

Testo Esercizio

Si consideri un'agenzia che opera come biglietteria ferroviaria, aerea e navale, accettando diversi modi di pagamento. Si identifichino le principali entità coinvolte illustrando le gerarchie IS-A. Si descriva la procedura di prenotazione di un biglietto aereo, subordinata alla verifica della disponibilità di un posto, e la procedura di pagamento con carta di credito o bancomat.

Sommario

- Parte 1 (vista logica, aspetti strutturali) – rappresentazione delle gerarchie e delle classi principali del sistema in UML;
- Parte 2 (vista logica, aspetti dinamici) – rappresentazione di uno scenario d'utilizzo del sistema mediante diagrammi di sequenza.

Note.

La soluzione proposta è solo una possibile soluzione. In generale, non esiste la “soluzione corretta” o il “modello migliore”. I modelli si costruiscono esattamente come avviene per qualsiasi artefatto umano. Molti dettagli possono essere soggettivi e dipendono fondamentalmente dalla capacità di astrazione, dallo spirito di osservazione e dalla creatività dell'autore. Questa proposta di soluzione rappresenta solo una linea guida: essa costituisce un invito a sperimentare, ad essere creativi, ma allo stesso tempo ad essere coerenti. Gli aspetti fondamentali di un modello UML sono i seguenti:

- Individuare i **concetti essenziali** del sistema e rappresentarli opportunamente;
- Caratterizzare ciascun concetto con il **livello di dettaglio** utile per risolvere il problema;
- Mantenere la **coerenza** tra le diverse viste (o prospettive) del modello. Se mostriamo un'interazione tra due oggetti in un diagramma di sequenza, per esempio, il metodo descritto nel diagramma deve essere stato prima definito nella classe, con gli opportuni argomenti (se ce ne sono).

Un modello è ragionevole quando contiene queste tre caratteristiche.

Svolgimento

Parte 1

Come negli esercizi precedenti, identifichiamo le principali classi che costituiscono il sistema. Dall'analisi linguistica del testo identifichiamo subito il concetto generale di **biglietto**. La biglietteria deve vendere i seguenti tipi di biglietti:

- Biglietti aerei;
- Biglietti navali;
- Biglietti ferroviari.

Caratterizziamo quindi la classe base e, successivamente, le classi derivate. Supponiamo che per tutti i tipi di biglietti sia possibile specificare la "classe" nella quale il viaggiatore vuole viaggiare. In prima classe, ad esempio, gli alloggiamenti (i posti a sedere nel caso di treni e aerei) risultano più curati e confortevoli. Ogni biglietto, inoltre, si riferisce ad un particolare **viaggio** che comprende:

- Luogo di partenza;
- Luogo di destinazione;

Considerando le caratteristiche dei diversi tipi di viaggio, scegliamo di non rappresentare l'ora di partenza e la durata del viaggio come caratteristiche di *Biglietto*. Pensiamo, ad esempio, ai viaggi ferroviari: esistono generalmente molti treni che un passeggero può prendere nell'arco di un intervallo di tempo limitato. Specificare un'ora di partenza ed una di arrivo in questo caso sembra essere troppo restrittivo. Ha molto più senso, invece, definire un'ora di emissione del biglietto e un periodo di validità dello stesso. Entro tale periodo il viaggiatore può prendere un qualsiasi treno che parte dal luogo di partenza e arriva al luogo di destinazione (oppure ad una stazione intermedia). Caratterizziamo questi elementi come attributi della classe *BigliettoFerroviario*.

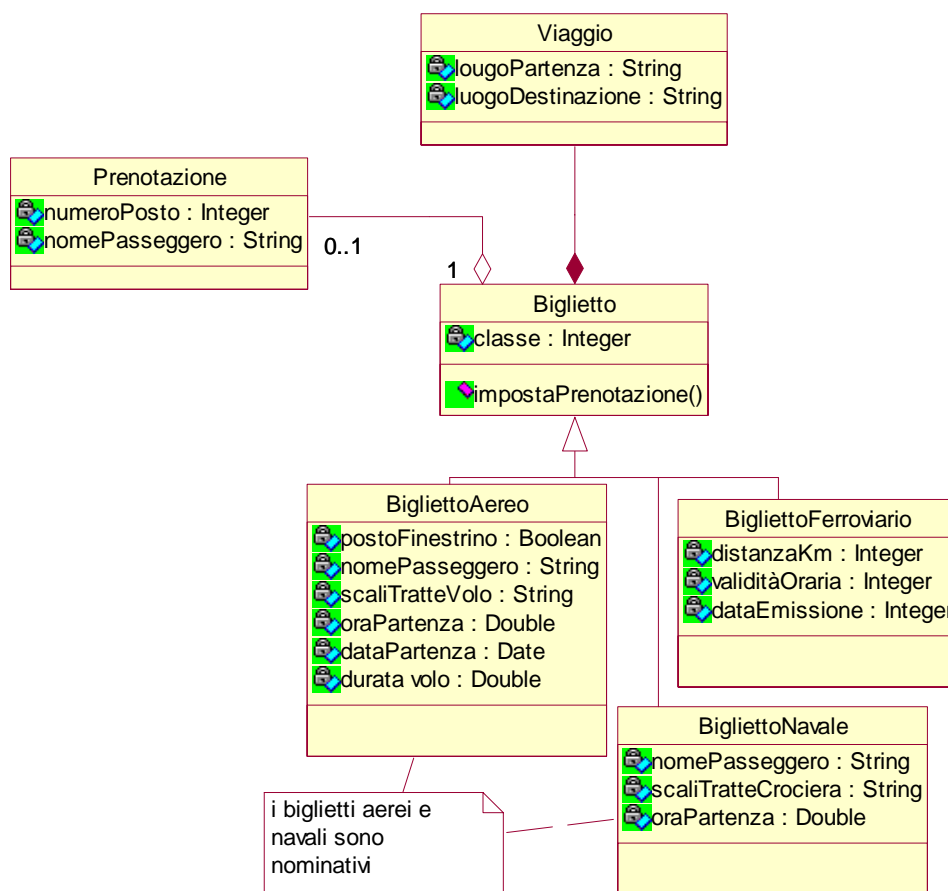


Figura 1 – Gerarchia Biglietto

Un altro attributo tipico dei biglietti ferroviari, inoltre, è la distanza chilometrica del viaggio. Consideriamo ora i biglietti aerei. Il numero di voli aerei disponibili in un prefissato intervallo di tempo è sicuramente inferiore rispetto a quello dei treni. Un biglietto aereo, infatti, viene solitamente emesso per un particolare volo che parte in un giorno e ad un'ora prefissati. Optiamo in questo caso per l'inserimento di un attributo che definisce la data, l'ora di partenza e la durata del volo come caratteristica di un biglietto aereo. Un volo, inoltre, può essere caratterizzato da opportuni scali. Inseriamo, perciò, anche l'attributo *scaliTratteVolo*. I biglietti aerei e quelli navali, inoltre, hanno la caratteristica di essere nominativi (a differenza di quelli ferroviari). Aggiungiamo un attributo *nomePasseggero* nelle classi *BigliettoAereo* e *BigliettoNavale*. Analogamente ai voli aerei, inoltre, possiamo immaginare che anche i viaggi in nave siano caratterizzati da uno o più scali. Per un viaggio in nave, infine, prevediamo solo un'ora di partenza (ma potrebbe essere forse più realistico prevedere una data di partenza ed una data di arrivo, dato che i viaggi in nave – ed in particolare le crociere – durano spesso uno o più giorni).

Un'ultima caratteristica associata ad un viaggio è la prenotazione. Immaginiamo di poter prenotare tutti i tipi di viaggi. Una prenotazione sarà caratterizzata da un numero che riserva uno specifico posto (o alloggiamento, nel caso di una nave) al passeggero. Ogni biglietto può avere al più una prenotazione. Anche la prenotazione è nominativa. La Figura 1 illustra la gerarchia *Biglietto*.

Esaminiamo adesso la **biglietteria** come entità. Quando l'utente si reca in biglietteria può compiere le seguenti operazioni:

- Chiedere informazioni al personale addetto in biglietteria relativamente ad un particolare viaggio;
- Acquistare un biglietto per un particolare viaggio;
- Effettuare una prenotazione;
- Pagare con diversi strumenti d'acquisto (bancomat, contanti, carta di credito, ecc.).

Immaginiamo che questi servizi vengano erogati direttamente dal personale addetto alla biglietteria che si avvale di appositi **terminali** video installati in biglietteria. Dovendo modellare questa situazione, possiamo caratterizzare la biglietteria come un oggetto composto dai seguenti terminali:

- **Terminale per le prenotazioni** (*TerminalePrenotazioni*);
- **Terminale elettronico per il pagamento** (*TerminaleElettronicoPagamenti* – accetta solo transazioni elettroniche di pagamento quali bancomat, carta di credito, ecc.);
- **Cassa** (*CassaPagamentiTradizionali* – per il pagamento in contanti e assegni).

Questi tre tipi di terminale non hanno un'interfaccia comune, per cui non è opportuno utilizzare l'ereditarietà per rappresentarli. Utilizzeremo invece la composizione per descrivere la relazione che esiste tra queste entità e la biglietteria. Supponiamo inoltre che il terminale elettronico pagamenti possa fungere sia da bancomat, sia da lettore carte di credito (come in effetti accade nella realtà). In alternativa, potremmo qui utilizzare convenientemente l'ereditarietà per distinguere tra i due terminali (lettore bancomat e lettore carta di credito). Questa volta, infatti, si tratta in entrambi i casi di terminali elettronici per il pagamento, caratterizzati da un'interfaccia comune (autenticazione dell'utente, verifica disponibilità credito, effettuazione transazione).

Nel nostro modello aggiungiamo le classi relative ai terminali menzionati e all'addetto alla biglietteria. Quest'ultima classe contiene i metodi per iniziare tutte le interazioni con l'utente. In particolare l'utente si rivolgerà all'addetto biglietteria per effettuare i pagamenti, richiedere una prenotazione, ritirare i biglietti. L'**addetto alla biglietteria** funge da intermediario tra l'utente e il sistema. Aggiungiamo un'ultima relazione tra la biglietteria e la classe biglietto, evidenziando il fatto che una biglietteria può essere vista anche come un "contenitore" di biglietti. Utilizziamo in questo caso l'aggregazione poiché il numero di biglietti può variare nel tempo a seconda della quantità di richieste e della disponibilità. La Figura 2 illustra le relazioni tra biglietteria, addetto alla biglietteria e biglietto.

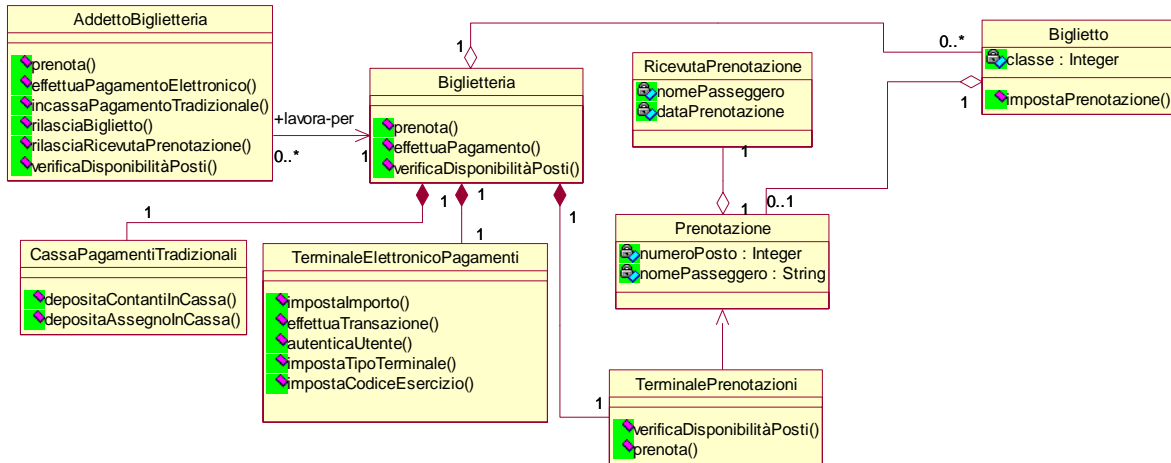


Figura 2 – Le relazioni tra biglietteria, addetto biglietteria e biglietto

Introduciamo ora il concetto di **strumento di pagamento** per descrivere la possibilità da parte del cliente della biglietteria di effettuare diversi tipi di pagamenti. Distinguiamo, in particolare, tra pagamenti elettronici (carta di credito e bancomat) e pagamenti tradizionali (contanti, assegno). Avremo una gerarchia la cui classe base più astratta è *StrumentoDiPagamento*. Essa sarà caratterizzata dall'importo da pagare. Le classi derivate saranno invece caratterizzate dagli attributi che caratterizzano ciascun particolare strumento di pagamento (il numero di conto corrente, oppure la scadenza della carta di credito, ecc.). Ciascuno strumento di pagamento, infine, "sa" come effettuare la propria procedura di pagamento (il metodo *paga()* degli strumenti di pagamento elettronici saprà come autenticare l'utente e completare la transazione con il conto corrente bancario, ad esempio). La Figura 3 illustra la gerarchia *StrumentoPagamento*.

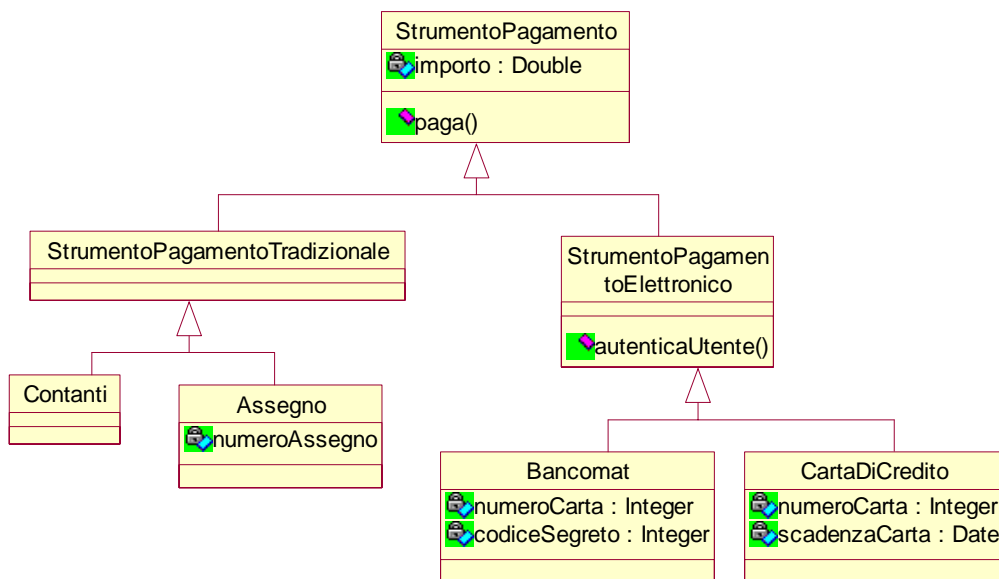


Figura 3 – Gerarchia degli strumenti di pagamento

Arrivati a questo punto dobbiamo legare il concetto di strumento di pagamento con l'utente ed un'entità che ci serve per rappresentare la banca (la transazione per il pagamento avviene tra la biglietteria e la banca dell'utente). Per semplificare l'esercizio, introduciamo la classe *ContoCorrente* che descrive il conto corrente dell'utente e immaginiamo che il terminale di pagamento interagisca direttamente con questa classe per portare a termine l'operazione di

pagamento. Utilizzando la relazione di aggregazione, possiamo dire che un utente possiede un conto corrente e possiede uno o più strumenti di pagamento. Indirettamente, tali strumenti di pagamento sono associati al conto corrente. Il diagramma di Figura 4 illustra questa situazione.

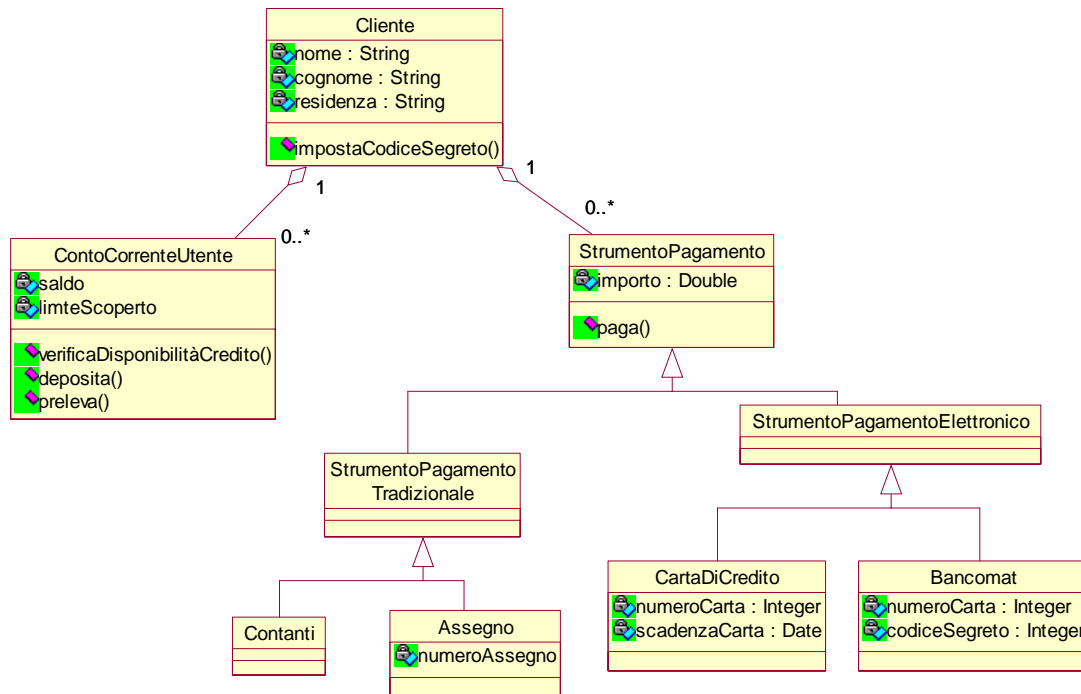


Figura 4 – Cliente, conto corrente e strumento di pagamento

Arrivati a questo punto abbiamo mettiamo tutti i concetti identificati sin qui in un unico diagramma e otteniamo la Figura 5.

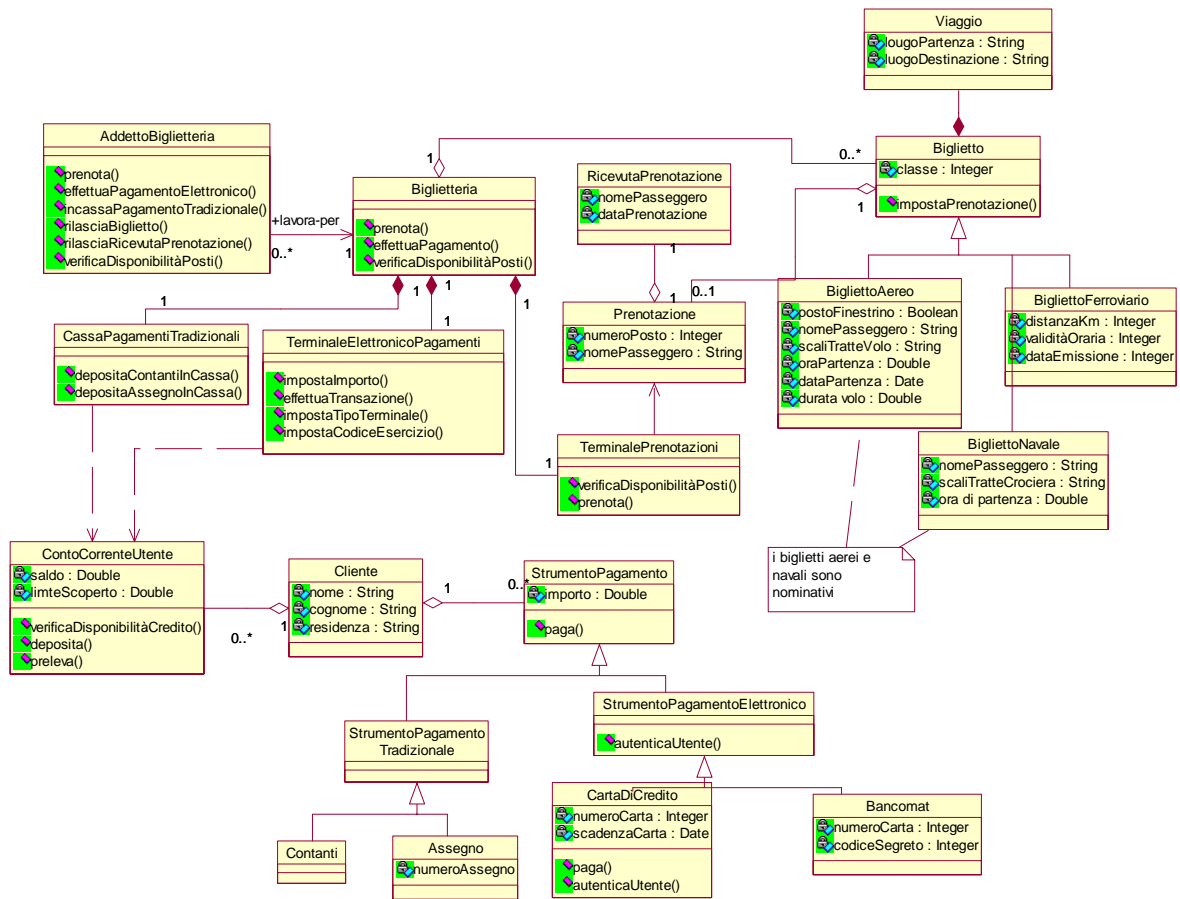


Figura 5 - La soluzione proposta completa

Parte 3

Realizziamo ora lo scenario richiesto dall'esercizio. Dobbiamo descrivere mediante un diagramma di sequenza la procedura di prenotazione di un biglietto aereo. Nello scenario illustrato dobbiamo descrivere anche la procedura di verifica della disponibilità di un biglietto, altrimenti la procedura di prenotazione non può essere completata con successo. (La verifica della disponibilità di un biglietto è comunque una funzionalità che è disponibile anche al di fuori del contesto della prenotazione: un utente potrebbe semplicemente chiedere informazioni su tale disponibilità e poi non prenotare, né comprare nulla). Gli oggetti coinvolti sono i seguenti:

- *Cliente* – è l'oggetto che "scatena" l'evento richiesta prenotazione;
- *AddettoBiglietteria* – è l'oggetto intermediario tra il cliente e la biglietteria;
- *Biglietteria* – consente di accedere a tutte le funzionalità del sistema
- *TerminalePrenotazioni* – è l'oggetto che gestisce le prenotazioni;
- *RicevutaPrenotazione* – è l'oggetto fornito all'utente dal sistema come risultato positivo di una prenotazione (poteva essere ugualmente valido far ritornare un oggetto di tipo *Prenotazione*; *RicevutaPrenotazione* descrive forse meglio la distinzione tra un'entità concettuale – la prenotazione, e un'entità fisica – la ricevuta).

Il diagramma di sequenza illustrato in Figura 6 descrive lo scenario (di successo) della verifica della disponibilità di un biglietto aereo e della conseguente prenotazione.

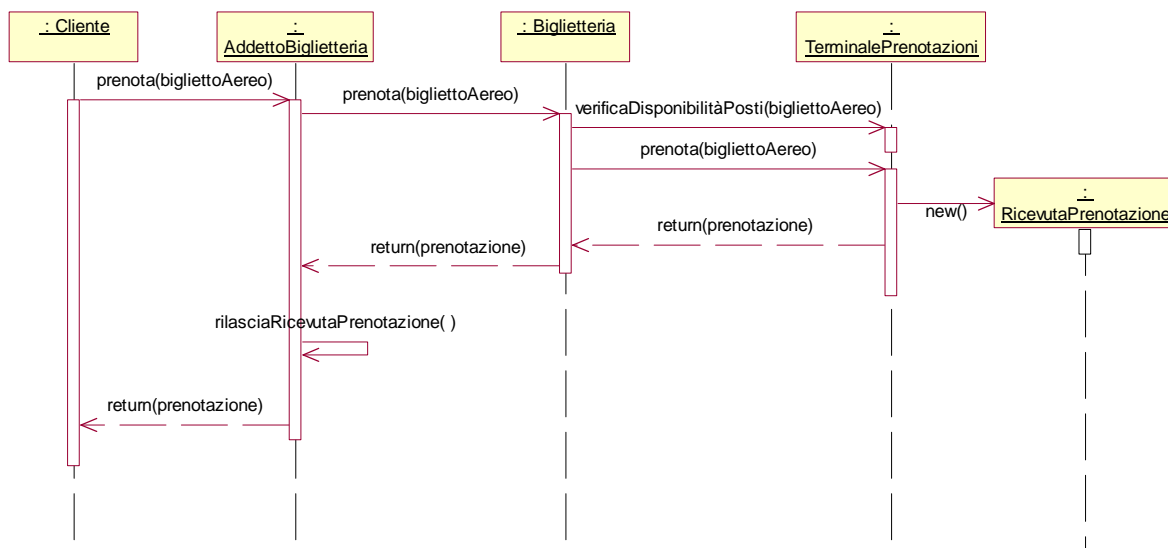


Figura 6 – Creazione di una nuova figura e sua colorazione.

Concludiamo l'esercizio con lo scenario di pagamento che vede coinvolti i seguenti oggetti:

- *Cliente*;
- *AddettoBiglietteria*;
- *TerminaleBancomat* (che funge da lettore carta di credito e che supponiamo essere una specializzazione di *TerminalElettronicoPagamenti*)¹
- *CartaDiCredito* (il particolare strumento di pagamento scelto dall'utente che "sa come effettuare una transazione");
- *ContoCorrenteUtente*.

¹ E' possibile utilizzare la classe più generica *TerminaleElettronicoPagamenti* al posto di *TerminaleBancomat*. Ho scelto quest'ultimo solo perché descrive meglio lo scenario di pagamento con un Bancomat.

Poiché nel diagramma di classe nelle figure precedenti non abbiamo sempre mostrato completamente la definizione dei tipi e dei metodi (per motivi di spazio), di seguito descriviamo in Figura 7 ciascuna classe, illustrandone l'interfaccia completa. La Figura 8 illustra invece una possibile implementazione dello scenario richiesto dall'esercizio.

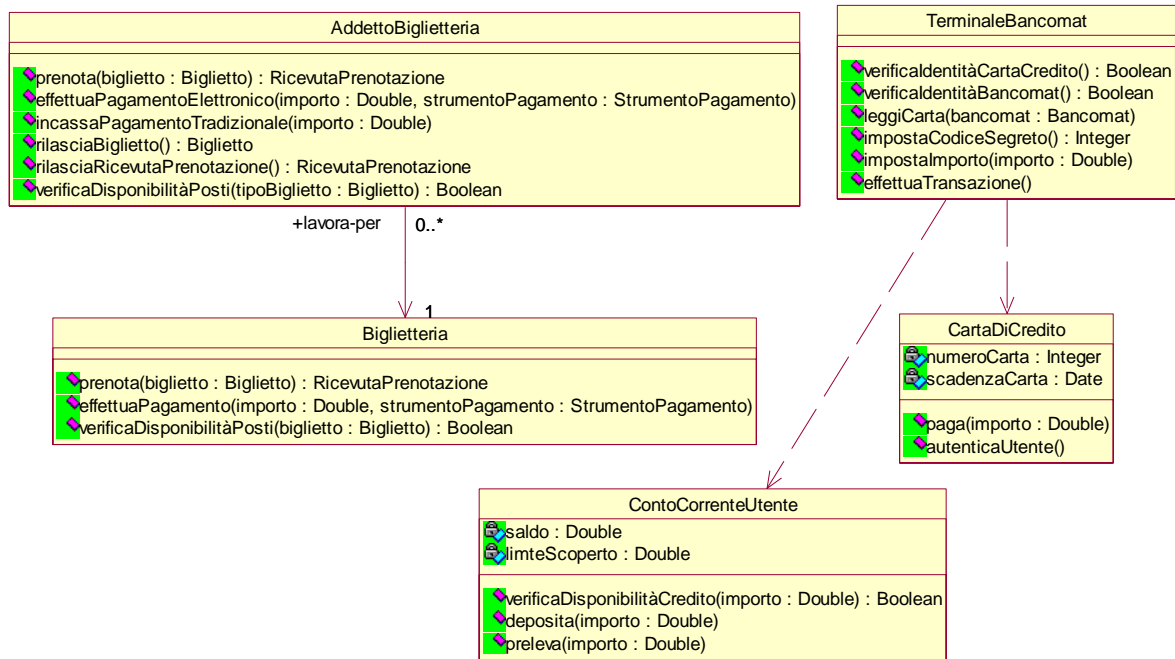


Figura 7 - Le classi coinvolte nel diagramma di sequenza richiesto con la descrizione completa dell'interfaccia

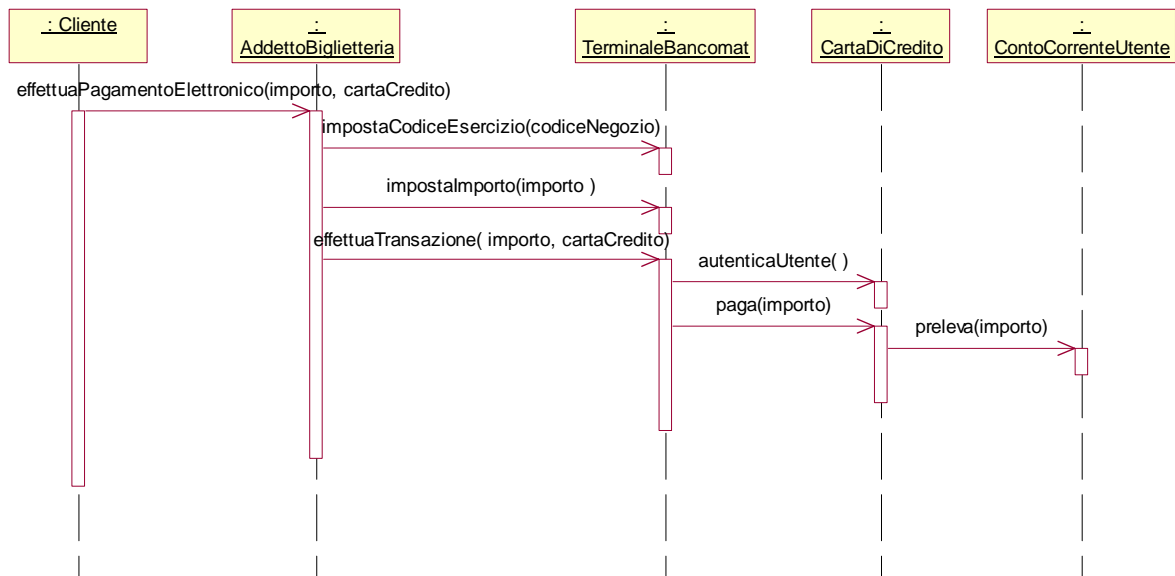


Figura 8 - Scenario di pagamento mediante carta di credito

Concludiamo con un'ultima osservazione: lo scenario descritto assume la conoscenza dell'importo del biglietto di viaggio. Come possiamo notare, infatti, esso è un parametro del metodo *effettuaPagamento()* della classe *AddettoBiglietteria*. Sapreste modificare il diagramma di sequenza

assumendo la conoscenza dello strumento di pagamento (sempre la carta di credito) e del tipo di biglietto, piuttosto che direttamente del suo importo?

In alternativa, sempre con riferimento al diagramma di Figura 8, sapreste modificare lo scenario evidenziando che il sistema fornisce come risposta (esito positivo dello scenario) un biglietto per il viaggio pagato ed, eventualmente, una ricevuta della transazione bancaria?