



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

CORSO DI LAUREA IN
INFORMATICA APPLICATA
SCUOLA DI
SCIENZE TECNOLOGIE E FILOSOFIA DELL'INFORMAZIONE

Programmazione e Modellazione a Oggetti – *Progetto a.a. 2020/2021*



“ProGest - Gestionale”

Sviluppatori:

Simone Cossi, mat. 290796

INDICE

- 1) Specifica del Software
- 2) Studio del Problema
- 3) Scelte Architettureali
- 4) Documentazione sull'utilizzo
- 5) Use Cases con UML

PRIMA PARTE – Specifica del software

Sviluppo dell'app "*ProGest*" secondo il paradigma di OOP, utilizzando linguaggio C# con il Framework "Windows Form" per facilitare lo sviluppo del Front-End e un Database SQL per quanto riguarda l'archiviazione delle informazioni.

L'app permette di tenere traccia di varie informazioni riguardanti un'azienda vitivinicola (per la simulazione ho deciso di prendere d'esempio l'azienda di famiglia).

L'app permette di inserire, modificare e rimuovere informazioni riguardanti elementi dell'azienda.

SECONDA PARTE – Studio del problema

PUNTI CRITICI:

- 1) Archiviazione e gestione dei dati
- 2) Grafica
- 3) Complessità generale dell'applicazione e quantità di informazioni inseribili

SOLUZIONE DEI PUNTI CRITICI:

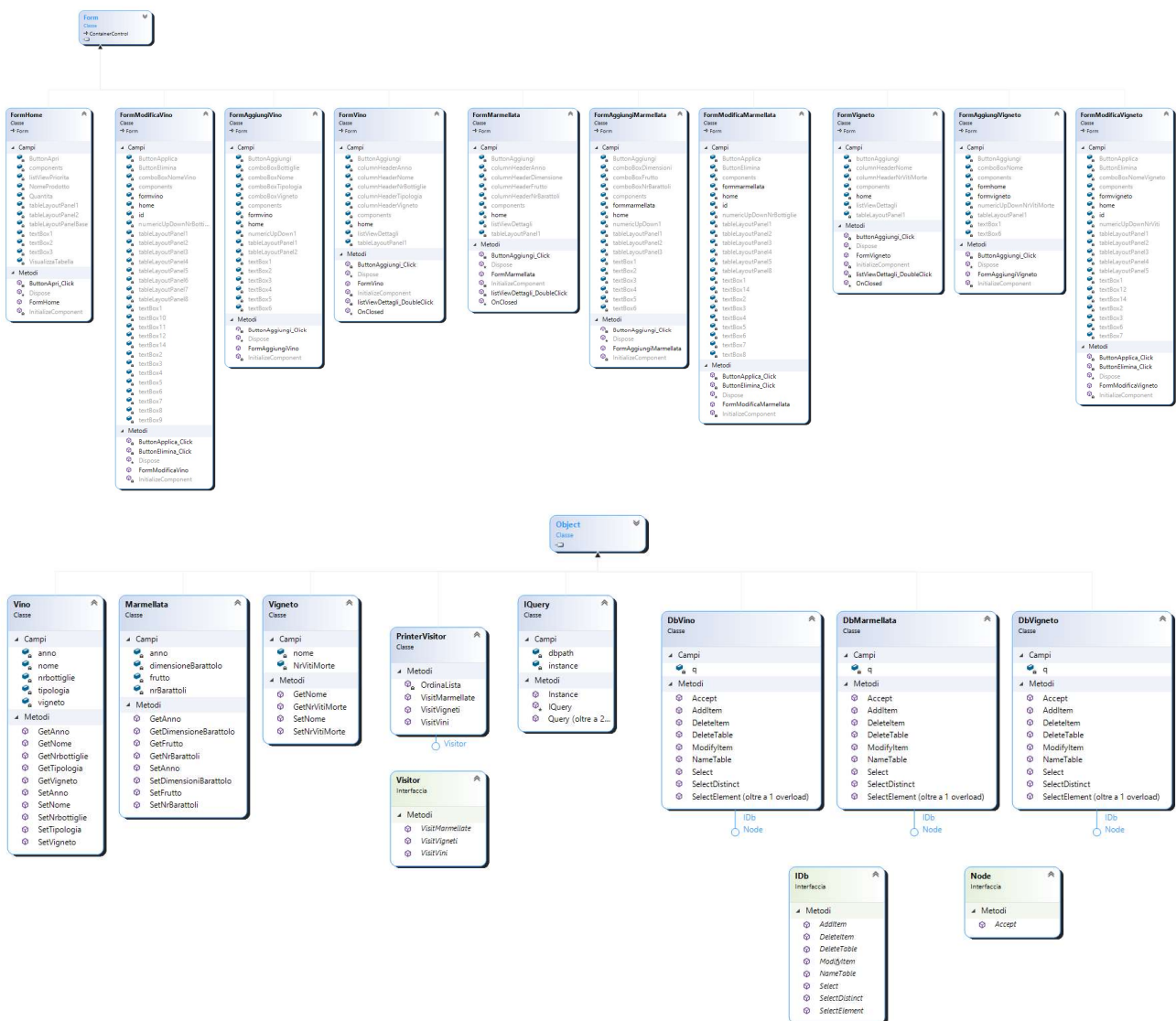
- 1) Utilizzo di un Database SQL locale implementato grazie alla libreria SQLite
- 2) Utilizzo del “Framework Windows Form” che permette una facile implementazione e sviluppo della parte grafica che permette di testare le funzionalità dell'app
- 3) La creazione di un app gestionale non è sicuramente un lavoro semplice a causa della vastità di informazioni che una reale azienda deve gestire. Perciò ho deciso di soffermarmi di più sui prodotti dell'azienda di famiglia con giusto un collegamento tra l'origine di un prodotto e il prodotto stesso.

TERZA PARTE - Scelte Architettrurali

L'applicazione ha una netta distinzione tra le classi che gestiscono la parte grafica e le classi che sono alla base del reale funzionamento dell'app: archiviazione, gestione ed elaborazione delle informazioni.

Pertanto, analizzando l'UML del progetto, è possibile notare questa netta distinzione:

le classi che si occupano della parte grafica che derivano dalla classe superiore "Form" mentre le altre classi derivano direttamente da "Object".



PUNTI CRUCIALI:

Creazione di un'interfaccia IDb: quest'interfaccia fornisce i vari metodi necessari per la comunicazione con un database. Quest'interfaccia nel progetto viene utilizzata da 3 classi che dialogano con il db locale. L'interfaccia è stata inserita nel progetto con lo scopo di interrogare un eventuale Database online in modo da poter accedere agli stessi dati da dispositivi diversi.

FACTORY METHOD PATTERN: per l'aggiunta di nuovi elementi all'interno del Database è risultato utile l'utilizzo del Factory Method, senza però l'utilizzo dell'interfaccia. Questo perché i vari oggetti non posseggono metodi e attributi in comune. Il pattern, anche se non completo, permette di creare degli oggetti di vario tipo da passare successivamente a un metodo, che effettuerà una query al Database, che aggiungerà l'elemento.

SINGLETON PATTERN: durante lo sviluppo delle classi che eseguono le query al Database si è riscontrata una grande ridondanza di righe di codice pressoché uguali, pertanto implementando il singleton si è andata a semplificare la struttura di tutte le varie classi e pure l'utilizzo poiché basta passare i parametri al metodo senza dover sempre modificare l'intero codice per i singoli parametri.

VISITOR PATTERN: Volendo inserire una lista nella Homepage contenente i vari elementi con maggiore priorità di vendita/risoluzione era necessario un metodo che andasse a prelevare i dati dal database, li elaborasse in base ai criteri scelti e li restituisse. Il Visitor Pattern applicato alle classi che comunicano con il db e alla classe "PrinterVisitor" recupera dal Database i dati li elabora e li restituisce in modo da poterli stampare a video.

Questo avviene in modo tale da non dover estendere tutti gli oggetti con metodi molto simili (simile al singleton) e da non dover aggiungere metodi non strettamente inerenti alle classi.

ESEMPIO: nel progetto il Visitor Pattern viene implementato nelle classi che comunicano con il Database, andare ad aggiungere il metodo che prende i dati, li elabora e li restituisce potrebbe causare confusione, perciò, si è preferito esternalarlo.

QUARTA PARTE – Documentazione sull'utilizzo

Il programma è abbastanza intuitivo e semplice, tuttavia, quando si apre una pagina e si visualizzano tutti gli elementi di una determinata categoria, per modificare un elemento è necessario fare un doppio click sul nome dell'elemento desiderato.

Per quanto riguarda i valori da inserire durante l'aggiunta o la modifica di un elemento, sono presenti dei consigli e se l'utente dovesse sbagliare l'inserimento di un campo nessun problema perché il programma non proseguirà fino al corretto inserimento dei dati.

QUINTA PARTE – Use Cases con UML

Caso d'uso: Caricamento Home Page
ID: UC1
Attore: A1 User, A2 Db Locale
Presupposti: <ul style="list-style-type: none">• App appena avviata o ritorno sulla Home Page
Corso base degli eventi <ol style="list-style-type: none">1) L'applicazione viene avviata da A12) Vengono Eseguite delle query a A23) Viene contattato A24) A2 ritorna i valori desiderati5) Vengono elaborati i valori dati da A26) Vengono Stampati a video i Dati7) A1 seleziona una tabella8) A1 preme su "Apri Tabella"
Post condizioni: <ul style="list-style-type: none">• Viene aperta una su 3 differenti schermate in base alla tabella scelta da A1
Percorsi alternativi
<ol style="list-style-type: none">1) A1 preme sulla X per terminare il programma2) Il programma Termina

Caso d'uso: Pagina di gestione di una Tabella
ID: UC2
Attore: A1 User, A2 Db Locale
Presupposti: <ul style="list-style-type: none"> A1 si sposta in questa pagina tramite il menu di selezione e il pulsante "Apri Tabella"
Corso Base Degli Eventi <ol style="list-style-type: none"> Viene eseguita una query a A2 A2 ritorna i dati I dati vengono stampati a video A1 preme sul pulsante "Aggiungi ***" Vengono effettuate delle query a A2 A2 ritorna i dati Vengono inseriti i dati ottenuti da A2 in dei menu di selezione A1 riempie tutti i campi selezionando dati già esistenti o inserendone di nuovi A1 preme su "Aggiungi" Viene eseguita una query a A2
Post condizioni <ul style="list-style-type: none"> Viene Aggiornato A2 Si torna alla pagina di gestione di una tabella
Percorsi alternativi <ol style="list-style-type: none"> A1 preme 2 volte sul nome di un elemento Vengono effettuate delle query a A2 A2 torna dei valori I valori ottenuti da A2 vengono stampati a Video A1 riempie o meno dei campi A1 preme su "Applica Modifiche" o su "Elimina Prodotto" Viene effettuata una query a A2 A2 viene aggiornato A1 preme sulla X e torna alla Home Page