

Università di Roma “La Sapienza”

A.A. 2006-2007

Facoltà di Ingegneria

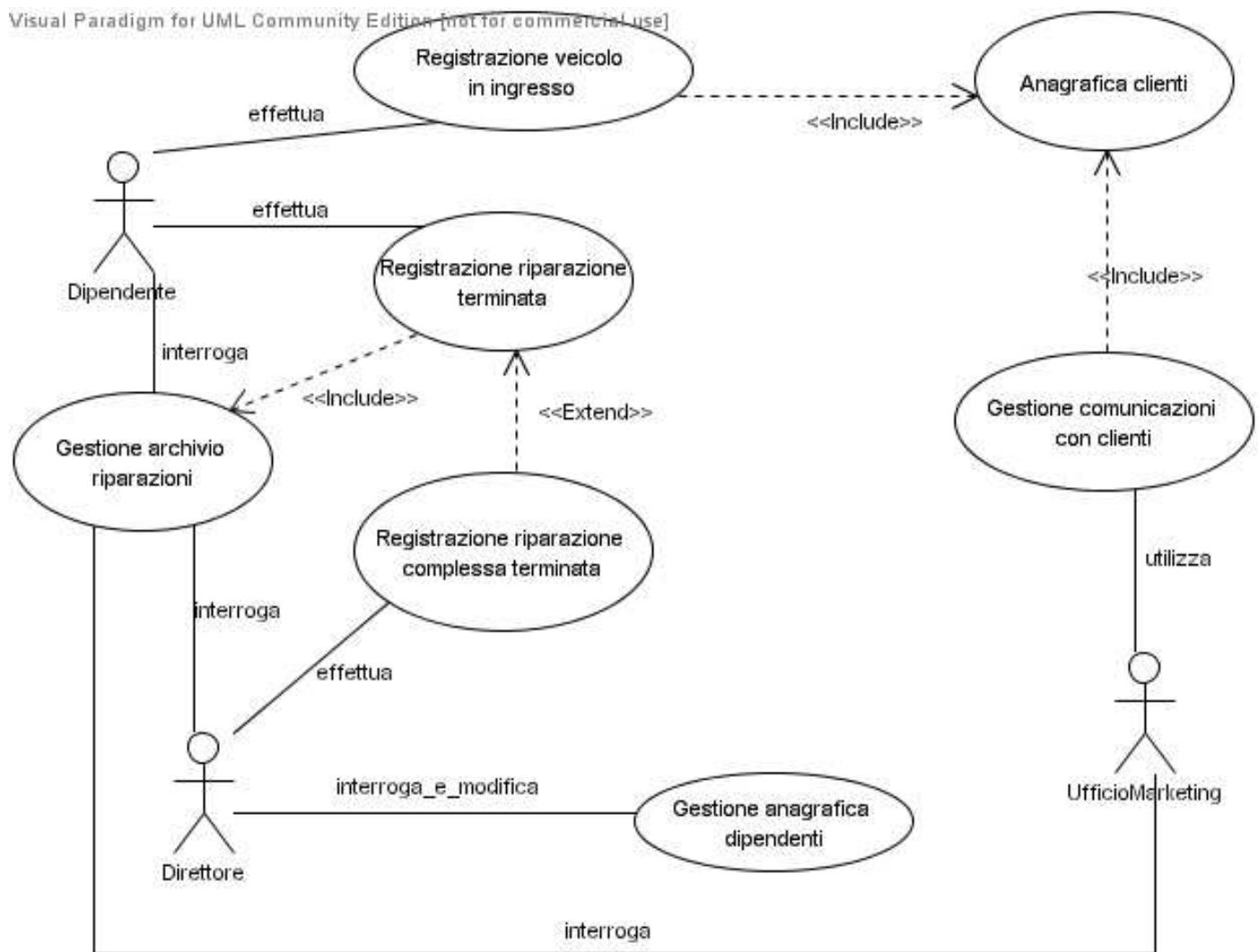
Corso di Laurea in Ingegneria Informatica

**Corso di “PROGETTAZIONE DEL SOFTWARE I”
(Canale A-L & M-Z)**

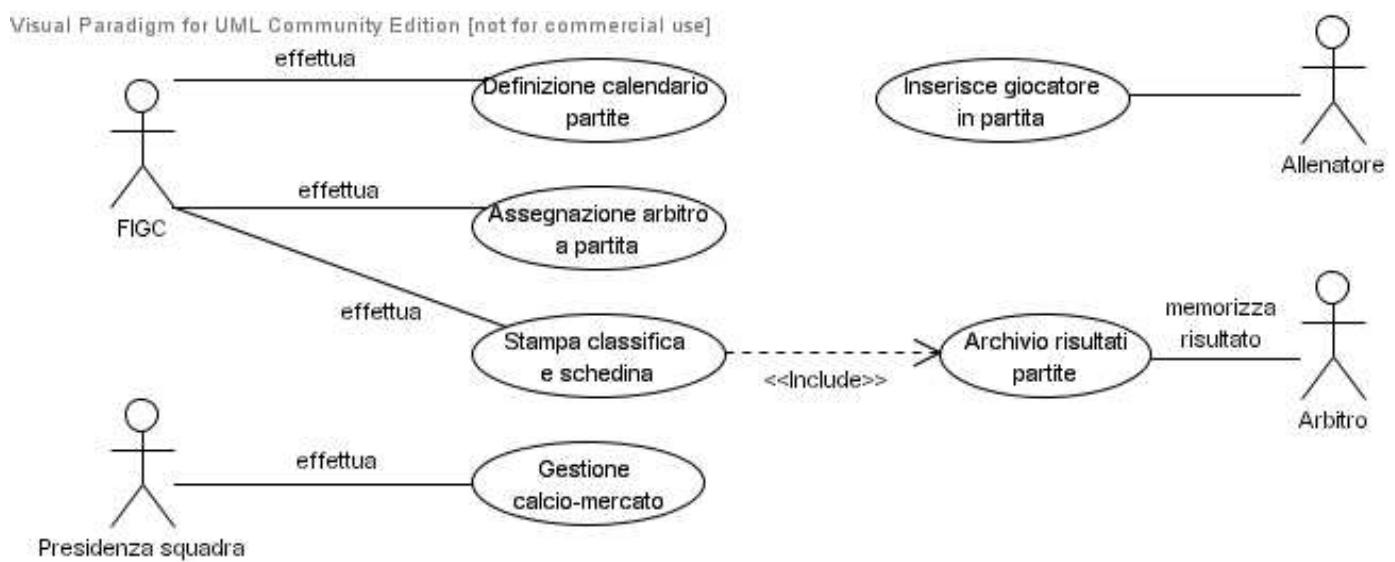
**ESERCITAZIONE 4:
LA FASE DI ANALISI**

**Sintesi del diagramma degli use case e della sua specifica a partire dai
requisiti (soluzione)**

1 Catena di officine, versione 2



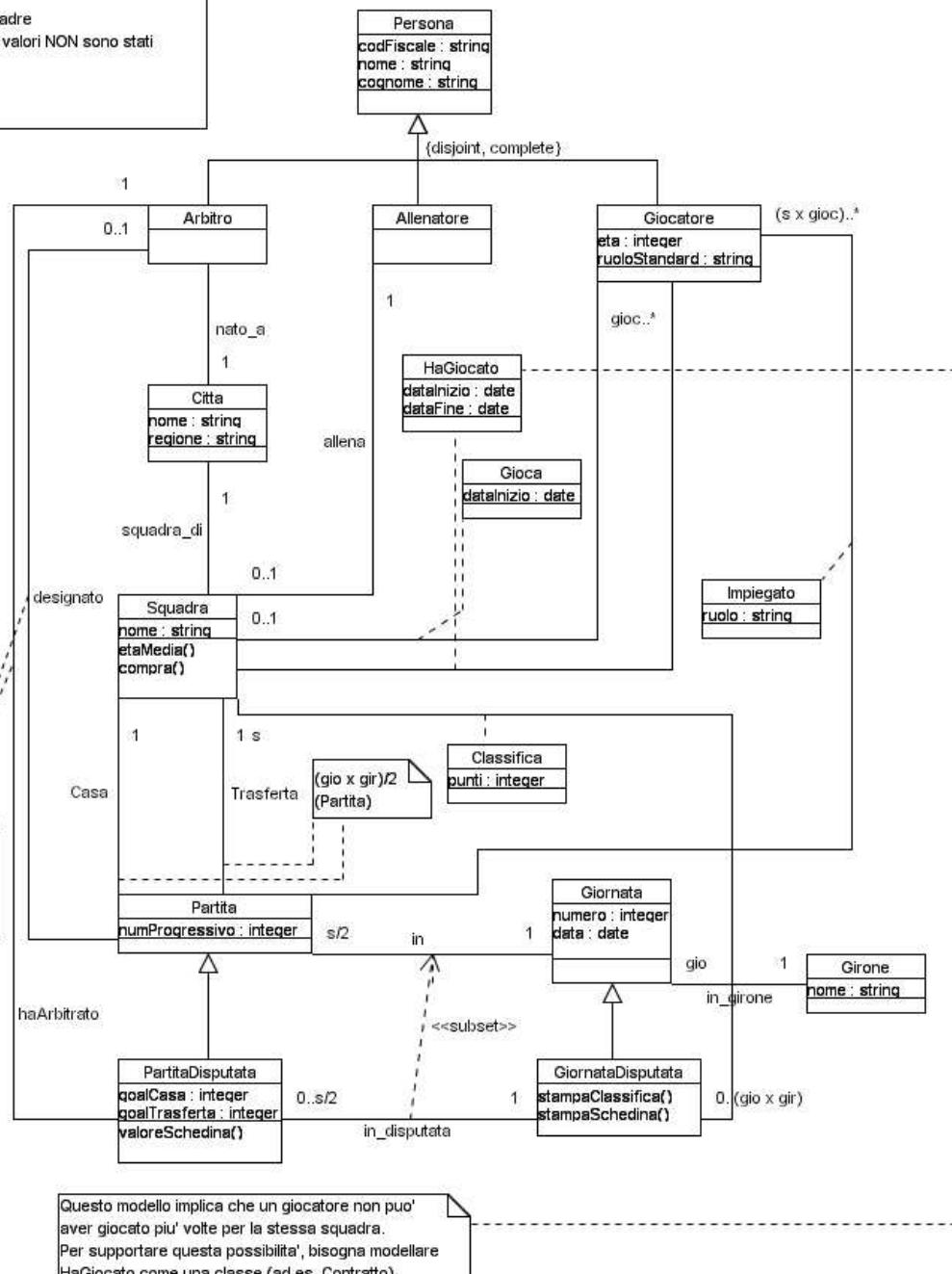
2 Campionato di calcio



Nel diagramma si fa riferimento ai seguenti valori numerici:

- s: numero di squadre (pari)
- gio: numero di giornate da disputare per girone
- gir: numero di gironi
- gioc: numero giocatori delle squadre

NB: Eventuali legami tra i suddetti valori NON sono stati considerati per semplicità.



```

InizioSpecificoClasse Squadra
compra(g:Giocatore, oggi:data)

Pre:
<g,this>  $\notin$  Gioca
// g non fa attualmente parte della squadra this.

Post:
<g,this>  $\in$  Gioca  $\wedge$  Gioca.dataInizio(<g,this>) = oggi  $\wedge$ 
// Viene creato un nuovo link Gioca tra g e this con il valore per il campo dataInizio pari a oggi e...
( $\exists$ s  $\in$  Squadra  $\wedge$  pre(<g,s>  $\in$  Gioca))  $\rightarrow$ 
// ...se esiste un link Gioca uscente da g allora
(<g,s>  $\notin$  Gioca  $\wedge$ 
// il link viene rimosso e...
<g,s>  $\in$  HaGiocato  $\wedge$ 
// viene creato un link HaGiocato tra g e s...
HaGiocato.dataInizio(<g,s>) = pre(Gioca.dataInizio(<g,s>))  $\wedge$ 
// ...con il valore per l'attributo dataInizio pari a pre(Gioca.dataInizio(<g,s>)) e...
(HaGiocato.dataFine(<g,s>) = ieri))
// ...quello per l'attributo dataFine pari alla data del giorno precedente oggi, che indichiamo con ieri.

```

etaMedia() : real

Pre:

 \exists g g \in Giocatore \wedge <g,this> \in Gioca
// Esiste almeno un link di tipo Gioca tra this e un oggetto di classe Giocatore.

Post:

 $G = \{g \in Giocatore | <g, this> \in Gioca\}$
// Detto G l'insieme dei giocatori che militano nella squadra this,
result = $(\sum_{g \in G} Giocatore.eta(g)) / |G|$
// Poniamo result pari alla media delle età dei giocatori in G.

FineSpecificoClasse

InizioSpecificoClasse PartitaDisputata

valoreSchedina(): {1,X,2}

Pre:
true // Nessuna precondizione

Post:
((goalCasa(this) > goalTrasferta(this) \rightarrow result=1) \wedge
(goalCasa(this) = goalTrasferta(this) \rightarrow result=X) \wedge
(goalCasa(this) < goalTrasferta(this) \rightarrow result=2)

FineSpecificoClasse

InizioSpecificoClasse GiornataDisputata

stampaSchedina()

Pre:

true // Nessuna precondizione

Post:

Per ogni (p, c, t) $p \in PartitaDisputata \wedge c \in Squadra \wedge t \in Squadra \wedge \langle p, this \rangle \in in_disputata \wedge \langle c, p \rangle \in Casa \wedge \langle t, p \rangle \in Trasferta$, Esegui{stampa(Squadra.nome(c), Squadra.nome(t), PartitaDisputata.valoreSchedina(p))}
// Per ogni PartitaDisputata p in relazione in_disputata con this, stampa i nomi i) della squadra che ha giocato in casa c, ii) di quella che ha giocato in trasferta t ed il risultato in schedina.

stampaClassifica()

Pre:

true // Nessuna precondizione

Post:

Per ogni (s) $s \in Squadra \wedge \langle this, s \rangle \in Classifica$, Esegui{stampa(Squadra.nome(s), Classifica.punti(<this, s>))}
// Per ogni link Classifica tra this e un oggetto di tipo Squadra, stampa il nome della squadra ed i punti ad essa relativi.

FineSpecificoClasse

La specifica degli use case è lasciata come esercizio.

3 Ristorante, versione 2



InizioSpecificoUseCase GestionePrenotazione

InserisciCliente(nome:String, tel:String) : Cliente

Pre: $\neg(\exists c \in \text{Cliente} \wedge \text{Cliente.nome}(c) = \text{name} \wedge \text{Cliente.tel}(c) = \text{tel})$

// Non esiste alcun oggetto *Cliente* con gli stessi valori per gli attributi *name* e *tel*.

Post: $(\exists c \in \text{Cliente} \wedge \text{Cliente.nome}(c) = \text{name} \wedge \text{Cliente.tel}(c) = \text{tel}) \wedge \text{result} = c$

// Viene creato un oggetto *c* di classe *Cliente* con i valori degli attributi pari ai parametri attuali. *result* è pari a *c*.

EffettuaPrenotazione(cliente:Cliente, data:date, ora:time, npersone:integer > 0, smoking:boolean)

: Assegnazione

Pre:

$T = \{t \in \text{Tavolo} | \text{Tavolo.smoking}(t) = \text{smoking} \wedge \neg(\exists a | a \in \text{Assegnazione} \wedge \langle t, a \rangle \in \text{usatoIn} \wedge \text{Assegnazione.data}(a) = \text{data} \wedge (\text{ora} - 1 \leq \text{Assegnazione.ora}(a) \leq \text{ora} + 1))\}$

// Sia *T* l'insieme dei tavoli con attributo *Tavolo.smoking* pari al parametro *smoking*, non usati in prenotazioni dello stesso giorno e in un intervallo di + o - un'ora rispetto al parametro *ora*.

$\sum_{t \in T} \text{Tavolo.numPosti}(t) \geq \text{npersone}$

// L'insieme dei tavoli *T* garantisce il posto al numero di persone relativo alla prenotazione.

Post:

$\exists a | a \in \text{Assegnazione} \wedge \text{Assegnazione.data}(a) = \text{data} \wedge \text{Assegnazione.ora}(a) = \text{ora} \wedge \text{Assegnazione.numPersone}(a) = \text{npersone} \wedge \text{Assegnazione.smoking}(a) = \text{smoking} \wedge$

// Viene creato un nuovo oggetto *a* di classe *Assegnazione*, con i valori degli attributi pari ai parametri attuali, $\langle \text{cliente}, a \rangle \in \text{Prenota} \wedge // a$ avrà un link *Prenota* con l'oggetto *cliente*. Inoltre viene creato un insieme *T'*...

$T' = \{\langle t, a \rangle | t \in \text{Tavolo} \wedge \text{Tavolo.smoking}(t) = \text{smoking} \wedge \langle t, a \rangle \in \text{usatoIn} \wedge \neg(\exists a' | a' \in \text{Assegnazione} \wedge a' \neq a \wedge \langle t, a' \rangle \in \text{usatoIn} \wedge \text{Assegnazione.data}(a') = \text{data} \wedge (\text{ora} - 1 \leq \text{Assegnazione.ora}(a') \leq \text{ora} + 1))\}$

//...di link *usatoIn* con oggetti *t* di classe *Tavolo* con attributo *Tavolo.smoking(t)* pari a *smoking*. I tavoli assegnati non sono collegati con alcun altro oggetto di classe *Assegnazione* con attributi *data* pari a *data* ed *ora* nell'intervallo $[\text{ora} - 1, \text{ora} + 1]$.

$\sum_{t \in T'} (\text{Tavolo.numPosti}(t) \geq \text{npersone}) \wedge$

// L'insieme *T'* è tale che la somma dei posti assegnati sia maggiore o uguale a *npersone*.

$\neg(\exists c | c \in \text{Cameriere} \wedge \langle c, a \rangle \in \text{serve}) \wedge \neg(\exists p | p \in \text{Portata} \wedge \langle p, a \rangle \in \text{InConto}) \wedge$

// Infine, l'oggetto *Assegnazione* creato non possiede alcun link con oggetti di classe *Cameriere*, né con oggetti di classe *Portata*

result = *a*

// *result* è pari ad *a*.

disdici(ass:Assegnazione)

Pre:

$\neg(\exists p | p \in \text{Portata} \wedge \langle \text{ass}, \text{this} \rangle \in \text{InConto})$

// Non vi sono link tra *ass* ed oggetti di classe *Portata* (di conseguenza il conto è ancora pari a 0).

Post:

$(\forall c | \text{pre}(c \in \text{Cliente} \wedge \langle c, \text{ass} \rangle \in \text{Prenota})) \rightarrow (\langle c, \text{ass} \rangle \notin \text{Prenota}) \wedge$

// L'unico (per i vincoli di molteplicità) link *Prenota* viene eliminato

$(\forall t | \text{pre}(t \in \text{Tavolo} \wedge \langle t, \text{ass} \rangle \in \text{UsatoIn})) \rightarrow (\langle t, \text{ass} \rangle \notin \text{UsatoIn}) \wedge$

// I link *UsatoIn* vengono eliminati

$(\forall c | \text{pre}(c \in \text{Cameriere} \wedge \langle c, \text{ass} \rangle \in \text{Serve})) \rightarrow (\langle c, \text{ass} \rangle \notin \text{Serve}) \wedge$

// I link *Serve* vengono eliminati

// (La precondizione garantisce che non esistono link *InConto*)

ass \notin *Assegnazione*

// L'oggetto *ass* viene eliminato.

assegnaCamerieri(ass:Assegnazione, camerieri:Cameriere[])

Pre:

true // Nessuna precondizione

Post:

$\forall c | c \in \text{Cameriere} \rightarrow \langle c, \text{ass} \rangle \in \text{Serve}$

// Per ogni oggetto *c* \in *Cameriere*, viene creato un link di tipo *serve* tra *ass* e *c*.

InizioSpecificoUseCase GestioneConto

AggiungiPortata(ass:Assegnazione, portata:Portata, n:intero > 0)

Pre: Le stesse di ass.aggiungiPortataInConto(n, portata)

Post: Le stesse di ass.aggiungiPortataInConto(n, portata)

(Viene invocata l'operazione ass.aggiungiPortataInConto(n, portata))

Conto(ass:Assegnazione) : valuta

Pre: Le stesse di ass.conto()

Post: result è pari a ass.conto()

(Viene invocata l'operazione ass.conto())

FineSpecificoUseCase

InizioSpecificoUseCase GestioneRisorse

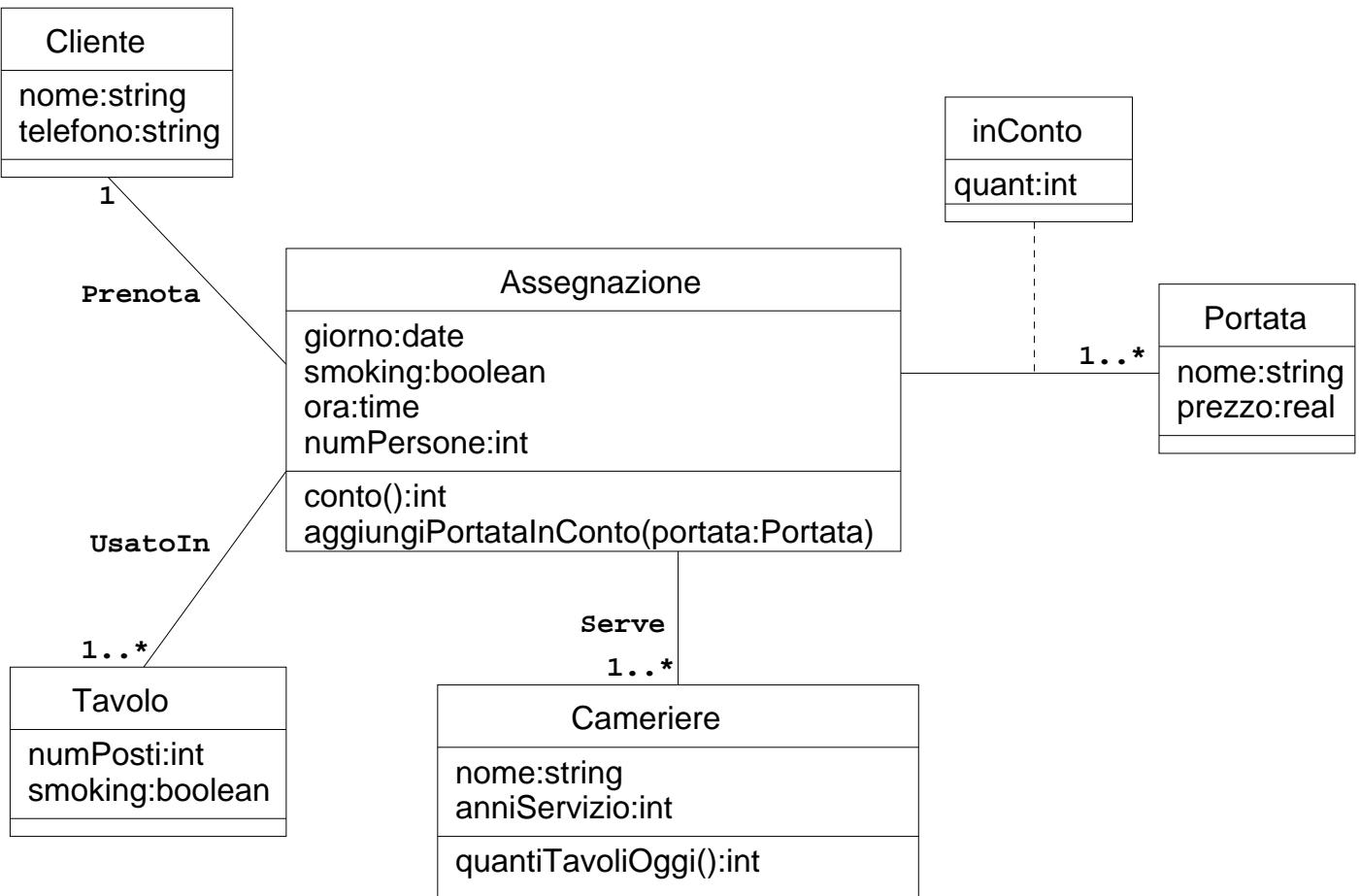
QuantiTavoliOggi(c:Cameriere, oggi:data):intero >= 0

Pre: Le stesse di c.quantiTavoliOggi()

Post: result è pari a c.quantiTavoliOggi(oggi)

(Viene invocata l'operazione c.quantiTavoliOggi(oggi))

FineSpecificoUseCase



InizioSpecificoClasse Assegnazione

aggiungiPortataInConto(n:int>0, portata:Portata)

Pre:

true // Nessuna precondizione

Post:

$(pre(<\text{portata}, \text{this}> \notin \text{InConto})) \rightarrow (<\text{portata}, \text{this}> \in \text{InConto} \wedge \text{InConto.quant}(<\text{portata}, \text{this}>) = n) \wedge$
 $//$ Se non esiste un link *InConto* tra *this* e *portata*, ne viene creato uno con attributo *quant* pari ad *n*...
 $(pre(<\text{portata}, \text{this}> \in \text{InConto})) \rightarrow (\text{InConto.quant}(<\text{portata}, \text{this}>) =$
 $pre(\text{InConto.quant}(<\text{portata}, \text{this}>) + n)$
 $//$...altrimenti il valore del campo *quant* del link già presente viene incrementato di *n*.

conto():valuta

Pre:

true // Nessuna precondizione.

Post:

$IC = \{<p, \text{this}> | p \in \text{Portata} \wedge <p, \text{this}> \in \text{InConto}\}$
 $//$ Detto *IC* l'insieme dei link di tipo *InConto* che collegano *this* ad oggetti di classe *Portata*
 $\text{result} = \sum_{<p, \text{this}> \in IC} \text{InConto.quant}(<p, \text{this}>) \times \text{Portata.prezzo}(p)$
 $//$ *result* è pari alla somma del numero di portate servite, ciascuna moltiplicata per il proprio prezzo.

FineSpecificoClasse

```
InizioSpecificaClasse Cameriere
    quantiTavoliOggi():intero >= 0
```

Pre:

true // Nessuna precondizione

Post:

$T = \{t | t \in Tavolo \wedge (\exists a \in Assegnazione \wedge Assegnazione.giorno(a) = oggi \wedge \langle t, a \rangle \in UsatoIn \wedge \langle this, a \rangle \in Serve)\}$

// Detto T l'insieme dei tavoli serviti oggi dal cameriere **this**,

result = $|T|$

// **result** è pari alla cardinalità di T .

```
FineSpecificaClasse
```