

Relazione del progetto di  
Programmazione e Modellazione a Oggetti

Smacchia Diego  
Matricola 294374

## 1. Specifica del progetto

Il progetto consiste nella simulazione della gestione di un albergo, composto da tre diverse tipologie di stanze:

- le camere standard;
- le camere superior;
- le suite.

Ogni stanza differisce dalle altre per il prezzo per notte e per i servizi disponibili per i clienti (un cliente sistemato in una suite avrà più servizi disponibili, ma la suite è più costosa delle altre stanze).

I clienti che frequentano l'albergo sono di due tipi: normali e viziati: i clienti normali hanno ogni giorno la possibilità del 40% di voler usufruire di un servizio mentre i clienti viziati hanno una possibilità del 100%, e un cliente ogni 5 è viziato.

L'albergo è composto da 18 stanze (10 stanze normali, 5 superior e 3 suite), e i clienti che vogliono prendere una camera sono 22, i cui nomi vengono presi in input da un file di testo.

Ogni cliente dispone di un budget determinato casualmente, dai 350 ai 1500 euro, e vuole alloggiare in una stanza per un numero di giorni che va da 1 a 7.

All'arrivo ogni cliente contatta il gestore dell'albergo per provare a prenotare una stanza, e se ce n'è almeno una ancora libera (e il suo budget lo consente) riesce ad ottenere un posto nell'albergo.

Ogni cliente vuole alloggiare nella migliore tipologia di stanza che può permettersi, ma in caso di mancanza di posto si accontenterà anche delle altre.

Durante la permanenza nell'albergo ogni cliente si rilassa ogni giorno, e può richiedere un servizio dall'albergo e pagarlo.

I clienti che non sono riusciti a prendere una stanza continueranno a provare fino a quando non riescono a fare la loro vacanza, e il programma termina con la chiusura dell'albergo quando non c'è rimasto più nessun cliente da gestire.

## 2. Studio del problema

Uno dei primi problemi da affrontare è la scelta tra la creazione di un programma che esegue da solo o che necessita di iterazione con l'utente, poi bisogna stabilire i compiti delle singole parti del programma e in che modo andranno ad interagire tra di loro.

Ho scelto di scrivere un programma che esegue da solo, in quanto è una simulazione, in modo che l'utente possa limitarsi a leggere i dati stampati a video (clienti che fanno il check-in, il check-out o che richiedono un servizio) e verificare il corretto funzionamento del programma.

Per quanto riguarda i compiti assegnati alle singole componenti:

- Gli oggetti di tipo camera potranno essere occupati e liberati, e permetteranno di accedere al prezzo e alla lista di servizi;

- I clienti occuperanno le camere, e permetteranno di accedere ai loro dati (nome, budget, giorni di vacanza, ...);

- Il gestore si occuperà dei check-in, check-out e di assegnare le stanze ai clienti.

Per quanto riguarda la comunicazione tra le parti del programma, i clienti non andranno mai ad interagire con l'albergo, può farlo solo il gestore dell'albergo, ma ogni cliente potrà interagire con la stanza che gli è stata assegnata per richiedere un servizio.

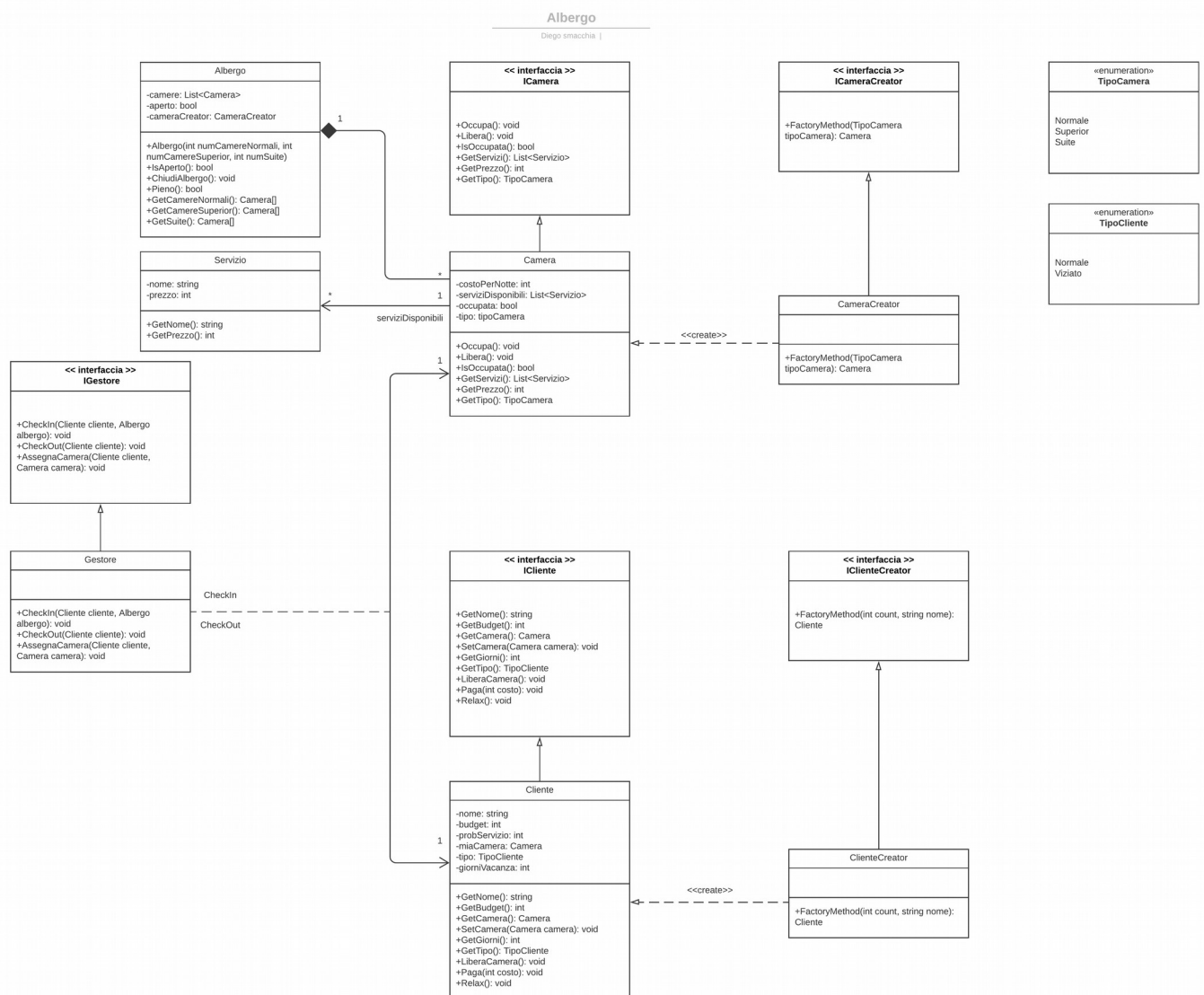
### 3. Scelte architetturali

Per implementare le diverse tipologie di camere e di clienti non ho realizzato delle classi derivate, ma ho creato due tipi di enumerazione: TipoCamera e TipoCliente, ho deciso di procedere in questo modo in quanto sia le diverse camere che i diversi clienti differiscono tra loro solo per un parametro o due, quindi creare più classi derivate non mi sembrava la scelta migliore.

Per la creazione delle camere e dei clienti ho deciso di utilizzare il pattern Factory Method, in modo da lasciare alle classi CameraCreator e ClienteCreator il compito di creare nuovi oggetti facendo tutte le valutazioni del caso (ad esempio quando si va a creare un cliente, il metodo FactoryMethod tiene conto di un intero che gli viene fornito in input per stabilire se si tratta di un cliente normale o viziato).

La classe Gestore implementa l'interfaccia IGestore, che contiene solo le firme dei metodi implementati nella classe.

Lo schema delle classi UML è il seguente:



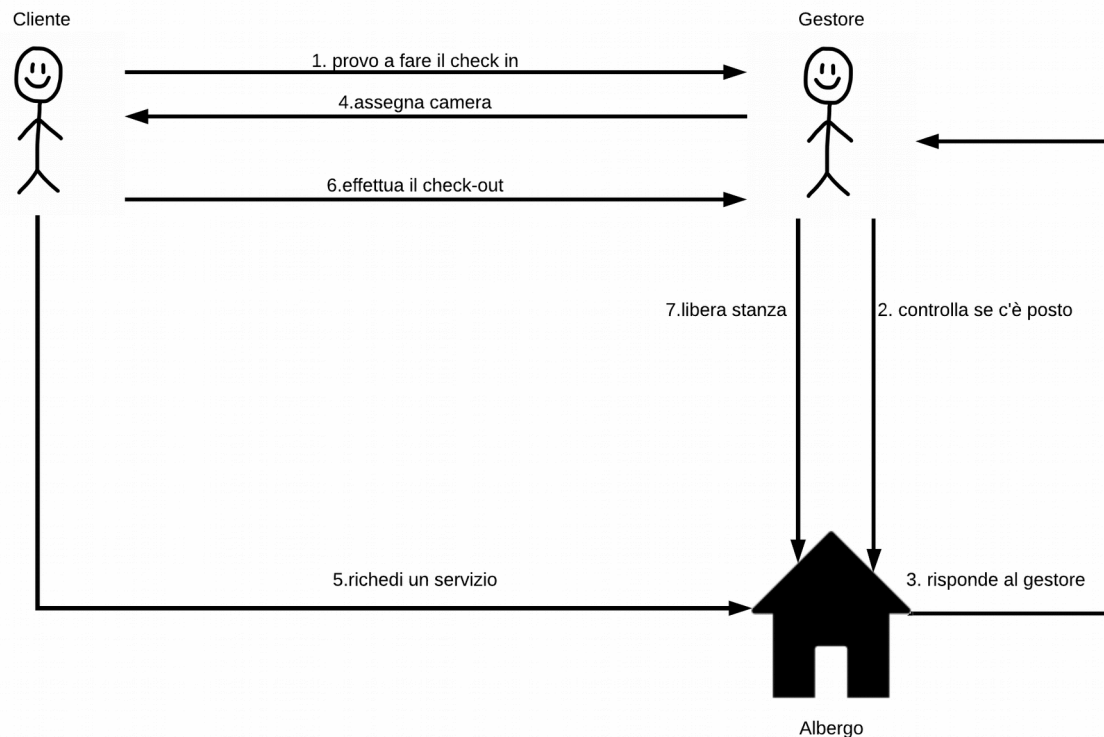
(nel caso in cui il diagramma non fosse leggibile lo includerò in formato pdf nella consegna)

#### 4. Documentazione sull'utilizzo

Il programma esegue senza bisogno di parametri in ingresso o passi particolari da seguire, l'unica cosa di cui necessita è un file di testo chiamato "nomi.txt", nel quale in ogni riga viene scritto il nome di un cliente.

## 5. Use case

Lo use case del programma che ho realizzato è il seguente:



Essendo una simulazione la cui esecuzione è automatica, c'era solo un diagramma use case da disegnare, poiché le azioni che vengono effettuate sono sempre le stesse, infatti per ogni cliente si svolgono queste azioni:

- 1) Il cliente tenta il check-in (ripetendo questo passaggio fino a quando non riesce a effettuarlo);
- 2) Il gestore fa tutti i controlli necessari (se c'è posto e se il cliente può permettersi la stanza);
- 3) L'albergo risponde con il suo stato attuale (pieno o libero);
- 4) Il gestore assegna una stanza al cliente;
- 5) Il cliente può richiedere un servizio ogni giorno;
- 6) Il cliente effettua il check-out;
- 7) Viene liberata la stanza occupata dal cliente.