

Studente: Scano William
Matricola: 293017

Corso di Programmazione e Modellazione a Oggetti
Progetto sessione Invernale 2020/2021

Docente: Prof. Saverio Delpriori

Specifica del software.

Specifica del software:

Si tratta di un programma C# in Windows Form per la gestione di un lido balneare composto da tre tipologie di offerte:

offerta base, offerta plus, offerta extra.

Ogni tipologia di offerta differisce dalle altre per il costo e per il numero di lettini/ombrelloni che offrono.

L'offerta base comprende: solo un lettino.

L'offerta plus comprende: l'ombrellone e due lettini.

L'offerta extra comprende: l'ombrellone e tre lettini.

Per ogni nuovo cliente il gestore deve inserire i suoi dati e verificherà se vi sono posizioni libere, in caso affermativo il cliente sarà salvato in un DB locale delle prenotazioni così da avere uno storico.

In caso negativo sarà salvato in una lista d'attesa.

E' possibile eliminare un qualsiasi cliente.

Studio del problema.

Studio del problema:

o Quali sono i punti critici?

- 1) Il programma deve poter memorizzare varie informazioni in locale.
- 2) Deve essere possibile fare prenotazioni per date future.
- 3) Riempire il database attraverso il numero di lettini e ombrelloni disponibili ogni giorno.
- 4) Un punto critico è stato il fatto che per la grafica ho scelto un tema scuro ma purtroppo il pulsante per il datetimepicker non consente di cambiare colore dalle proprietà.

o Come si sceglie di affrontarli?

- 4) Per il tema scuro ho dovuto creare una classe personalizzata ereditata dal datetimepicker e poi ho dovuto impostare tutti i valori che volevo andare a cambiare e ho fatto l'override del metodo per cambiare colore predefinito.
- 3) Il riempimento del database è stato effettuato tramite un metodo contenente un ciclo che ripete una query di inserimento 365 volte, pari ai giorni in un anno.
- 2) Le prenotazioni di date future sono risolte grazie ad una tabella del db contenente per ogni giorno il numero di lettini e ombrelloni giornalmente disponibili.
- 1) Il programma si appoggia ad un database locale sul quale vengono salvati i dati dei vari clienti e il numero di lettini e ombrelloni. Il programma è stato pensato per essere usato in quelle situazioni in cui la connessione è sempre disponibile, connection oriented.

Scelte architetturali.

Scelte architetturali:

o **Descrizione dell'architettura software comprensiva di uno schema delle classi UML che descriva le componenti principali del sistema.**

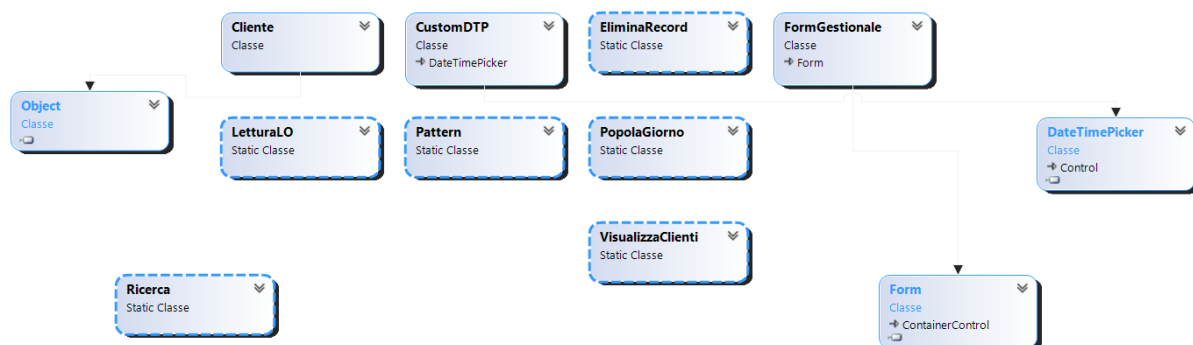
L'archiviazione delle informazioni avviene tramite query verso un database locale di tecnologia SQL Server.

Quest'ultimo prevede principalmente due tabelle, "Cliente" e "Giorno" che conterranno tutte le informazioni necessarie per il corretto funzionamento del programma.

L'applicazione lato client è invece composta da più classi, alcune con funzioni specifiche altre realizzate per mantenere la corretta struttura dei dati e rispettare i principi fondamentali della programmazione come l'incapsulamento.

Una classe importante è Cliente che contiene tutte le informazioni di un cliente, la sua istanza viene passata a run time a tutte le varie classi di utilità.

In "FormGestionale" sono presenti le chiamate dei metodi per la creazione delle offerte e la gestione della parte grafica (bottoni e textbox e controlli input).



o **Descrizione e motivazione dei design pattern utilizzati.**

Il pattern Factory è un pattern creazionale e definisce un'interfaccia per creare un oggetto, ma lascia le sottoclassi decidere quale classe istanziare. Permette a una classe di differire l'istanza da subclassare o consente di modificare il tipo di oggetto predefinito da creare. Factory elimina il bisogno di utilizzare specifiche per un'applicazione, infatti l'istanziamento di un oggetto non richiede una chiamata diretta al costruttore, il factory è delegato a farlo.

Il mio Product è "Offerta" cioè l'offerta generale da cui ereditano le altre, definisce l'interfaccia di oggetti che il factory method crea.

Il ConcreteProduct è una classe che implementa l'interfaccia Product e sono le tre offerte "OfferBase", "OfferPlus" e "OfferExtra" che vanno a fare i vari override dei metodi.

Il Creator è "OffertaFactory" ed è una classe astratta che dichiara il factory method.

I concrete Creator sono "OffertaBaseFactory", "OffertaPlusFactory" e "OffertaExtraFactory".

Queste classi che implementano la classe del Creator fanno l'override del factory method per ritornare un'istanza del ConcreteProduct.

Ho scelto il Factory rispetto al pattern Builder perché l'oggetto "offerta" non è un oggetto molto complicato e perché il suo costruttore prende in ingresso solamente due parametri (le date di inizio e fine prenotazione). Poi col pattern Builder avrei istanziato le offerte prima di quando mi sarebbero effettivamente servite.

Il pattern Observer si usa quando si ha un soggetto e tanti observer che dipendono dallo stato del soggetto. Questo tipo di interazione è detto public-subscribe, cioè pubblico senza sapere se qualcuno mi sta ascoltando.

Si applica quando un oggetto dovrebbe essere in grado di notificare gli altri oggetti senza sapere chi sono questi ultimi.

L'interfaccia ISubject contiene i metodi per collegare/scollegare e notificare.

L'interfaccia IObserver richiama il metodo per la notifica effettiva.

ProductSubject si occupa dell'implementazione dei metodi dell'interfaccia ISubject e crea una lista di Observer.

ProductObserver nel mio caso "ClientInAttesa" implementa il metodo per la notifica contenuto in IObserver.

Nel software, ho deciso di implementarlo in modo che, quando un cliente non può essere registrato perché nella data desiderata non vi è più posto, egli viene registrato in una lista di osservatori. Se per caso si dovesse liberare un posto per via di una prenotazione cancellata, il gestore verrà notificato di tutti i clienti che avevano mostrato interesse in una prenotazione per quella data e i contatti per informarsi se sono ancora interessati.

Documentazione sull'utilizzo.

Documentazione sull'utilizzo:

o (Se applicabile) Con quali parametri va eseguito il software, una volta compilato?

Il software, una volta compilato, si presenta con un'interfaccia grafica nella quale è possibile aggiungere una prenotazione inserendo tutti i parametri richiesti: nome, cognome e numero di telefono del cliente, la data di inizio e fine prenotazione e uno dei tre possibili tipi di offerta. Per il nome e cognome è possibile inserire solo lettere e non più lunghe di trenta caratteri. Il telefono accetterà solo numeri e non più di dieci.

Le date saranno selezionabili da un menù a tendina, chiaramente non è possibile fare prenotazioni nel passato e la data di inizio non può essere maggiore di quella di fine.

Se non si seleziona nessuna data viene applicata la data odierna.

Le offerte sono selezionabili da un menù a tendina, basta selezionare quella desiderata.

Se uno dei parametri non viene compilato non sarà possibile effettuare la prenotazione.

Inseriti tutti i parametri basta cliccare il pulsante "SALVA" per vedere apparire la prenotazione nella tabella sottostante.

E' possibile ricercare un cliente tramite Id univoco, se presente sarà filtrato nella tabella.

Una prenotazione può essere eliminata prima selezionandola e poi cliccando sul pulsante "ELIMINA" per cancellarla definitivamente dal record.

o (Se applicabile) Ci sono passi particolari da eseguire per la compilazione?

Per il primo avvio va specificato il percorso del database nella stringa di connessione "path" in FormGestionale.cs.

Non vi è un ordine preciso per l'inserimento dei valori al fine di aggiungere una prenotazione, l'importante è inserire tutti i valori richiesti.

Solo per eliminare una prenotazione bisogna prima selezionarla con un doppio click sull'id e poi con un click sul pulsante "ELIMINA" per cancellarla definitivamente dal record.

Use Cases con relativo schema UML.

o Descrivere i più significativi.

L'unico attore è il gestore, il quale può compiere principalmente tre azioni:

Inserire una prenotazione, cancellarla e ricercare un cliente tra quelli salvati.

L'azione di salvataggio si estende in due casi, il salvataggio su database e quello in lista.

Eliminare un cliente include la presenza di almeno una prenotazione sia già presente e il bisogno di selezionarla.

La ricerca può essere effettuata sempre anche a database privo di prenotazioni semplicemente non accadrà nulla.

