



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

Corso di Laurea in Informatica Applicata

Programmazione e modellazione ad oggetti – *Progetto sessione straordinaria 2021/22*

Sviluppatore:

Laura Franchi
Matricola: 293476

INDICE

| | |
|----------------------------------------------------|----------|
| PARTE PRIMA – Specifica del Progetto | pagina 4 |
| PARTE SECONDA – Studio del Problema | pagina 5 |
| PARTE TERZA – Scelte architettureali | pagina 6 |
| PARTE QUARTA – Documentazione sull'utilizzo | pagina 8 |
| PARTE QUINTA – Use cases | pagina 9 |

1. Specifica del progetto

Il progetto consiste nella simulazione della gestione di un Centro Estetico.

Il centro è composto di 15 postazioni: 7 svolgono servizi per capelli, 5 per il viso e 3 per le unghie.

Ogni postazione ha un prezzo diverso, che è comprensivo del primo servizio base.

Ciascun tipo di postazione contiene inoltre una lista di servizi disponibili.

I clienti avranno un nome, preso da un file di testo, un budget e un numero casuale di servizi che vogliono effettuare. Sarà inoltre scelto casualmente il tipo di cliente (capelli, viso, unghie).

Il gestore si occupa di gestire i clienti: assegnare una postazione, farli pagare, liberare le postazioni.

Una volta occupata una postazione, i clienti che lo desiderano potranno chiedere altri servizi oppure pagare e uscire per lasciare spazio a quelli in coda.

Se il cliente richiede un servizio che non può permettersi verrà fatto uscire.

I clienti in coda continueranno a fare richiesta di entrare finché non verranno gestiti. Il programma termina quando la lista di clienti termina.

2. Studio del problema

Ho deciso di sviluppare il programma come una simulazione, di conseguenza non interagisce con l'utente e si limita a stampare a video i risultati e l'andamento del programma, in modo da poterne verificare il corretto funzionamento. Vengono quindi semplicemente stampati a video messaggi che notificano che un cliente ha preso una postazione, che effettua un servizio o che esce dal Centro Estetico, e l'apertura e chiusura del centro stesso.

Per quanto riguarda la lista dei clienti, ho deciso di leggere i nomi da un file di testo utilizzando la classe `StreamReader`, perché mi sembrava la soluzione più semplice, essendo una simulazione.

Per quanto riguarda i vari compiti da assegnare alle componenti del programma, gli oggetti di tipo `Postazione` possono essere occupati e liberati e permettono di restituire il loro prezzo e la lista dei servizi;

i `Clienti` possono occupare e liberare le postazioni e permettono di accedere ai loro dati (budget, nome, servizi desiderati ecc);

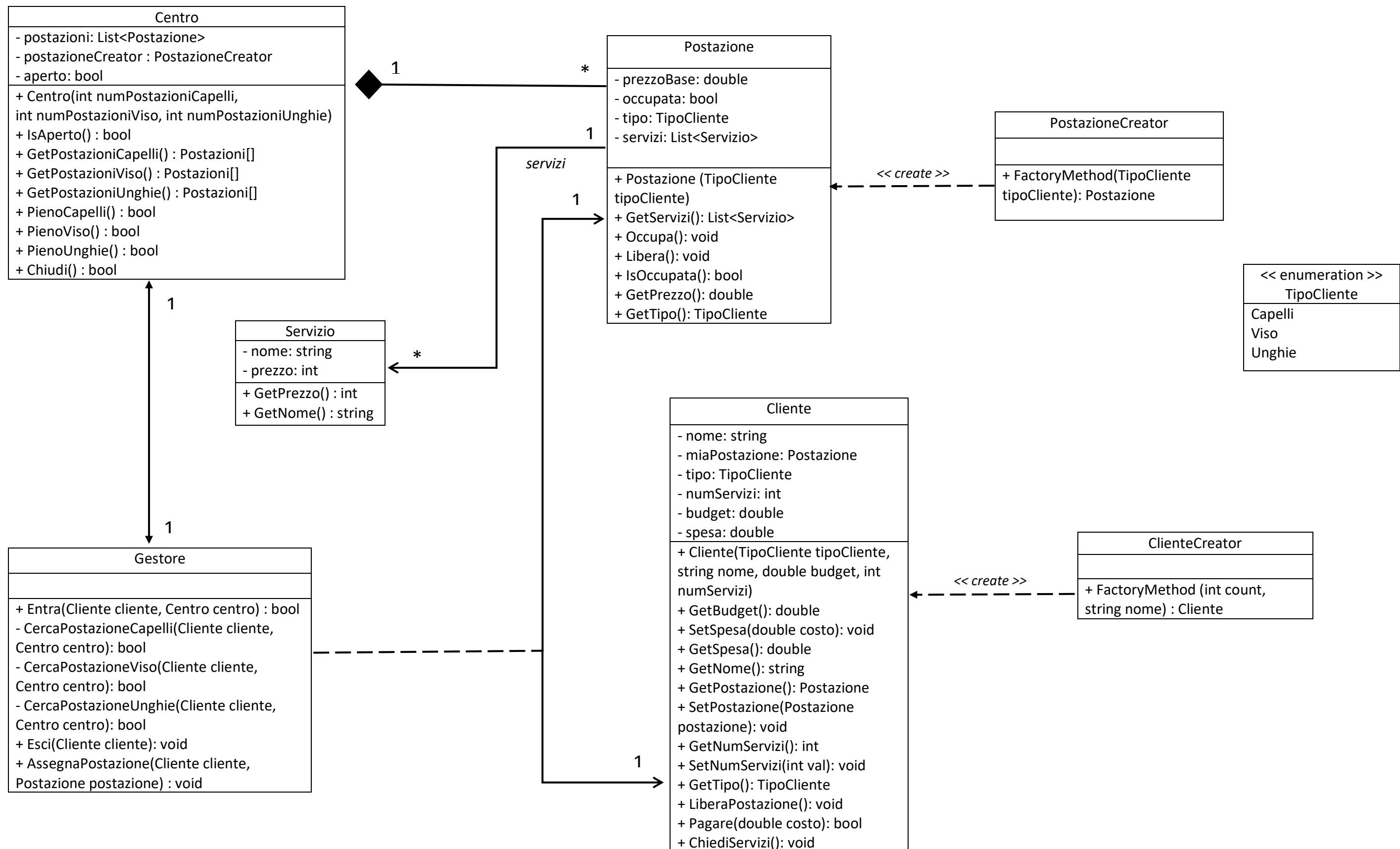
il gestore si occupa di far entrare e uscire i clienti dal centro, farli pagare e di assegnargli una postazione.

Solo il gestore può interagire con il Centro e ogni cliente può interagire con la postazione che occupa.

3. Scelte architetturali

Per implementare i tre tipi di clienti ho usato un tipo di enumerazione “TipoCliente”. Per la creazione dei clienti e delle postazioni mi sono servita del pattern creazionale FactoryMethod, questo pattern definisce un'interfaccia per la creazione di un oggetto, ma lascia che le sottoclassi decidano quale classe istanziare. In questo modo, le classi PostazioneCreator e ClienteCreator hanno il compito di creare i nuovi oggetti, facendo però alcune valutazioni. In particolare, la classe ClienteCreator utilizza dei numeri random per decidere di che tipo sarà il cliente, e per inizializzare il budget e il numero di servizi che essi vogliono effettuare.

Lo schema delle classi UML è nella pagina seguente:

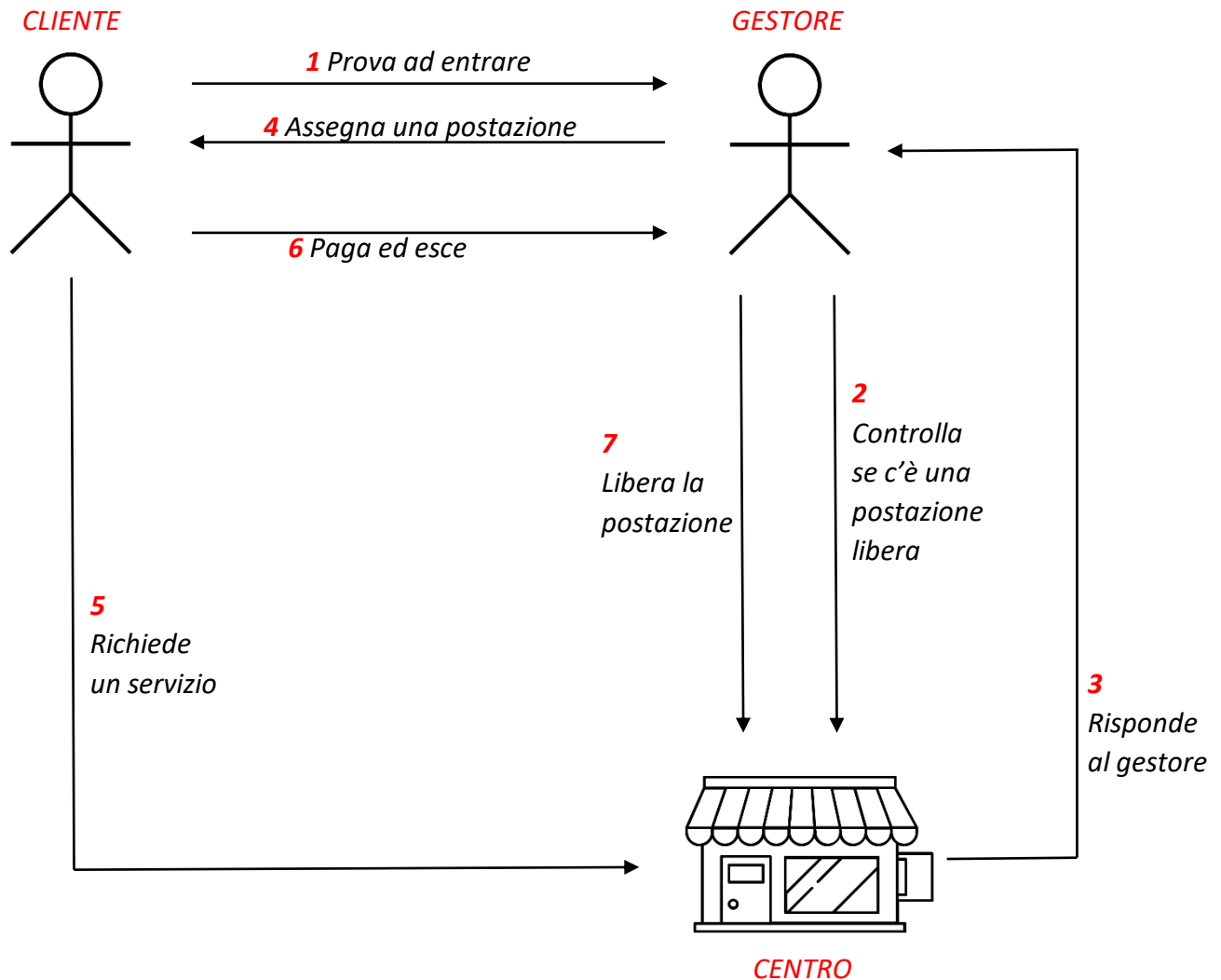


4. Documentazione sull'utilizzo

Il programma esegue da solo senza che l'utente interagisca in nessun modo, è una simulazione.

Perché il programma funzioni è necessario un file di testo denominato "nomi.txt" in cui su ogni riga è scritto il nome di un cliente.

5. Use case



Le azioni che il programma svolge sono sempre le stesse, essendo una simulazione automatica, e sono:

- 1) Il cliente prova ad entrare nel centro e prendere una postazione (riprova finché non ci riesce);
- 2) Il gestore controlla se c'è una postazione del tipo desiderato libera e se il cliente può effettivamente permettersela
- 3) Il centro risponde al gestore
- 4) Il gestore assegna una postazione al cliente
- 5) Il cliente effettua i servizi desiderati
- 6) Il cliente paga i servizi ricevuti ed esce dal centro estetico
- 7) Il gestore libera la postazione occupata dal cliente appena uscito