

# **Relazione sul progetto di Programmazione ad Oggetti a.a. 2015/2016**

## **Gestione Hotel**

Stefano Lia  
matricola: 1097641

# Ambiente di sviluppo

- Compilatore: GCC 4.9.1, 64 bit
- Versione Qt Creator: 5.5.1
- Sistema Operativo: Ubuntu 15.10

## Compilazione ed esecuzione

Attraverso la shell del Terminale posizionarsi all'interno della cartella "Freedom" ed eseguire il comando qmake. Una volta generato il Makefile, sempre all'interno della cartella "Freedom", avviare il comando make. Questa operazione genererà un file eseguibile Freedom che potrà essere avviato tramite un doppio click.

Nota: per i computer del laboratorio è necessario, prima di eseguire il comando qmake, digitare "qt-532.sh".

## 1. Introduzione

Il progetto è stato pensato per un ipotetico hotel del quale si vogliono gestire eventuali prenotazioni, dunque, il programma deve essere disponibile al gestore dell'hotel o a chiunque lavori per quest'ultimo.

L'hotel presenta 40 stanze:

piano 1) 10 Camere Singole (Monolocali)

piano 2) 10 Camere Doppie

piano 3) 6 Camere Doppie e 4 Suite

piano 4) 6 Suite e 4 Royal Suite

L'ipotetico "gestore" può vedere la struttura dell'hotel (ovvero come sono distribuite le stanze) e, nel caso, renderle inagibili, ossia non utilizzabili per un certo periodo. E' possibile, sempre da parte del gestore, effettuare delle prenotazioni, inserendo il nome (di chi prenota) e durata di quest'ultime. Si può poi accedere ad una QTableWidget che rappresenta tutte le prenotazioni che sono state fatte fino a quel momento. Queste informazioni vengono salvate rispettivamente in due files: "inputstanze.txt" e "outputPrenotazioni.txt" in modo da salvarle in maniera permanente (ovvero anche dopo la chiusura dell'applicazione).

Il progetto rispetta il modello della MV (Model View), cercando di tenere il più possibile distaccata la parte logica dalla parte grafica facilitando eventuali modifiche o estensioni del codice. Le classi dedicate alla parte grafica sono "Reserve" per effettuare le prenotazioni, "Reservations" per visualizzarle, "Struttura" che si occupa di illustrare la struttura dell'hotel, il tutto accessibile dalla pagina principale rappresentata dalla classe "MainWindow".

Nota: La struttura è stata "inventata" e non corrisponde a nessun hotel esistente.

## 2. Parte logica

Si è deciso di dotare il contenitore della “lista” dei puntatori smart poiché il progetto era inizialmente pensato con varie liste di prenotazioni a seconda della tipologia della stanza. Poteva, dunque, capitare di effettuare una prenotazione uguale per due stanze di tipologie diverse. Pertanto, se in futuro si vorrà dotare il programma di questa funzionalità, il puntatore Smart è già disponibile. Essendo, infine, smart il puntatore gestisce la memoria in maniera anche profonda.

E' presente una classe iteratore che può fungere sia da iteratore costante che non costante, in modo che sia possibile rispettivamente sia usarli in sola lettura che effettuare modifiche ai nodi della lista.

Prima di elencare la gerarchia, è utile dire che l'hotel è stato pensato come un vector di vector, ovvero a due dimensioni: la prima rappresentante i piani e la seconda le stanze. Esse vengono costruite subito all'apertura del programma e i campi dati vengono letti dal file “inputstanze.txt”.

### 2.1 Stanza

Stanza è la classe base della gerarchia. Essa è astratta perché fornisce 3 metodi virtuali puri:

- 1) PostiDisponibili() : ritorna il numero di posti che può avere una stanza.
- 2) CostoPrenotazione(): ritorna il prezzo che varia a seconda della stanza.
- 3) Nome(): ritorna il nome della stanza sotto forma di QString.

Essa è anche virtuale e, come tale, fornisce il distruttore virtuale per permettere la distruzione “polimorfa” degli oggetti stanza e derivati.

### 2.2 CameraDoppia

CameraDoppia rappresenta la classica stanza a due posti, più eventuale bambino. Si tratta di una classe concreta perché reimplementa i metodi virtuali di Stanza.

### 2.3 CameraSingola

CameraSingola rappresenta la stanza ad un posto ed è anch'essa una classe concreta.

### 2.4 Suite

Suite rappresenta una delle stanze di lusso che l'hotel possiede. E' una classe concreta e ha disponibili fino a 4 posti. E' disponibile anche un campo statico sconto, che fornisce il minimo di giorni necessari per attivare lo sconto su quella determinata stanza.

## 2.5 RoyalSuite

Questa classe non eredita direttamente da stanza, ma eredita da suite perché è anch'essa una suite. E' una classe concreta con un massimo di 4 posti disponibili al suo interno. Eredita lo sconto dalla Suite.

## 3 Descrizione dell'uso di codice polimorfo

I puntatori "polimorfi" alla classe base Stanza sono presenti in 2 parti:

- 1) Nel vector di vector.
- 2) Come campo info di un nodo della lista delle prenotazioni.

E' logico quindi pensare che venga utilizzato del codice polimorfo nei metodi delle classi. Sicuramente, esso è presente durante la "costruzioni" delle varie tipologie concrete delle stanze.

Nei metodi della classe Contenitore, viene passato sempre un puntatore alla stanza, trattasi di un "puntatore" polimorfo che eventualmente può accedere ai vari campi della classe.

Anche la GUI utilizza del codice poliformo. Ad esempio, nella classe Struttura quando viene selezionata una stanza, per ognuna di esse, viene scritto il numero di posti resi da quest'ultima disponibili, chiamando la funzione virtual PostiDisponibili() const.

Infine, per quanto riguarda la tabella contenente le prenotazioni, si può parlare di codice polimorfo. Vengono, infatti, richiamate le funzioni virtuali "Nome()" (che ritorna una QString con il nome associato alla stanza) e CostoPrenotazione(int,int) (che calcola il prezzo complessivo da pagare per la prenotazione della singola stanza). Quest'ultimo calcola il prezzo tramite una formula:  $C \cdot \text{days} + (\text{tassa} \cdot \text{days}) \cdot \text{pers}$ , dove

C = è un prezzo fisso che varia a seconda della stanza considerata.

days = sono la durata in giorni della prenotazione.

tassa = è un dato comune a tutte le stanze. Si tratta, infatti, della tassa di soggiorno da pagare per ogni giorno trascorso nell'hotel.

pers = è il numero di persone per il quale è stata fatta la prenotazione.

Nella classe Suite e Royal Suite è presente anche lo sconto se i giorni di prenotazione della stanza in questione sono almeno sei.

Esso sarà, dunque, diverso per ogni classe nella gerarchia e verrà chiamato in maniera polimorfa.

## 4 Manuale per la Gui

Di seguito verranno fornite delle indicazioni utili per l'utilizzo corretto del programma.

Inizialmente è importante dire che non è possibile aprire la stessa finestra due volte e, se la pagina principale (ovvero quella che corrisponde alla classe MainWindow e che si apre all'inizio) verrà chiusa, insieme a lei, verranno chiuse tutte le altre finestre aperte.

Dalla pagina iniziale si può accedere a due pagine cliccando le immagini, una sotto la scritta "**Struttura**" e l'altra sotto la scritta "**Prenotazioni**".

Accedendo alla pagina "Struttura", nel primo riquadro a sinistra si può scegliere una tipologia di stanza. Scelta quest'ultima, appariranno alcune informazioni sul tipo scelto: i posti disponibili e la collocazione di esse all'interno dell'hotel.

In questa pagina si può, inoltre, rendere inagibile una certa stanza, selezionandone il piano e il numero. Una volta resa inagibile non sarà più possibile prenotare quella stanza finché non la si renderà di nuovo agibile.

Nel caso eventuale in cui si voglia rendere inagibile una stanza che è già nella lista delle prenotazioni, questa operazione non sarà resa disponibile, mostrando un avviso sull'impossibilità dell'operazione.

Cliccando invece, l'altra immagine, si aprirà una finestra all'interno della quale si dovranno inserire i dati per effettuare la prenotazione. Nel caso in cui la prenotazione non sia andata a buon fine è disponibile un pulsante "verifica disponibilità" tramite il quale si può effettivamente vedere se la stanza in questione sia disponibile o meno in quei determinati giorni. Inoltre, questo pulsante, è stato aggiunto in modo tale che si riesca a vedere se una determinata stanza sia prenotabile senza dover per forza prenotarla.

Cliccando, infine, l'ultimo pulsante "Visualizza Prenotazioni" si può accedere alla QTableWidget che raffigura tutte le prenotazioni effettuate fino a quel momento.

In questa pagina sono presenti 4 pulsanti:

- 1) **Ordina:** la lista in sé è ordinata in modo tale da mostrare in testa l'ultima prenotazione effettuata. Cliccando questo pulsante le si può ordinare per data di inizio. Originariamente questo pulsante era stato pensato come una QComboBox che rendeva disponibile vari tipi di ordinamento (per Tipologia, Piano, Numero, ecc...).
- 2) **Aggiorna:** questo pulsante ha una duplice funzionalità. La prima, nel caso in cui si effettuino delle prenotazioni con la lista di esse già aperte, con il pulsante aggiorna vengono inserite all'interno della lista secondo l'ordine prestabilito. La seconda è che, in caso la fine di una delle prenotazioni superi la data corrente, quest'ultima viene automaticamente eliminata dalla lista.
- 3) **Rimuovi:** per rimuovere effettivamente una prenotazione, prima di cliccare il pulsante, è necessario selezionare la riga della lista da rimuovere. Si possono selezionare anche più righe nel caso di una molteplice rimozione dalla lista.
- 4) **Rimuovi Tutto:** rimuove tutte le prenotazioni all'interno della lista.

Ogni qualvolta si chiuderà una pagina inerente alle prenotazioni tramite l'apposito bottone "x", queste verranno salvate nel file "outputPrenotazioni.txt" presente nella directory corrente del progetto.