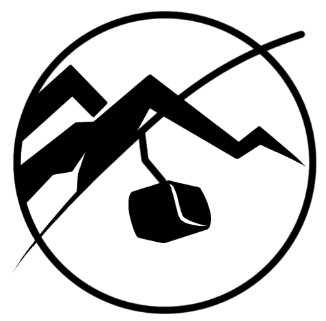
****

**PROGETTO DI INGEGNERIA DEL SOFTWARE**

**SKI ONLINE**

**Analisi dei requisiti**

# Indice dei contenuti

[Indice dei contenuti 2](#_Toc118970639)

[Scopo del documento 2](#_Toc118970640)

[Diagramma delle classi 2](#_Toc118970641)

**Scopo del documento**

Il presente documento riporta la definizione dell’architettura del progetto SKI ONLINE usando diagrammi delle classi in Unified Modeling Language (UML) e codice in Object Constraint Language (OCL). Nel precedente documento è stato presentato il diagramma degli use case, il diagramma di contesto e quello dei componenti. Ora, tenendo conto di questa progettazione, viene definita l’architettura del sistema dettagliando da un lato le classi che dovranno essere implementate a livello di codice e dall’altro la logica che regola il comportamento del software. Le classi vengono rappresentate tramite un digramma delle classi in linguaggio UML. La logica viene descritta in OCL perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

# Diagramma delle classi

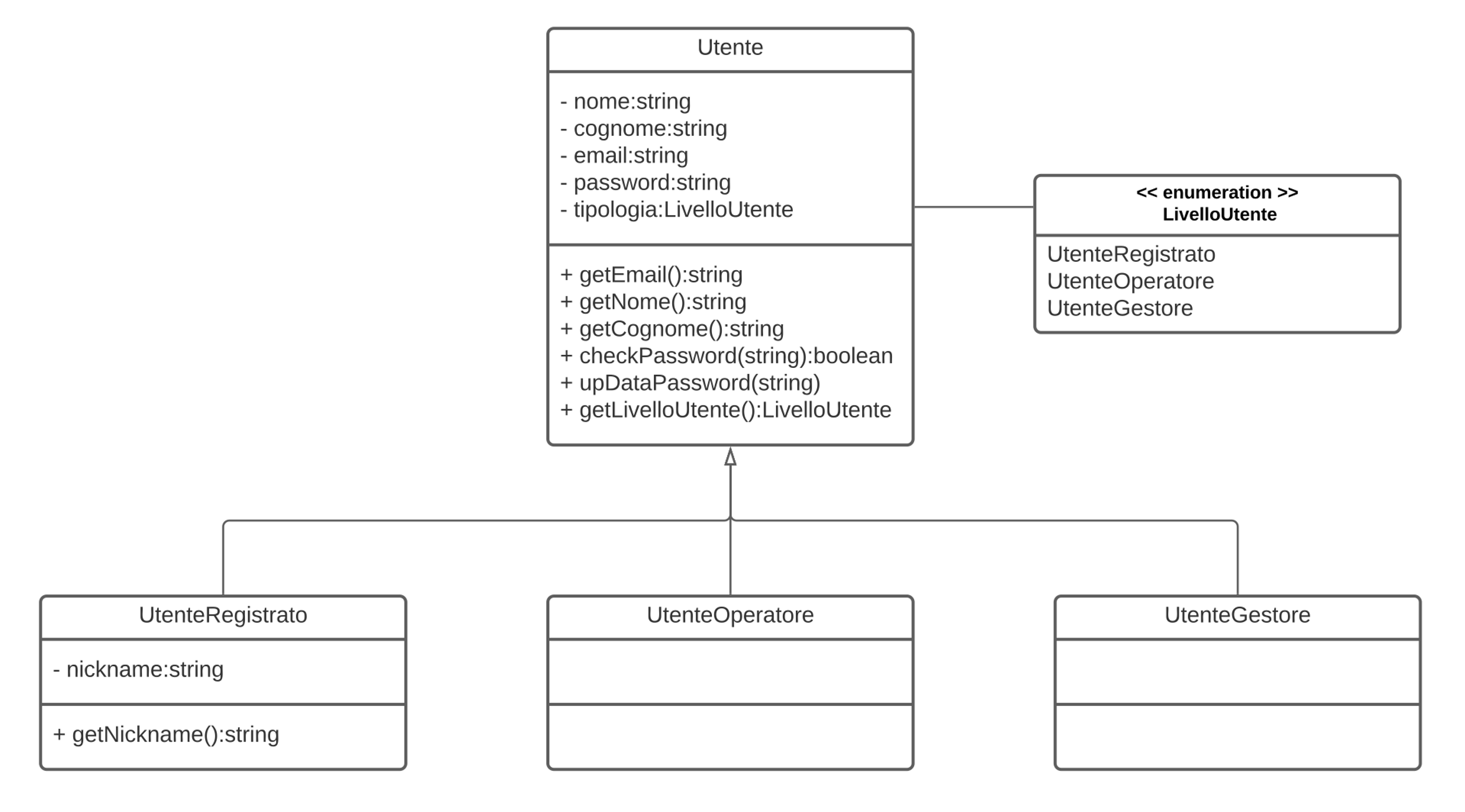
Nel presente capitolo vengono presentate le classi previste nell’ambito del progetto SKI ONLINE. Ogni componente presente nel diagramma dei componenti diventa una o più classi. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi che identificano i dati gestiti dalla classe e una lista di metodi che definiscono le operazioni previste all’interno della classe. Ogni classe può essere anche associata ad altre classi e, tramite questa associazione, è possibile fornire informazioni su come le classi si relazionano tra loro.

Riportiamo di seguito le classi individuate a partire dai diagrammi di contesto e dei componenti. In questo processo si è proceduto anche nel massimizzare la coesione e minimizzare l’accoppiamento tra classi.

È descritta in modo formale la logica prevista nell’ambito di alcune operazioni di alcune classi. Tale logica viene descritta in Object Constraint Language (OCL) perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

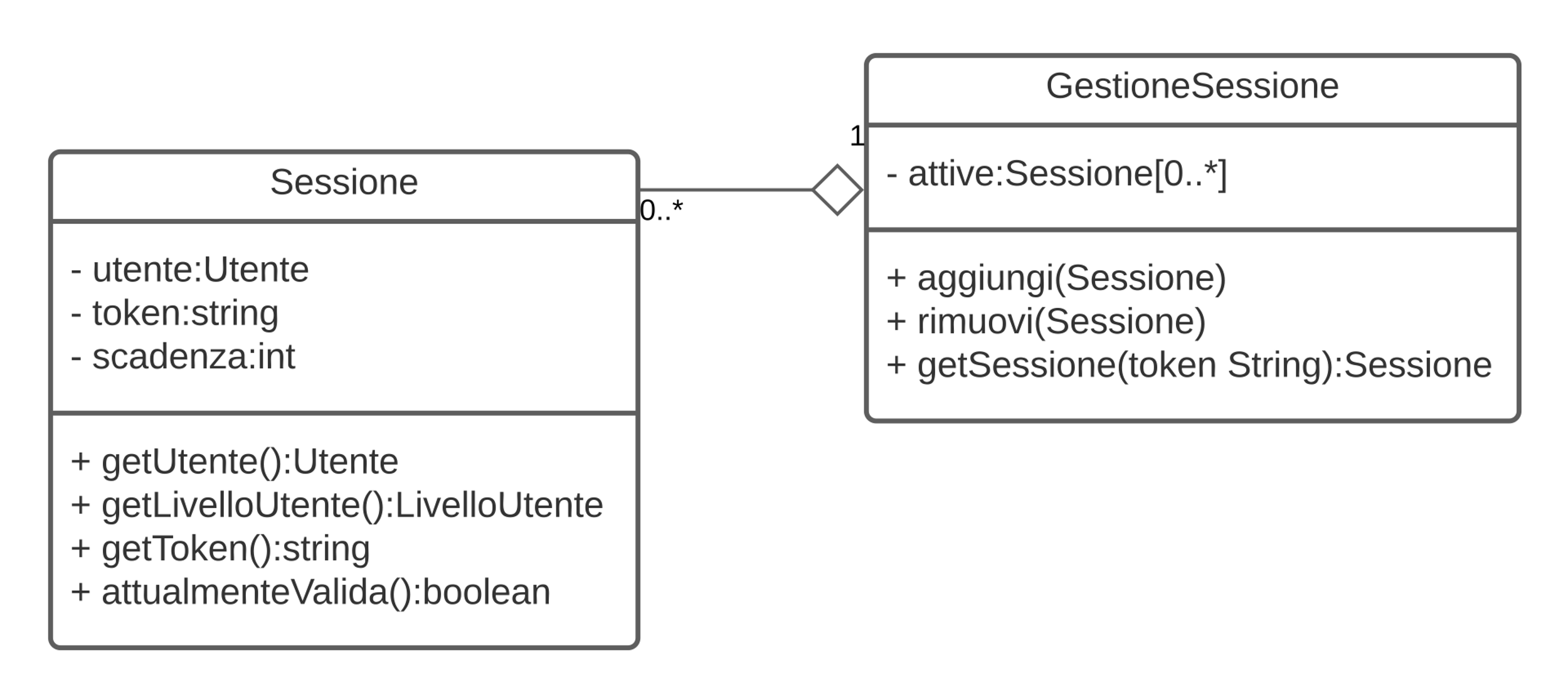
## Utenti del sistema

Analizzando i vari livelli di utenza che sono previsti nel sistema, ovvero: “Utente Registrato”, “Utente Operatore” e “Utente Gestore”, sono state identificate tre classi corrispondenti ai tre livelli di utenza, accomunate da una classe madre comune “Utente”. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



## Gestione sessione

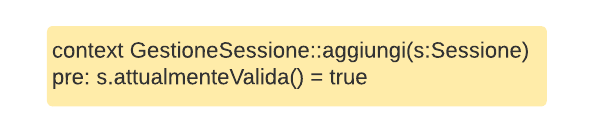
Analizzando la componente “Gestione sessione” questa è stata scomposta in due classi “Sessione” e “Gestione sessione”, la prima si occupa di manipolare i dati di una singola sessione mentre la seconda di raccogliere e gestire tutte le sessioni attualmente attive. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*Il datatype “Utente” e l’enumeration “LivelloUtente” fanno riferimento a quelli definiti precedentemente*

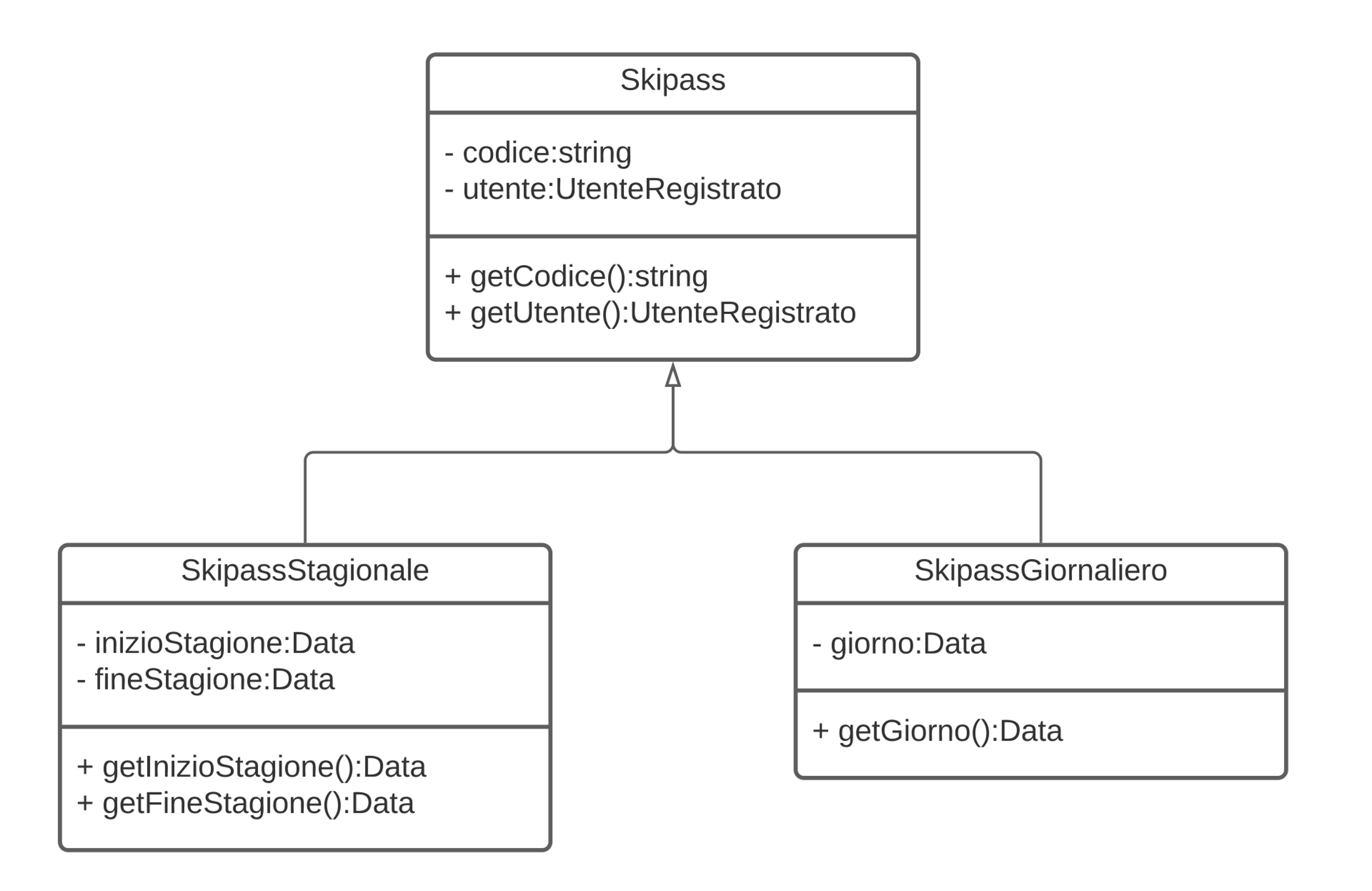
### Specifica OCL

Per poter aggiungere una nuova sessione all’elenco delle sessioni attive, questa deve essere valida (ovvero non ancora scaduta). Questa condizione sulla classe GestioneSessione è espressa in OCL attraverso una precondizione con questo codice



## Skipass

Analizzando le due diverse tipologie di skipass acquistabili, ovvero “Skipass giornaliero” e “Skipass stagionale”, sono state identificate due classi corrispondenti alle due tipologie di skipass, accomunate da una classe madre “Skipass” comune. Di seguito il dettaglio di queste classi con i propri attributi e metodi.





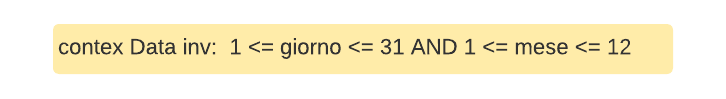
### Specifica OCL

Ogni skipass stagionale ha un periodo di validità a partire dall’inizio della stagione fino alla sua fine, per questo motivo una data di fine della stagione antecedente a quella di inizio non avrebbe senso logico. Questa condizione sulla classe SkipassStagionale è espressa in OCL attraverso una invariante con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

Inoltre, per quanto riguarda il datatype Data, il valore numerico del giorno e del mese devono essere coerenti con i valori ammissibili da calendario. Questa condizione è espressa in OCL attraverso una invariante con questo codice:



## Gestione maestri

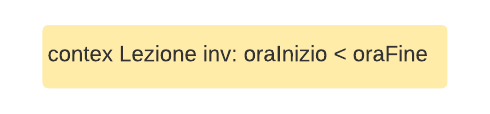
Analizzando le componenti “Gestione maestri” e “Grafica gestione stato maestri”, sono state identificate quattro classi, due direttamente correlate alle due componenti, mentre le rimanenti “Lezione” e “Maestro” derivate dalla tipologia di dati che le prime due utilizzano. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*Il tipo di dato “Gestione sessione” si riferisce alla classe definita precedentemente*

### Specifica OCL

Ogni lezione ha un orario di inizio in cui il maestro e l’allievo si incontrano e un orario di fine nel quale si salutano, per questo motivo un orario di fine antecedente all’orario di inizio non avrebbe senso logico. Questa condizione è espressa in OCL attraverso una invariante con questo codice:



Un maestro ha uno stato di disponibilità ad effettuare lezioni rappresentato da un valore booleano, attraverso il metodo modifica disponibilità questo valore viene cambiato passando da disponibile a non disponibile oppure viceversa. Questa condizione è espressa in OCL attraverso una postcondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

Quando la classe Gestore maestri viene incaricata di prenotare una lezione, attraverso il suo metodo prenotaLezione, il maestro associato alla lezione deve avere la disponibilità discussa nel punto precedente a valore true (ovvero disponibile) al momento della registrazione nel calendario. Questa condizione è espressa in OCL attraverso una precondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

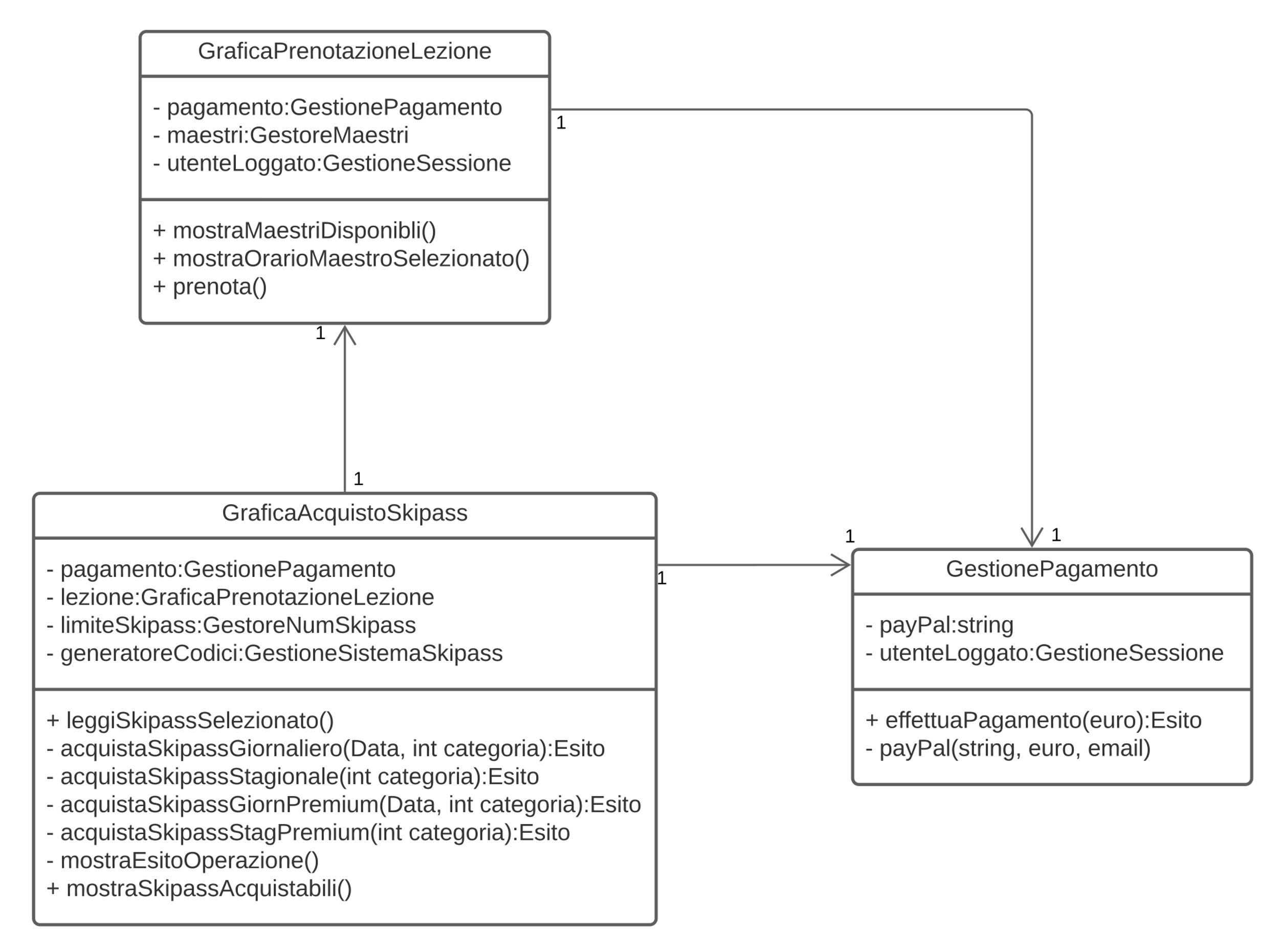
Infine ogni metodo disponibile nella classe GraficaGestioneStatoMaestri può essere eseguito solo da un utente di livello “Utente Gestore”. Questa condizione è espressa in OCL attraverso una precondizione con questo codice, per semplicità è stata riportata la condizione solo su mostraMaestri, ma sono da intendersi altre due identiche su aggiornaStatoMaestroSelezionato e leggiNomeMaestroDaAggiungere:

Immagine che contiene testo

Descrizione generata automaticamente

## Acquisto skipass e prenotazione lezioni

Analizzando le componenti “Grafica acquisto skipass”, “Gestione pagamento” e “Grafica prenotazione lezione”, si è proceduto a identificare tre classi con le medesime funzionalità. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*I tipi di dato “Gestione sessione”, “Gestione maestri” ed “Esito” si riferiscono alle classi (ed enumeration) definite precedentemente*

*I tipi di dato “Gestione numero skipass” e “Gestione sistema skipass” si riferiscono a delle classi che verranno definite successivamente*

### Specifica OCL

Quando la classe GraficoAcquistoSkipass viene incaricata di acquistare uno skipass, attraverso uno dei quattro metodi acqustaSkipass, bisogna verificare che la richiesta di pagamento a GestionePagamento sia andata a buon fine. In caso affermativo la classe deve richiamare il metodo per la generazione dello skipass in GestioneNumeroSkipass corrispondente. Inoltre nel caso di acquisto di skipass giornalieri bisogna procedere con l’acquisto solo se non è stato raggiunto il limite di vendite in un dato giorno. Queste condizioni sono espresse in OCL attraverso una precondizione e una postcondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

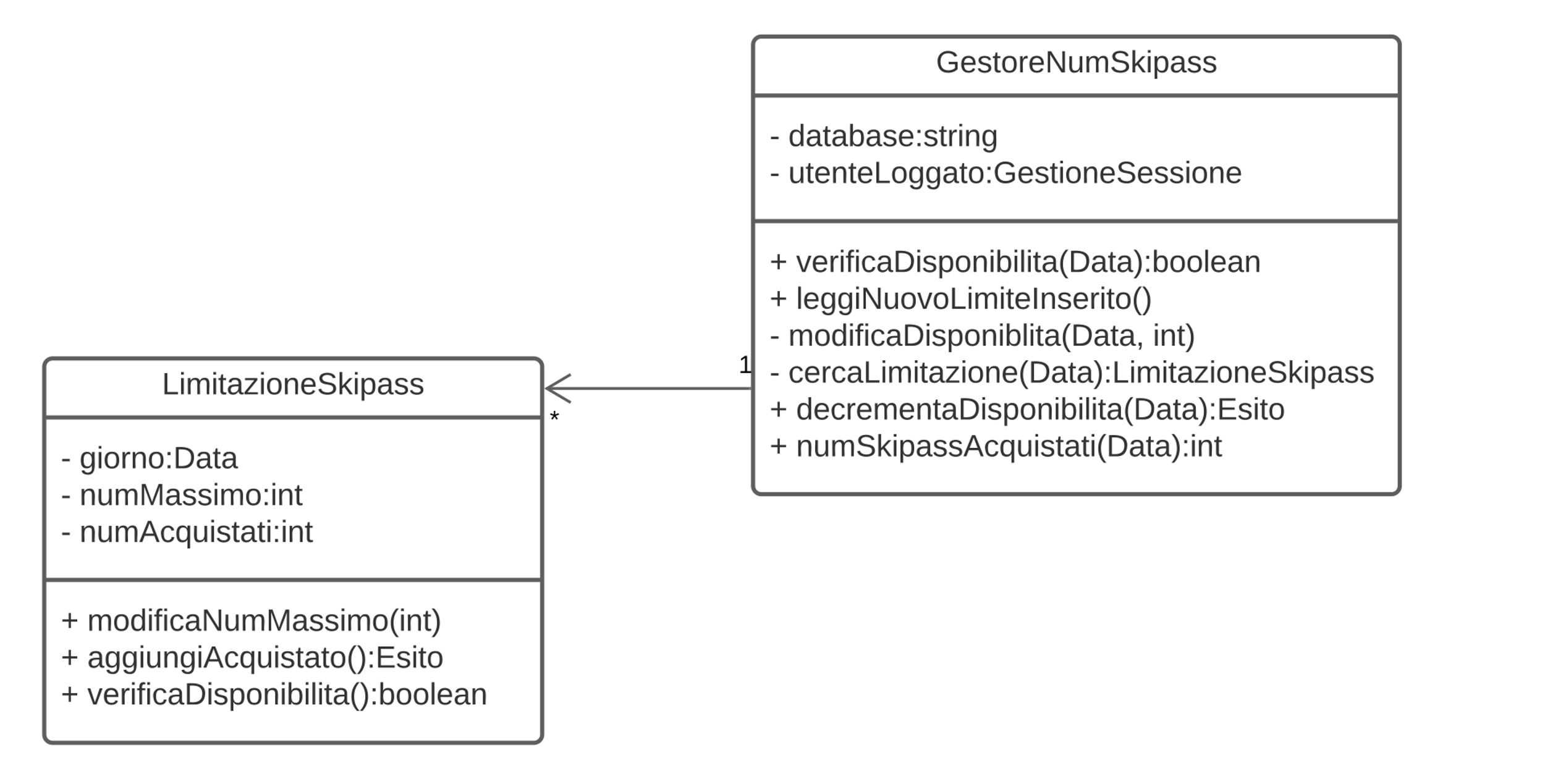
Inoltre ogni metodo disponibile nella classe GraficaPrenotazioneGestione può essere eseguito solo da un utente di livello “Utente Registrato”. Questa condizione è espressa in OCL attraverso una precondizione con questo codice, per semplicità è stata riportata la condizione solo su mostraMaestriDisponibili, ma sono da intendersi altre due identiche su mostraOrarioMaestroSelezionato e prenota:

Immagine che contiene testo

Descrizione generata automaticamente

## Limitazione numero skipass acquistabili

Analizzando la componente “Gestione numero skipass acquistabili” questa è stata scomposta in due classi “GestoreNumeroSkipassAcquistabili” e “LimitazioneSkipass”, la prima si occupa di aggiungere, rimuovere e modificare limitazioni in diverse giornate, mentre la seconda si occupa di manipolare la singola limitazione in una giornata.



*I tipi di dato “Data”, “Gestione sessione” ed “Esito” si riferiscono alle classi definite precedentemente*

### Specifica OCL

Nella classe LimitazioneSkipass l’attributo numMassimo indica il numero massimo di skipass acquistabili nel giorno specificato nell’attributo corrispondente, mentre l’attributo numAcquistati indica il numero di skipass già acquistati sempre nel giorno indicato. Per questo motivo un numero massimo di skipass acquistabili minore del numero di skipass effettivamente acquistati genererebbe una evidente contraddizione e non si deve poter verificare. Questa condizione è espressa in OCL attraverso una invariante con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

Sempre nella classe LimitazioneSkipass quando si richiede di incrementare il numero di skipass già acquistati bisogna verificare che la disponibilità sia ancora positiva e il numero deve essere subito aggiornato, onde evitare problemi di accesso concorrente. Queste condizioni sono espresse in OCL attraverso una precondizione e una postcondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

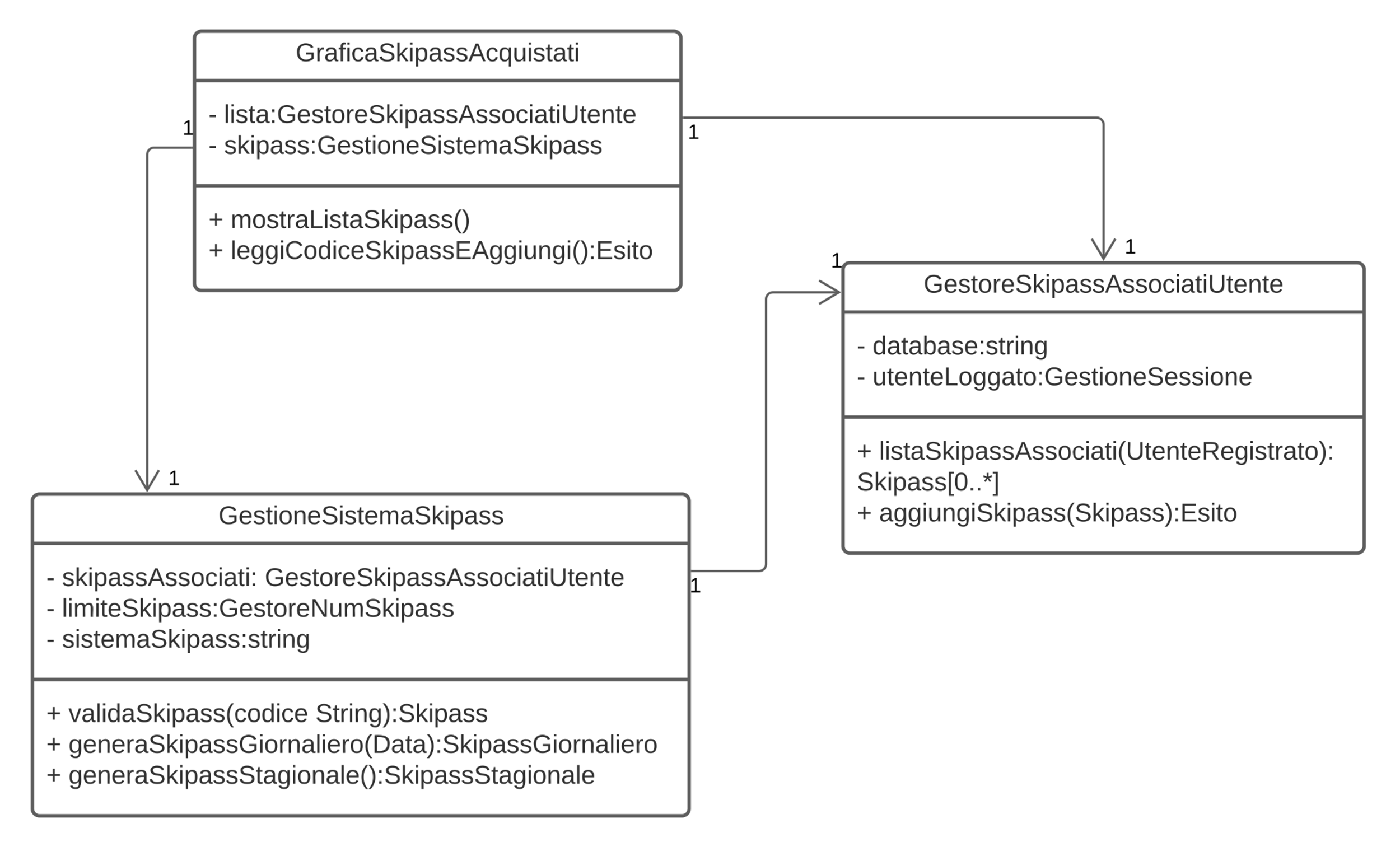
Infine nella classe GestoreNumSkipass la modifica della disponibilità del numero massimo di skipass acquistabili in un giorno può essere effettuata solo da un utente di livello “Utente Gestore”, per questo motivo il metodo leggiNuovoLimiteInserito (e di conseguenza modificaDisponibilità che viene richiamato) può essere eseguito solo da un utente gestore. Questa condizione è espressa in OCL attraverso una precondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

## Gestione skipass associati ad una utenza

Analizzando le componenti “Gestione sistema skipass preesistente”, “Grafica skipass acquistati” e “Gestione skipass associati ad una utenza” si è proceduto a identificare tre classi con le medesime funzionalità. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*I tipi di dato “GestoreNumSkipass”, “Gestione sessione”, “Utente Registrato”, “Esito”, “Data”, “Skipass”, “Skipass giornaliero”, “Skipass stagionale” si riferiscono alle classi definite precedentemente*

### Specifica OCL

Quando la classe GestioneSistemaSkipass viene incaricata di validare un codice skipass, oppure di generare uno skipass stagionale, se l’esito della validazione nel primo caso, l’esito della generazione nel secondo, è andato a buon fine è necessario che lo skipass venga aggiunto al database e associato all’utente, attraverso il metodo aggiungiSkipass del Gestore Skipass Associati Utente. Questa condizione è espressa in OCL attraverso una postcondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

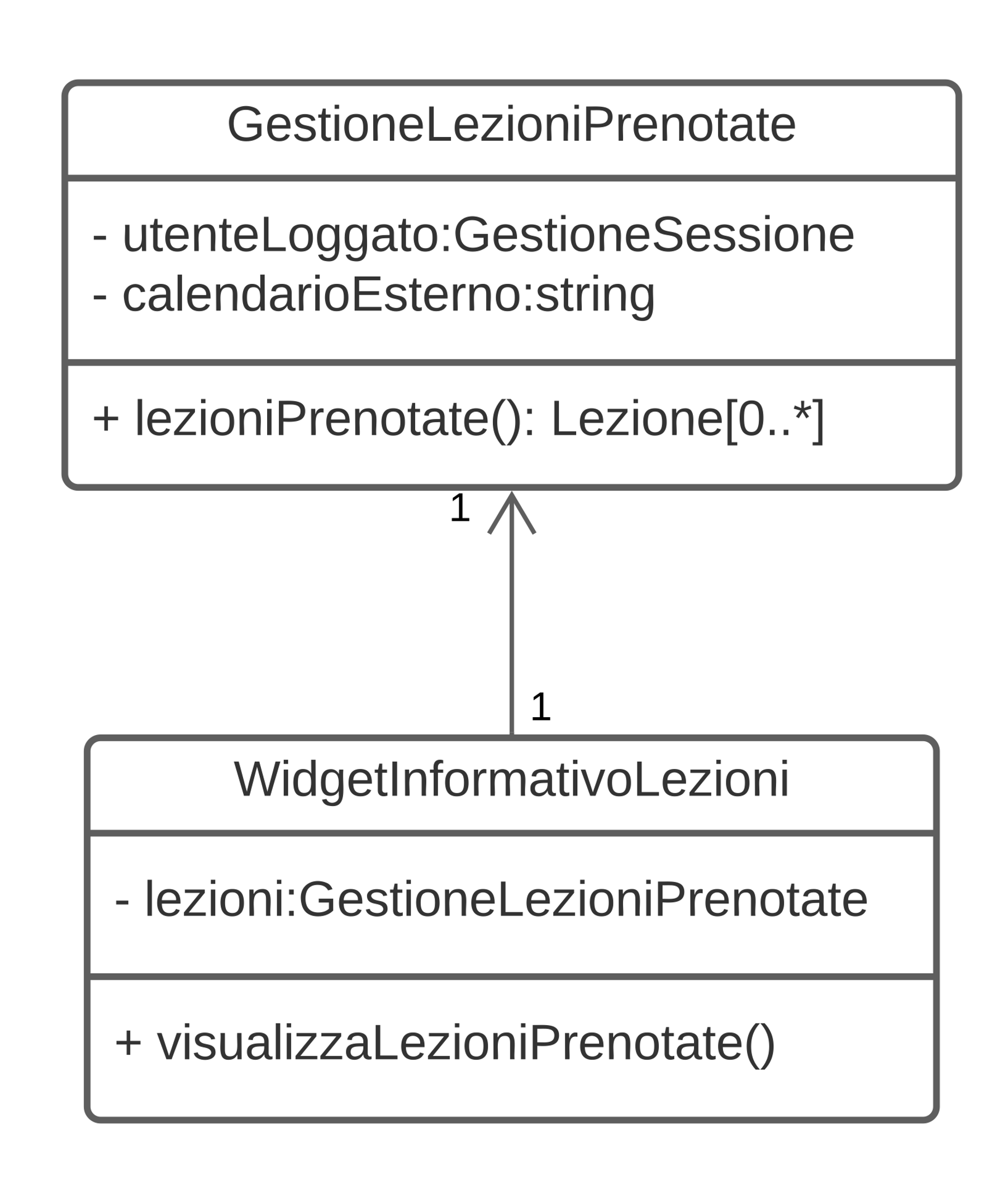
Inoltre, sempre la classe GestioneNumSkipass, nel caso in cui venga incaricata di generare uno skipass giornaliero, oltre alla postcondizione descritta precedentemente lo skipass ottenuto deve essere valido nel giorno richiesto all’utente e la generazione può essere effettuata solo se la richiesta di incremento del numero di skipass già acquistati è terminata correttamente. Queste condizioni sono espresse in OCL attraverso una precondizione e una postcondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

## Widget informativo lezioni

Analizzando le componenti “Widget informativo lezioni” e “Gestione lezioni prenotate” si è proceduto a identificare due classi con le medesime funzionalità. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*I tipi di dato “Lezione” e “Gestione sessione” si riferiscono alle classi definite precedentemente*

### Specifica OCL

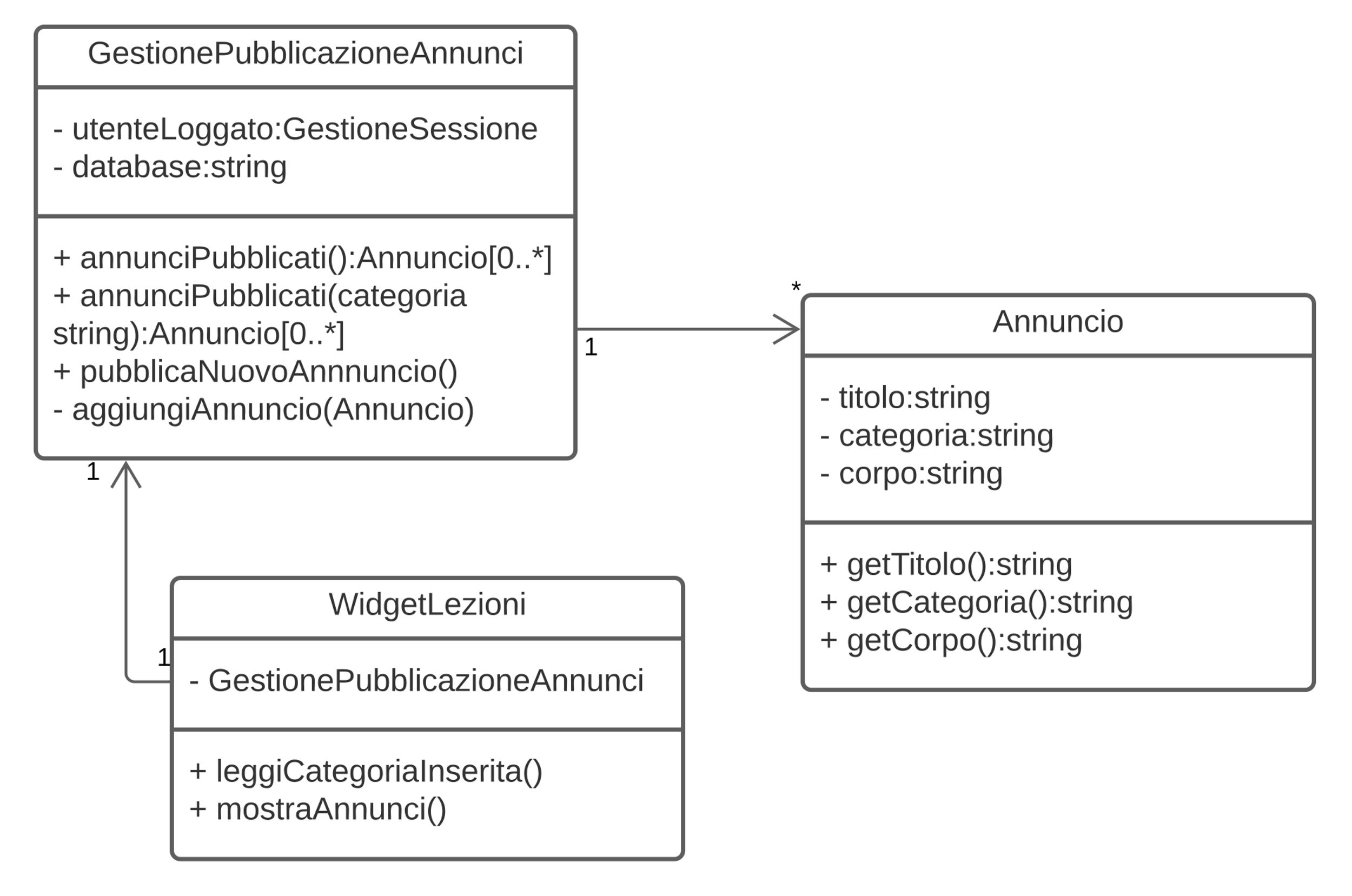
Il metodo lezioniPrenotate della classe GestioneLezioniPrenotate può essere eseguito solamente da un utente di livello “Utente Registrato”. Questa condizione è espressa in OCL attraverso una precondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

## Widget annunci pubblicati

Analizzando le componenti “Widget annunci” e “Gestione pubblicazione annunci” si è proceduto a identificare due classi con le medesime funzionalità. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



*Il tipo di dato “Gestione sessione” si riferisce alla classe definite precedentemente*

### Specifica OCL

Un Annuncio deve contenere per forza un titolo, che lo identifica, e un corpo, mentre il campo categoria è opzionale e quindi non obbligatorio nell’inserimento. Questa condizione è espressa in OCL attraverso una invariante con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

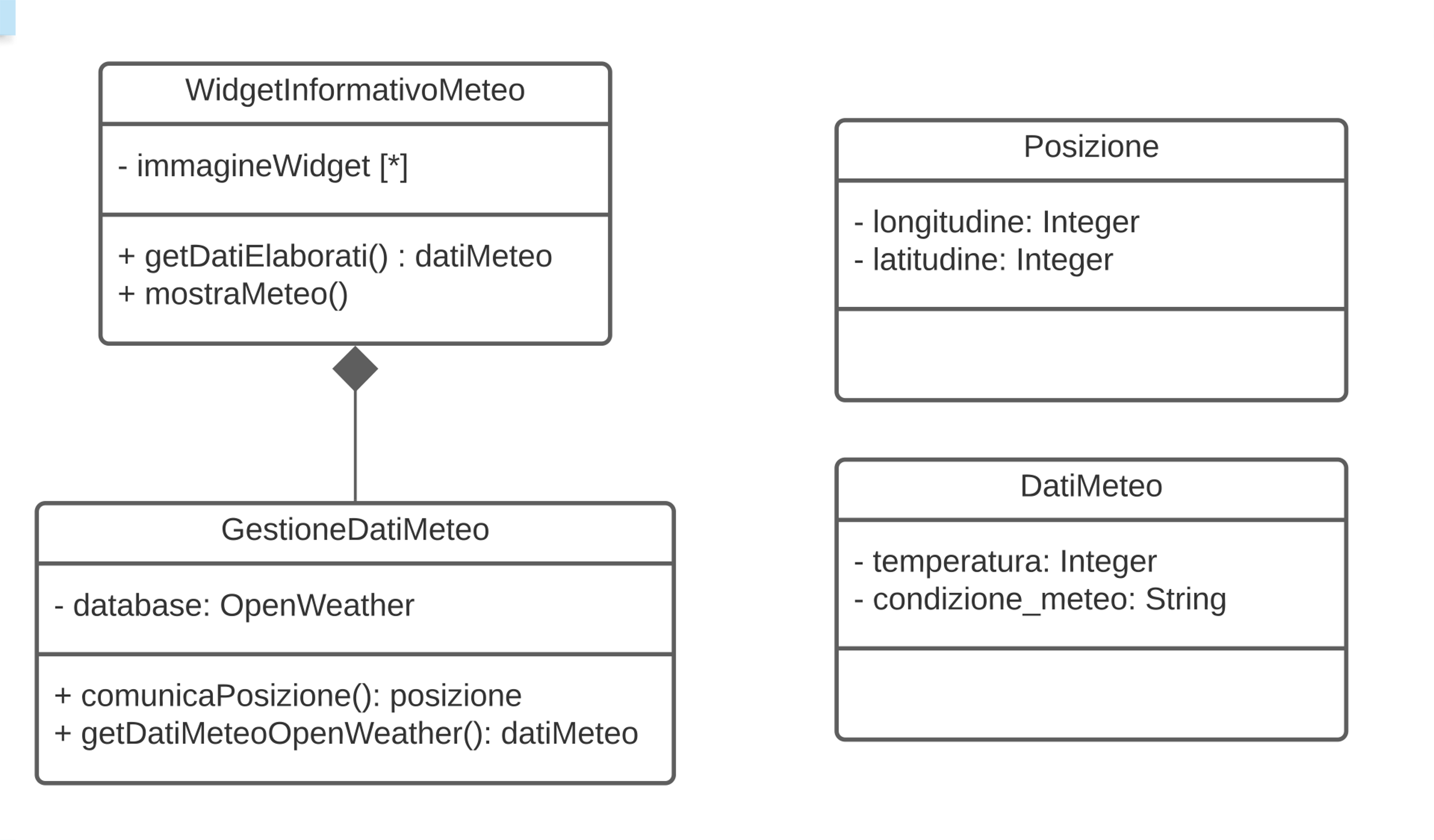
Inoltre il metodo pubblica nuovo annuncio (e di conseguenza aggiungiAnnuncio che viene richiamato) può essere eseguito solo da un utente di livello “Utente Gestore”. Questa condizione è espressa in OCL attraverso una precondizione con questo codice:

Immagine che contiene testo

Descrizione generata automaticamente

## Gestione widget meteorologico

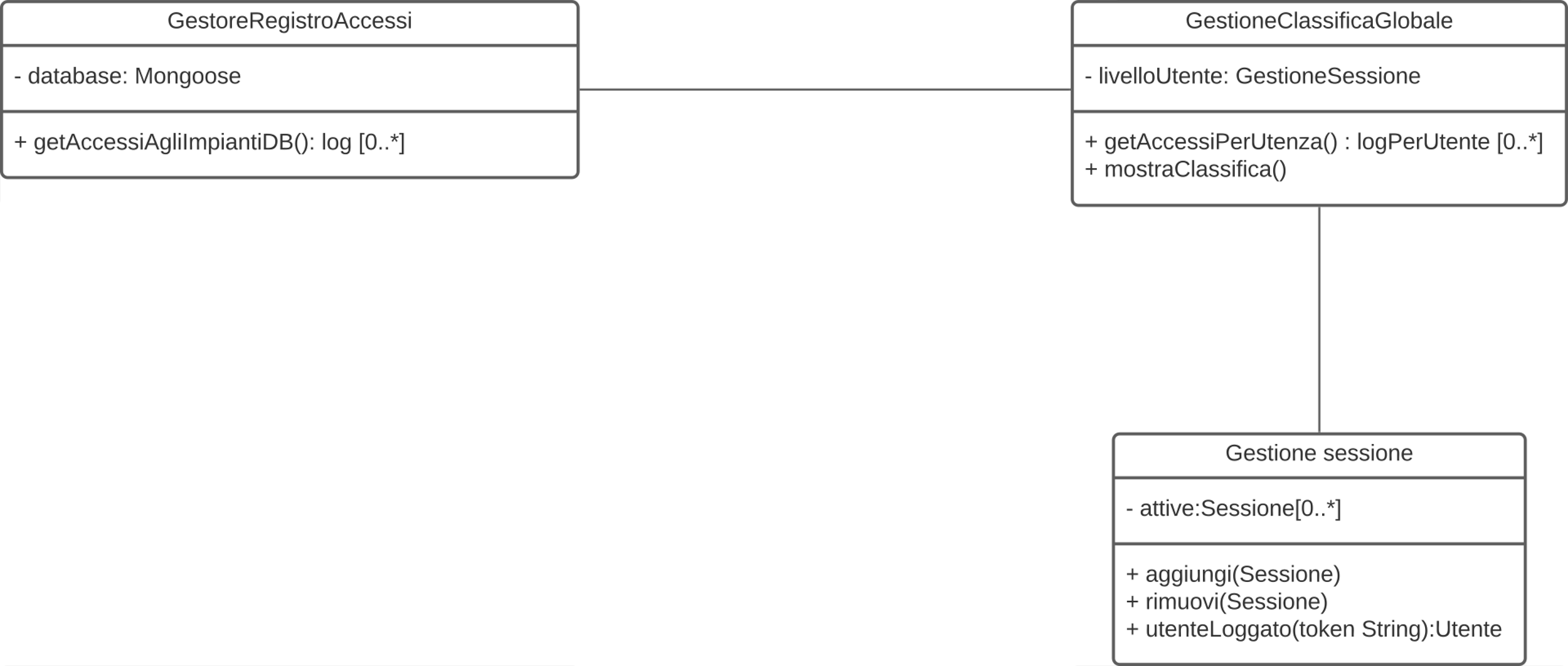
Analizzando le componenti “Widget informativo meteo” e “Gestione dati meteo” queste sono state identificate come classi per la gestione e visualizzazione del widget dedicato alla visione delle condizioni meteorologiche. Di seguito il dettaglio di queste classi con i propri attributi e metodi.



## Gestione stato impianti

## Gestione classifica Globale

Analizzando la componente “Gestione classifica Globale” è stata identificata come classe responsabile della visualizzazione della classifica globale. Definiamo ora la classe e le relative classi ad essa collegate.



## Gestione storico impianti