

Traccia di soluzione dell'esercizio del 14 Luglio 2004

1. Casi d'uso

I casi d'uso sono in Figura 1. L'attore "IlTempo" fa effettuare, a fine giornata, il controllo sulle restituzioni (la parte relativa alle interazioni con il gestore non è richiesta dall'esercizio).

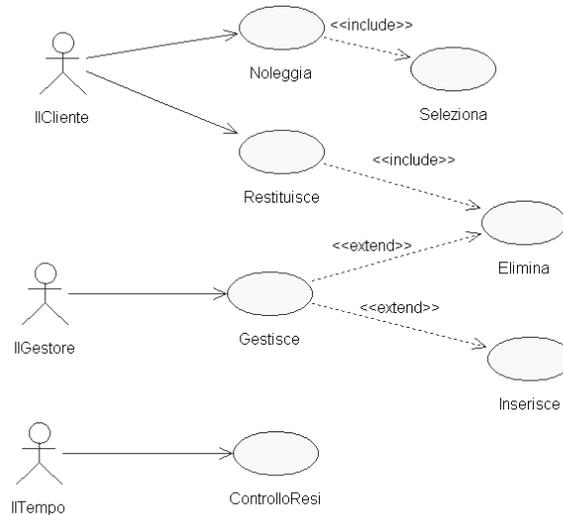


Figura 1: Diagramma dei casi d'uso.

2. Modello concettuale

Il modello concettuale (in versione iniziale) è riportato in Figura 2.

Nel modello di Figura 2 il noleggio aggrega gli articoli. Nel sistema è presente un certo numero di clienti, ognuno dei quali ha noleggiato almeno un articolo.

Si noti che Articolo è una classe astratta. L'ereditarietà consente di usare il polimorfismo per molte operazioni, ma il polimorfismo da solo non basta. Per esempio, possiamo prevedere che tutti gli oggetti abbiano il metodo `getsize()` che riporta la dimensione (lunghezza per gli sci, numero per gli scarponi), ma dobbiamo anche sapere se si tratta di un paio di sci o di un paio di scarponi. Dunque, nel modello di Figura 2 occorre prevedere che gli oggetti abbiano un tipo e un metodo (`gettype()`) che permette di sapere di che oggetto si tratta.

3. Diagramma di sequenza relativo a verifica disponibilità

Si veda il diagramma di Figura 3. Si noti che la sequenza risultante è molto prolissa (in alternativa si può prevedere un metodo che riporta direttamente l'ID se l'articolo soddisfa i requisiti).

Per quanto si riferisce al metodo `?noleggiato()`, lo schema di figura lascia immaginare che Articolo abbia una variabile che dice se l'articolo è o no noleggiato. Tale variabile può semplificare le cose, ma non è, a rigore, necessaria. Infatti, per sapere se è noleggiato un articolo può verificare se si trova in qualche relazione di noleggio (presuppone che Articolo abbia una componente –un puntatore– che lo lega a Noleggio. Tale componente deve essere inizializzata all'atto della creazione del relativo Noleggio. Vedere al punto 4).

4. Diagramma di sequenza noleggio

Si veda il diagramma di Figura 4. Il diagramma si riferisce al noleggio di un singolo articolo. Nello schema di figura si assume che il Cliente non sia già nel sistema, cioè che il cliente non stia già noleggiando qualche articolo.

Il metodo `noleggia(ID, Data)` di Negozio causa l'istanziamento del Cliente. Il cliente appena creato informa l'articolo scelto (in base a ID) che esso viene noleggiato. A sua volta l'articolo istanzia il relativo Noleggio.

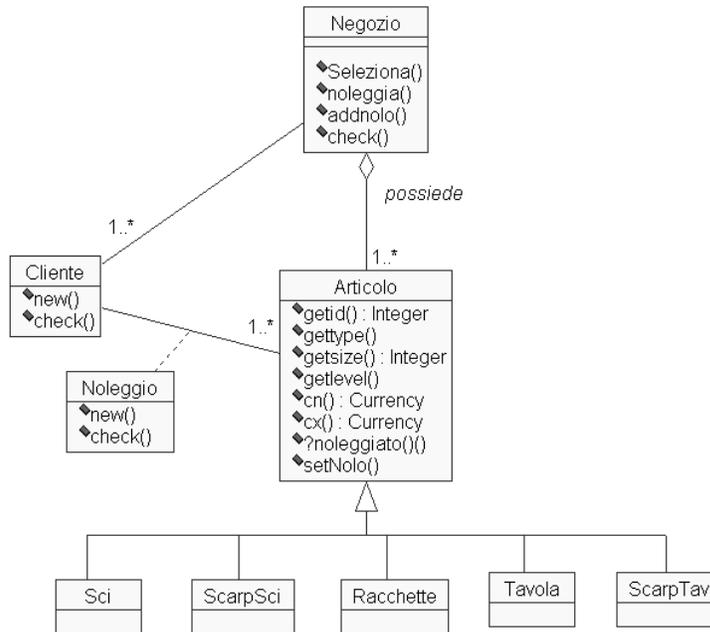


Figura 2: Modello concettuale.

Si noti che la sequenza poteva essere anche diversa; ad esempio: Negozio istanzia un Cliente, che istanzia un Noleggio. In ogni caso è necessario che gli oggetti abbiano conoscenza degli altri oggetti con cui sono messi in relazione. Per esempio, con riferimento alla Figura 4 un Articolo quando crea un Noleggio ottiene il riferimento ad esso. Per fare in modo che nel Noleggio ci sia il riferimento all'Articolo è necessario che questo riferimento sia tra i parametri del costruttore di Noleggio (questo parametro non è indicato in figura).

Con riferimento alla Figura 4, per noleggiare più di un articolo occorre prevedere il metodo `addnolo()` in Negozio. Il metodo, verificata la disponibilità dell'articolo richiesto, **non** deve portare alla creazione di un nuovo Cliente. Deve portare alla creazione di nuovo oggetto di associazione Noleggio, legandolo allo specifico cliente e all'articolo selezionato.

5. e 6.

I diagrammi relativi si ottengono in modo analogo. Alla restituzione, se l'articolo è danneggiato si usa `cx()`, in luogo di `cn()`, e si elimina l'articolo. In ogni caso si elimina il Noleggio relativo e il cliente che non ha più niente in carico (il cliente potrebbe restituire un articolo noleggiato, ma trattenerne altri; in questo caso non deve essere eliminato.).

7.

Qui l'agente che avvia l'operazione è il tempo chiamando il metodo `check()` di Negozio, che a sua volta chiama il metodo `check()` di Cliente per tutti i clienti correntemente rappresentati nel sistema. Il metodo `check()` di un cliente chiama il metodo `check()` di tutti i noleggi che il cliente ha in essere. Notare che l'ordine delle chiamate è determinato dal modello di Figura 2. Infatti Negozio è in relazione con Cliente, e cliente con Noleggio (**non** c'è una relazione diretta tra Negozio e Noleggio).

L'algoritmo è riportato al termine del paragrafo.

Nel Noleggio occorre aggiungere una componente (Inviata) che indichi se la lettera di diffida è già stata inviata¹; Inviata viene inizializzata a `false` dal costruttore e portata a `true` all'emissione delle lettera.

Si è immaginato che Printer sia un oggetto che presenta il metodo `stampa()` per la stampa della lettera. `System.date()` riporta la data corrente.

Si noti che l'algoritmo è molto rozzo: stampa una lettera per ogni articolo noleggiato. Un algoritmo più raffinato dovrebbe prevedere una sola lettera per più articoli scaduti assieme. Ciò comporterebbe: (1) lo

¹Questa componente non è strettamente necessaria (perché?), ma semplifica l'algoritmo.

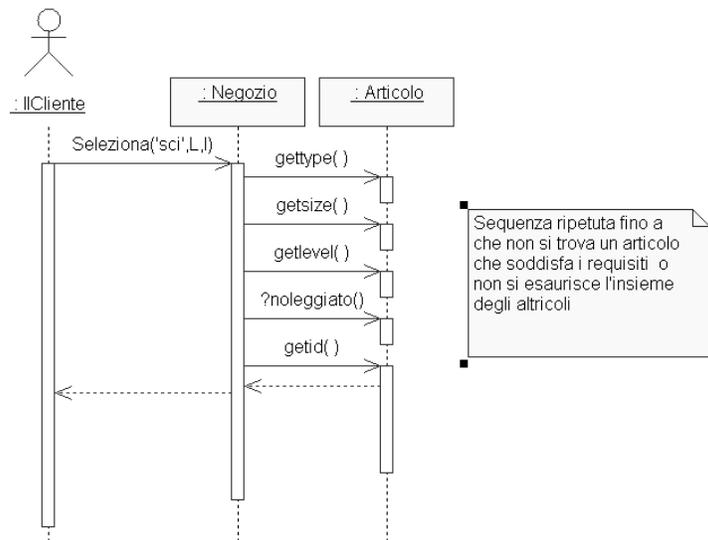


Figura 3: Diagramma di sequenza per la verifica di disponibilità. I primi tre metodi invocati servono a stabilire se il negozio possiede l'articolo di tipo, lunghezza e livello richiesti. La figura mostra solo il caso di risposte dell'articolo corrispondenti a quanto richiesto. In caso di disaccordo, si procede con il prossimo articolo. Alla fine se l'articolo esiste ed è disponibile si ottiene il suo ID (numero univoco).

spostamento della stampa in Cliente; (2) la trasformazione di `check()` di Noleggio, in modo che esso restituisca i dati di interesse al Cliente per la formazione della lettera; (3) l'eventuale aggiunta di un metodo a Noleggio per avere lo stato del noleggjo.

```

check() //di Negozio
for all cliente  Cliente.check()

check() //di Cliente
for all noleggjo  Noleggio.check()

check() //di Noleggio
{if ((Inviata==false)&(DataRestituzione+7==System.date()))
  {Printer.stampa(Cliente.getDati()) //Cliente è il cliente legato a questo noleggjo;
   Inviata= true}
}

```

8.

Le interfacce si ricavano elencando i metodi dei vari oggetti, come risultano dai diagrammi di sequenza. In Figura 5 viene data l'interfaccia di Articolo e di Negozio.

Non sono riportati i metodi relativi alla restituzione (eccetto `cn()` e `cx()`).

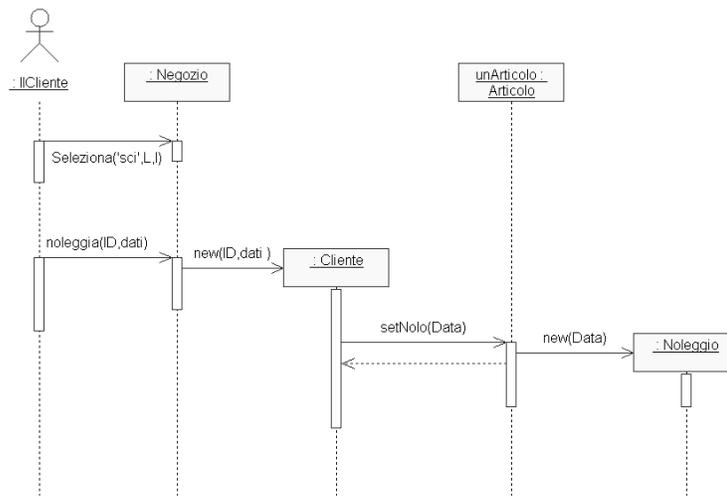


Figura 4: Diagramma di sequenza del noleggio. Si fa riferimento alla Figura 3 per quanto riguarda la disponibilità

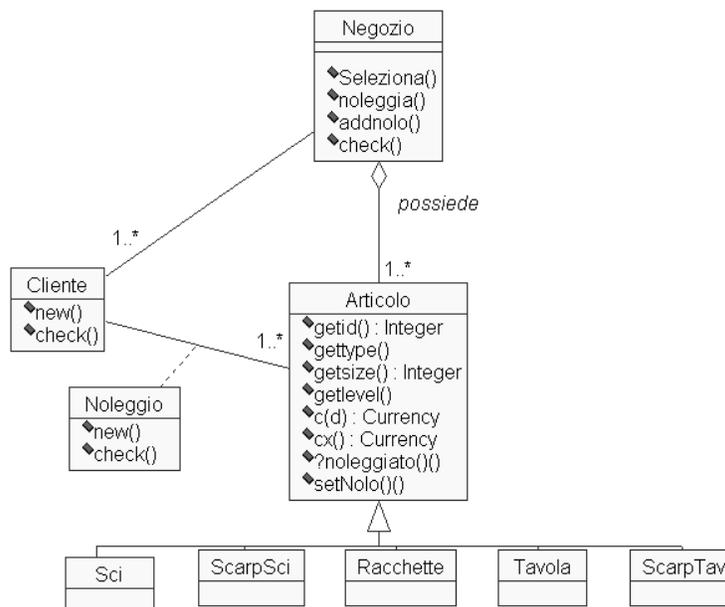


Figura 5: Interfacce di Negozio e di Articolo.