DAOProprietario.java

- Un'annotazione sul metodo doUpdate()
 - ⇒ il metodo doUpdate() ha una struttura standard
 - ⇒ aggiorna tutti gli attributi della ennupla, ma non aggiorna il valore della chiave primaria (per evitare problemi di integrità)
 - ⇒ riscrive anche i valori degli attributi che non sono cambiati, perchè dal punto di vista del costo uno o tutti è indifferente



Operazioni CRUD

>> DAOProprietario.java

>> controllo

- DAOException
 - ⇒ eccezione lanciata dai DAO ogni volta che ci sono problemi sulla persistenza
- Comportamento tipico
 - ⇒ i DAO effettuano operazioni JDBC
 - ⇒ catturano l'eventuale SQLException
 - ⇒ effettuano le operazioni di chiusura
 - ⇒ rilanciano DAOException al controllo in modo che possa prendere le decisioni necessarie



Operazioni CRUD

○ Regole di comunicazione

- ⇒ i componenti di controllo creano e manipolano i componenti del modello (DTO)
- ⇒ i componenti di controllo usano i componenti di persistenza (DAO) per effettuare operazioni sulla persistenza
- ⇒ i componenti del modello non conoscono i DAO
- ⇒ i componenti del controllo sono isolati rispetto alla tecnologia della persistenza (es: tipo di DBMS)



Operazioni CRUD

○ Nota

- ⇒ una variante è quella in cui modello e persistenza sono accoppiati
- ⇒ in questo caso i metodi CRUD sono riportati all'interno del bean corrispondente
- ⇒ tipicamente le operazioni di aggiornamento sono realizzate attraverso metodi di oggetto
- ⇒ le operazioni di ricerca attraverso metodi statici; intuizione: responsabilità naturale per la classe



Operazioni CRUD

○ Esempio

- ⇒ con questa architettura Proprietario.java avrebbe i seguenti metodi aggiuntivi
- ⇒ public void doInsert()
- ⇒ public void doUpdate()
- ⇒ public void doDelete()
- ⇒ public static Proprietario doSelectCodF(String cf)
- ⇒ public static List doSelectNome(String nome)



Creazione degli Identificatori

○ Un aspetto particolare dell'esempio

- ⇒ tutte le tabelle hanno identificatori naturali
- ⇒ non è necessario introdurre identificatori sintetici
- ⇒ questo rende relativamente semplice definire le chiavi primarie e le chiavi esterne

○ Ma...

- ⇒ non sempre è così



Creazione degli Identificatori

○ In alcuni casi

- ⇒ le classi individuate in sede di modellazione concettuale non hanno identificatori naturali
- ⇒ è necessario introdurre nelle tabelle identificatori sintetici che diventano chiavi primarie
- ⇒ tipicamente numeri interi per ragioni di prestazioni
- ⇒ es: risultato di una partita di calcio



Creazione degli Identificatori

○ Attenzione

- ⇒ in questo caso, l'utente tipicamente NON è in grado di fornire all'applicazione un valore corretto per la chiave
- ⇒ fornisce solo i valori impliciti per la ennuola
- ⇒ è il sistema che, in forma automatica, deve generare l'identificatore da attribuire alla nuova ennuola prima dell'inserimento
- ⇒ es: partita di calcio



Creazione degli Identificatori

- Soluzioni a questo problema
 - ⇒ ce ne sono varie, più o meno complesse
 - ⇒ l'ideale sarebbe una tecnica sufficientemente generale da essere sempre applicabile e indipendente dal DBMS (>>)
 - ⇒ una soluzione più semplice: utilizzare attributi di tipo intero autoincrementante del DBMS



Creazione degli Identificatori

- Esempio: PostgreSQL
 - ⇒ il tipo SERIAL
 - ⇒ dichiara che un attributo di una ennupla ha valori interi
 - ⇒ ad ogni nuova ennupla inserita viene attribuito automaticamente un valore nuovo (quello immediatamente successivo a quello dell'ultima ennupla inserita)



Creazione degli Identificatori

○ Esempio: partita di calcio

```
create table partita (
  id serial not null primary key,
  data date not null,
  squadracasa varchar(20) not null
      references squadre(nome),
  squadratrasferta varchar(20) not null
      references squadre(nome),
  goalcasa integer,
  goaltrasferta integer
)
```



Creazione degli Identificatori

○ Nel DAO

⇒ l'operazione di inserimento deve essere fatta inserendo tutti gli attributi tranne il valore seriale, che viene generato automaticamente dal DBMS

○ Esempio

⇒ insert into partite(data, squadracasa, squadratrasferta, goalcasa, goaltrasferta) values ('2002-10-01', 'Milan', 'Lazio', 0, 0);



Gestione delle Connessioni

- Nello strato della persistenza >> DataSource
 - ⇒ un componente molto importante
- DataSource
 - ⇒ componente che si incarica di registrare il driver e fornire connessioni a tutti i DAO
 - ⇒ fornisce metodi per la chiusura
 - ⇒ obiettivo: localizzare la modalità di creazione delle connessioni (consentendo, ad esempio, di usare driver e DBMS diversi)



Gestione delle Connessioni

- >> ODBC
- >> aciBase su Access
- Vantaggio
 - ⇒ è possibile facilmente cambiare il driver e il DBMS su cui effettuare le operazioni
- Esempio: aciBase su Access
 - ⇒ utilizzo il driver di tipo 1
 - ⇒ devo registrare la base di dati Access (automobili.mdb) come sorgente ODBC
 - ⇒ Pannello di Controllo >> Strumenti di Amministrazione >> Origine Dati (ODBC)



Gestione delle Connessioni

○ Creazione di una connessione

- ⇒ si tratta di gran lunga dell'operazione più costosa e rischiosa
- ⇒ complesso scambio tra client e server (autenticazione)
- ⇒ inoltre le connessioni sono tipicamente una risorsa limitata
- ⇒ infine, i DBMS tendono ad essere sensibili alle connessioni "appese"



Gestione delle Connessioni

○ Linea guida fondamentale

- ⇒ accertarsi di manipolare correttamente le connessioni e gli oggetti collegati (ResultSet, Statement)

○ Ciclo di vita

- ⇒ la connessione viene aperta dalla DataSource
- ⇒ viene utilizzata dal DAO
- ⇒ la connessione e gli oggetti collegati vengono chiusi
- ⇒ sezione try-catch-finally



Gestione delle Connessioni

- Altra linea guida
 - ⇒ limitare il numero di connessioni verso il DBMS
 - ⇒ nel nostro esempio, per semplicità una connessione per ciascuna nuova operazione (chiaramente esagerato)
- Perché non un'unica connessione ?
 - ⇒ le connessioni scadono
 - ⇒ le operazioni sulla stessa connessione "interferiscono" tra di loro



Gestione delle Connessioni

- La soluzione corretta
 - ⇒ utilizzare un "pool di connessioni"
- Gestore del pool di connessioni
 - ⇒ componente che può essere utilizzato dalla DataSource
 - ⇒ inizializza un numero fissato di connessioni con il DBMS
 - ⇒ le fornisce ai metodi che le richiedono riutilizzandole (>>)



Riassumendo

- Il Processo di Sviluppo
- Persistenza e DAO
 - ⇒ Operazioni CRUD
 - ⇒ Creazione degli Identificatori
 - ⇒ Gestione delle Connessioni



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.