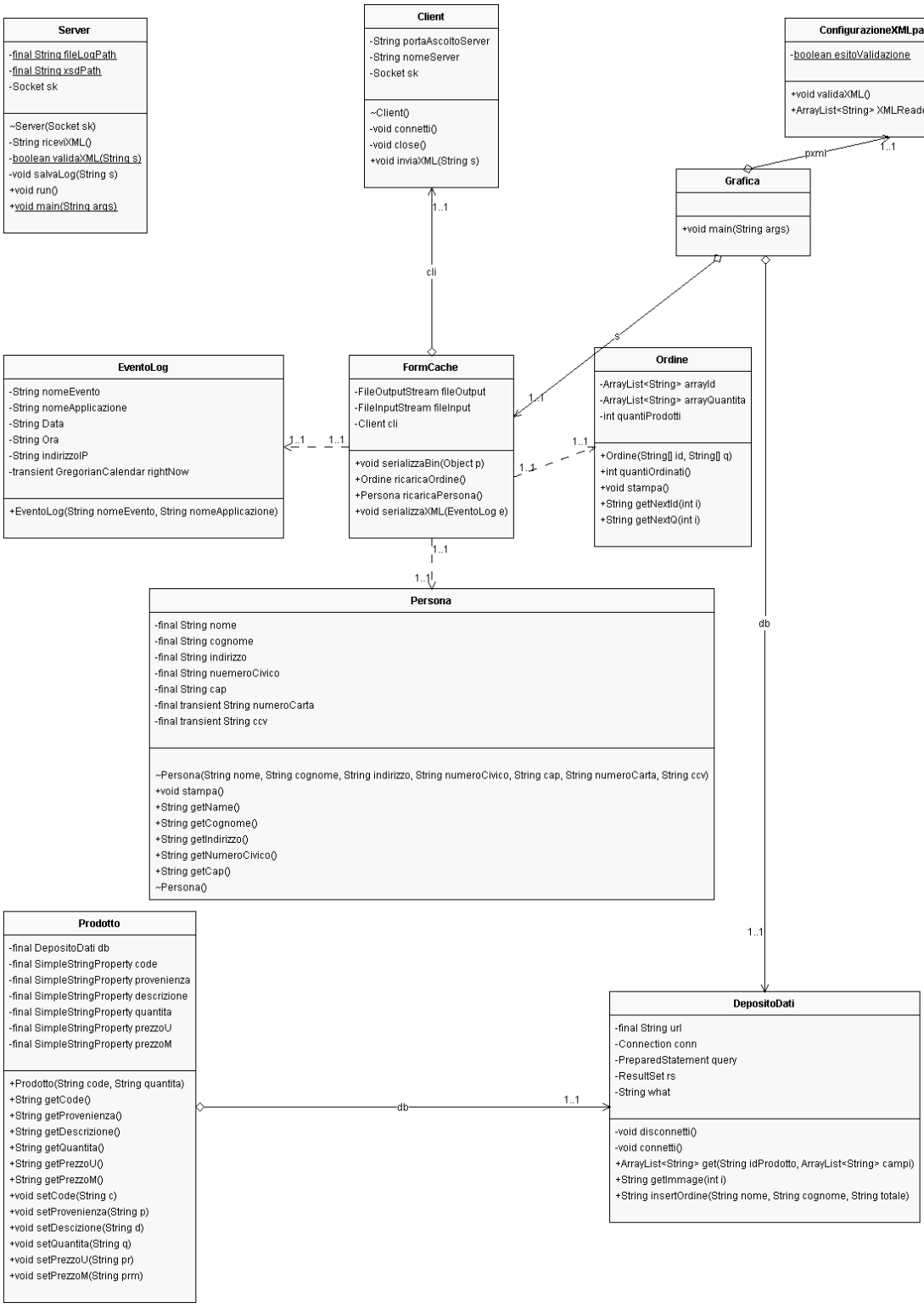
Spesa Online

progetto di spesa Online

Lorenzo Chelini - 15 febbraio 2015



Introduzione

L'applicazione consiste in una semplice interfaccia grafica che permette, sotto un opportuno caso d'uso ad un utente di fare acquisti online.

La classe principale è la classe Grafica, che si occupa dell'intrecciamento con l'utente e presenta i seguenti metodi principali:
private void setab() : imposta la TableView che conterrà i vari prodotti ordinati dall'utente (il numero massimo di prodotti è impostabile mediante il file di configurazione in xml), in particolare per ogni prodotto ne verranno mostrati nelle relative colonne della tabella il codice, la provenienza, una semplice descrizione, la quantità ordinata, il prezzo del singolo prodotto e il prezzo multiplo.
La tabella prevede due semplici bottoni, il bottone addButton che permette all'utente di inserire un nuovo prodotto al suo carrello e il bottone rmButton che svuota l'intero carrello.

private void setboxutente() : crea un oggetto Gridpane che contiene al suo interno le label e le Textfield relative ai dati immessi dall'utente, ovvero si prevede che l'utente inserisca nell'ordine il suo nome, il suo cognome, il suo indirizzo, il numero civico, il cap, il numero della carta di credito e il ccv.
Nella Gridpane sono inseriti anche delle TextField read-Only utilizzate dal sistema per mostrare all'utente il subtotale, il totale e le tasse applicate, queste ultime configurabili mediante il file di configurazione in xml.
Lo sfondo e il padding del Gridpane sono configurabili mediante il file di configurazione in xml.

private void setbBox() : crea un oggetto Hbox contenete due bottoni e imposta l'handler di risposta all'evento onClick e lo stile, i bottoni sono rispettivamente:
Il bottone bConferma che in risposta all'evento onClick inserisce l'ordine nel DB e invia l'evento al server di log.
Il bottone bAnnulla che svuota la form utente e invia l'evento al server di log.

private void setImmageBox() : crea un oggetto HBox contenete un rispettivamente un bottone di scorrimento, un VBox contenete un ImmageView e un TextField read-only, un secondo bottone di scorrimento.
Il metodo imposta anche i rispettivi handler per i rispettivi bottoni in seguito all'evento onClick in particolare per il bottone next l'handler avanza() per il bottone pre l'handler indietro().
Per la corretta visualizzazione dell'immagine in seguito alla pressione del bottone next o del bottone pre il sistema tiene traccia di una variabile globale, scorrimento, che punta sempre all'immagine attuale, in particolare ad esempio il metodo avanza(handler del bottone next) incrementerà scorrimento, se possibile, e successivamente verrà prelevato

dal db l'immagine con indice scorrimento, precisiamo che ciò che effettivamente viene prelevato dal db non è l'immagine ma il suo url.
Discorsi analoghi valgono per l'handler del bottone pre.

Classe DepositaDati ha il compito di interfacciarsi con il DB, i metodi principali sono i seguenti:
private void connetti() : apre una connessione verso il DB

public ArrayList<String> get(String idProdotto, ArrayList<String>campi) : interroga il DB con lo scopo di farsi restituire i campi, richiesti attraverso il parametro formale ArrayList<String>campi, del prodotto identificato dal parametro formale idProdotto. Il risultato della query è inserito in un array dinamico di tipo lista e restituito al chiamante.
Tabella coinvolta prodotti.

public String getImmage(int i) : interroga il db con lo scopo di farsi restituire l'url sotto forma di stringa relativo all'immagine di indice i.
Tabella coinvolta immagini.

public String insertOrdine(String nome,String Cognome, String totale) : Inserisce all'interno del DB una nuova tupla del tipo (Mario,Rossi,100) che rappresenta il totale dell'ordine appena effettuato da Mario Rossi e restituisce al chiamante una stringa che rappresenta l'dentificavito dell'ordine.
Tabella coinvolta ordine.

private void close() : chiude la connessione con il DB.

Classe FormCache ha il compito di gestire la cache dell'applicazione in particolare alla chiusura il salvataggio di alcune informazioni su file binario e all'apertura il ripristino di quest'ultime.
I metodi principali sono i seguenti:
public void serializzaBin(Object p) throws ClassNotFoundException : Dato in ingresso un Object verifica se questo è un istanza della classe persona oppure della classe ordine, e in base all'esito di tale verifica scrive su file binario persona.bin se si tratta di una persona oppure su file ordine.bin se si tratta di un ordine.

public Ordine ricaricaOrdine() throws ClassNotFoundException : Legge da file binario ordine.bin, se c'è qualcosa da leggere e in tal caso restituisce al chiamante un oggetto di tipo Ordine.

public Persona ricaricaPersona() throws ClassNotFoundException : Legge da file binario persona.bin, se c'è qualcosa da leggere e in tal caso restituisce al chiamante un oggetto di tipo Persona.

public void serializzaXML(EventoLog e) : Dato in ingresso un istanza di un oggetto EventoLog attraverso XStream lo converte in stringa xml e lo invia al server di log.

Classe ConfigurazioneXMLParametri ha il compito di validare all'avvio dell'applicazione il file di configurazione dei parametri attraverso un opportuno schema e quello di leggere le varie configurazioni(economiche,stilistiche e tecnologiche).
I metodi principali sono i seguenti:
public void validaXML() : Valida il file di configurazione parametri.xml attraverso un opportuno schema di validazione schemaParametri.xsd e setta una variabile statica privata della classe che rappresenta l'esito della validazione.

public ArrayList<String> XMLReader(String what) : metodo che si occupa di restituire i valori di configurazione al chiamante.
In dettaglio in base al valore del parametro formale viene selezionato il macro-nodo(economici,stilistici e tecnologici) e vengono catturati i nodi figlio, per ognuno di questi se ne prende il valore e il nome e si inseriscono in un array dinamico di tipo string che verrà restituito al chiamante.
Ad esempio se il metodo viene chiamato sul macro-nodo tecnologici ciò che viene restituito è il seguente : ipServer - LocalHost - portaServer - "1234".

Classe Client ha il compito di interfacciarsi con il server di log.
I metodi principali sono i seguenti:
costruttore : che legge da file di configurazione parametri.xml indirizzo ip e porta di ascolto del server di log.

public void connetti() : apre una connessione verso il server di log.

public void inviaXML(String s) : invia ls stringa di log al server.

Classe Persona,Ordine e EventoLog classi secondarie che implementano Serializable gli oggetto verranno infatti trasformati in uno stream di byte, in particolare un istanza della classe persona e della classe ordine sarà salvata su file binario alla chiusura dell'applicazione e riletta alla riapertura della stessa.
Un istanza della classe EventoLog sarà invece serializzata e inviata al file di log.
La classe Prodotto è legata alla TableView.