PROGETTO DI PROGRAMMAZIONE E MODELLAZIONE AD OGGETTI

Tabacchi

Nicholas Latini Matricola: 292905
28/06/2021

SPECIFICA DEL SOFTWARE

Il software intende simulare l'interfaccia di un tabacchi che deve seguire ogni step dell'utente, ovvero: selezione del prodotto da acquistare, selezione della quantità del prodotto da acquistare e pagamento. La macchina comunica con un database su file, che viene utilizzato per ricaricare la macchina con i prodotti e aggiornare la quantità dei prodotti una volta acquistati. La stampa dello scontrino serve solo a tenere traccia dei prodotti acquistati.

STUDIO DEL PROBLEMA

PUNTI CRITICI

Il punto critico del software viene rilevato nell'interfaccia di pagamento del prodotto. In questa interfaccia viene utilizzato un timer che allo scadere restituisce i soldi inseriti e fa tornare l'utente alla schermata principale. Il problema è che se si attendono un po' di secondi, dall'uscita dell'interfaccia,il MessaggeBox che comunica che stiamo tornando alla schermata principale comparirà due volte.

Per cercare di risolvere il problema ho inserito l'istruzione che stoppa il timer e lo disabilita, sia nel metodo dell'evento del tick del timer sia quando l'utente inserisce tutti i soldi. Nonostante questo non sono riuscito a risolvere il problema poiché è come se il tempo del timer passi comunque nonostante la scermata sia chiusa e il timer disattivato.

SCELTE ARCHITETTURALI

Innanzi tutto il progetto è stato suddiviso in cartelle per ogni classe tranne quelle principali come la classe macchina, così da rendere il progetto più leggibile e rendere più semplice la ricerca delle cartelle in cui inserire in futuro nuove classi. Le cartelle che compongono il progetto sono:

- Macchina;
- Carrello;
- Pagamento;
- Database;
- Scontrino;
- Visitors
- Prodotto

Ognuna di queste cartelle contiene la sua classe di riferimento.
La classe macchina, carrello e pagamento estendono la classe Form, ovvero delle finestre che al loro interno hanno degli elementi che dovranno interagire con l'utente, come per esempio i bottoni di scelta dei prodotti, per la scelta della quantità, per il pagamento, per la conferma o per l'annullamento dell'operazione.
La cartella database contiene l'interfaccia del database che comunica con la macchina e le due classi che contengono le operazioni di lettura e scrittura su file per aggiornare i due database utilizzati, uno per controllare il resto disponibile all'interno della macchina e l'altro per salvare lo stato dei prodotti ad ogni interazione con la macchina.

Classe Macchina

La classe macchina si occupa di interagire con il database dei prodotti per prelevarli e poi aggiornarne la quantità se l'operazione di pagamento è andata a buon fine e aggiornarne l'immagine visualizzata a schermo nel caso finiscano. Ad ogni click del prodotto selezionato crea un carrello che sarà istanziato con il prodotto selezionato. La macchina utilizza il metodo resetProdottiFiniti() per ricaricare tutti i prodotti nel db che erano finiti, questo succede ogni volta che si riavvia il programma.

Classe Carrello

La classe carrello si occupa di tenere traccia del prodotto selezionato, mostra a video immagine e prezzo del prodotto. Questa interfaccia consente di selezionare tramite dei pulsanti la quantità del prodotto che si vuole acquistare e si controlla che non superi quella disponibile nello storage della macchina. La classe macchina utilizza un metodo di questa classe per permettergli di prelevare il prodotto se l'acquisto è andato a buon fine (e lo si verifica tramite la variabile booleana statoOperazione) per poter aggiornare la quantità dei prodotti nel db. In questa classe viene utilizzato anche un Visitor, che è utilizzato per visualizzare il prezzo totale da pagare.

Classe Pagamento

La classe pagamento di occupa di gestire l'interfaccia e le operazioni di pagamento, ovvero l'inserimento dei soldi, di controllare che l'utente inserisca tutti i soldi e che lo faccia entro il tempo limite. Quando l'operazione è andata a buon fine, ovvero che i soldi sono stati inseriti tutti correttamente viene stampato uno "scontrino".

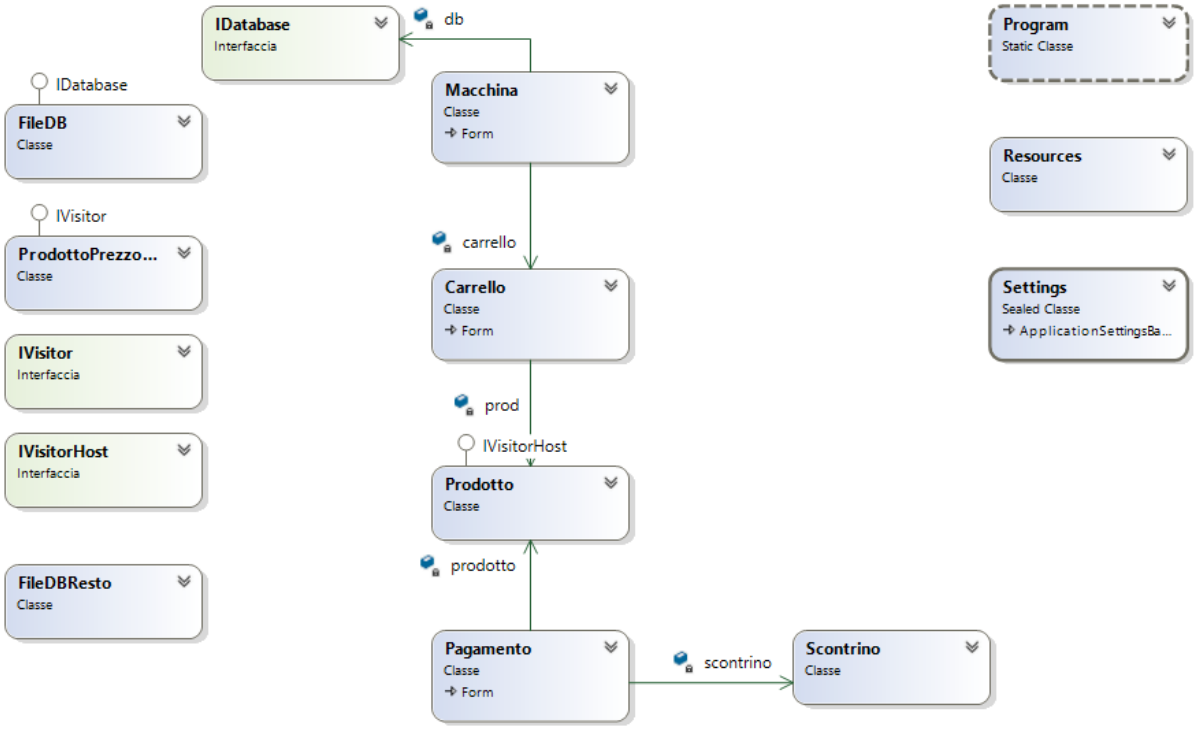
Classe Scontrino

La classe scontrino si occupa di stampare su file i prodotti che son stati acquistati finora e ci indica se dobbiamo ricevere il resto o lo abbiamo ricevuto.

Classe Prodotto

La classe prodotto estende l'interfaccia IVisitorHost per implementare il metodo che consente ai visitor di accedere alla classe. La classe prodotto contiene solo le caratteristiche o informazioni dei singoli prodotti, e i get/set di queste.

UML


IDatabase
Interfaccia
db
Program
Static Classe
IDatabase
FileDB
Classe
Macchina
Classe
Form
Resources
Classe
IVisitor
ProdottoPrezzo...
Classe
carrello
Carrello
Classe
Form
Settings
Sealed Classe
ApplicationSettingsBa...
IVisitor
Interfaccia
prod
IVisitorHost
IVisitorHost
Interfaccia
Prodotto
Classe
FileDBResto
Classe
prodotto
Pagamento
Classe
Form
scontrino
Scontrino
Classe

DESCRIZIONE DEI DESIGN PATTERN UTILIZZATI

Sono stati utilizzati due design pattern:

- Singleton pattern
- Visitor Pattern

Il singleton pattern viene utilizzato nelle due classi che si occupano di interagire con i db. Questo pattern ci garantisce di creare un'unica istanza della classe offrendo quindi un unico punto di accesso alla classe. In poche parole, quando richiameremo il metodo statico GetIstance() da qualche parte nel codice, esso ci restituirà l'istanza della classe se era già stata creata oppure ne crea una nuova se non ne era stata creata nessuna in precedenza.

Il visitor pattern viene utilizzato per visitare il prodotto, e viene utilizzato nella classe carrello per aggiornare il prezzo totale ogni volta che si aumenta la quantità del prodotto da acquistare. In pratica il visitor del prezzoTotale accede al prodotto tramite il metodo Visit() e si occuperà di effettuare la moltiplicazione di (prezzo prodotto * quantità) così da trovare il prezzo totale.

USE CASES