

Progetto Metodologie di Programmazione

Aliberto Manuel 5645310 manuel.aliberto@stud.unifi.it

Ghera Matteo 5616343 matteo.ghera@stud.unifi.it

D'Autilia Mattia 5765968 mattia.dautilia@stud.unifi.it

Testo dell'esercizio scelto

Ogni giorno/mese in un'azienda bisogna calcolare (e inviare) la paga ad ogni impiegato. Ci sono diversi tipi di impiegati (1) impiegati pagati ad ora, (inviando a fine giornata il loro tesserino su cui sono indicate le ore di lavoro) (2) impiegati con salario fisso (3) impiegati pagati con commissione sulle vendite effettuate (4) impiegati che ricadono nell'intersezione di due categorie, ad esempio, di (2) e (3). Ovviamente, i dati suddetti possono essere modificati: ad esempio, la classificazione di un impiegato. Ogni impiegato può scegliere un particolare metodo per ricevere il pagamento (assegno, bonifico, etc...).

C'è uno scheduling dei pagamenti: ogni impiegato è pagato con tempistiche diverse, ad es., quelli ad ora e sono pagati ogni settimana, quelli a stipendio sono pagati ogni mese, etc...

Modello reale

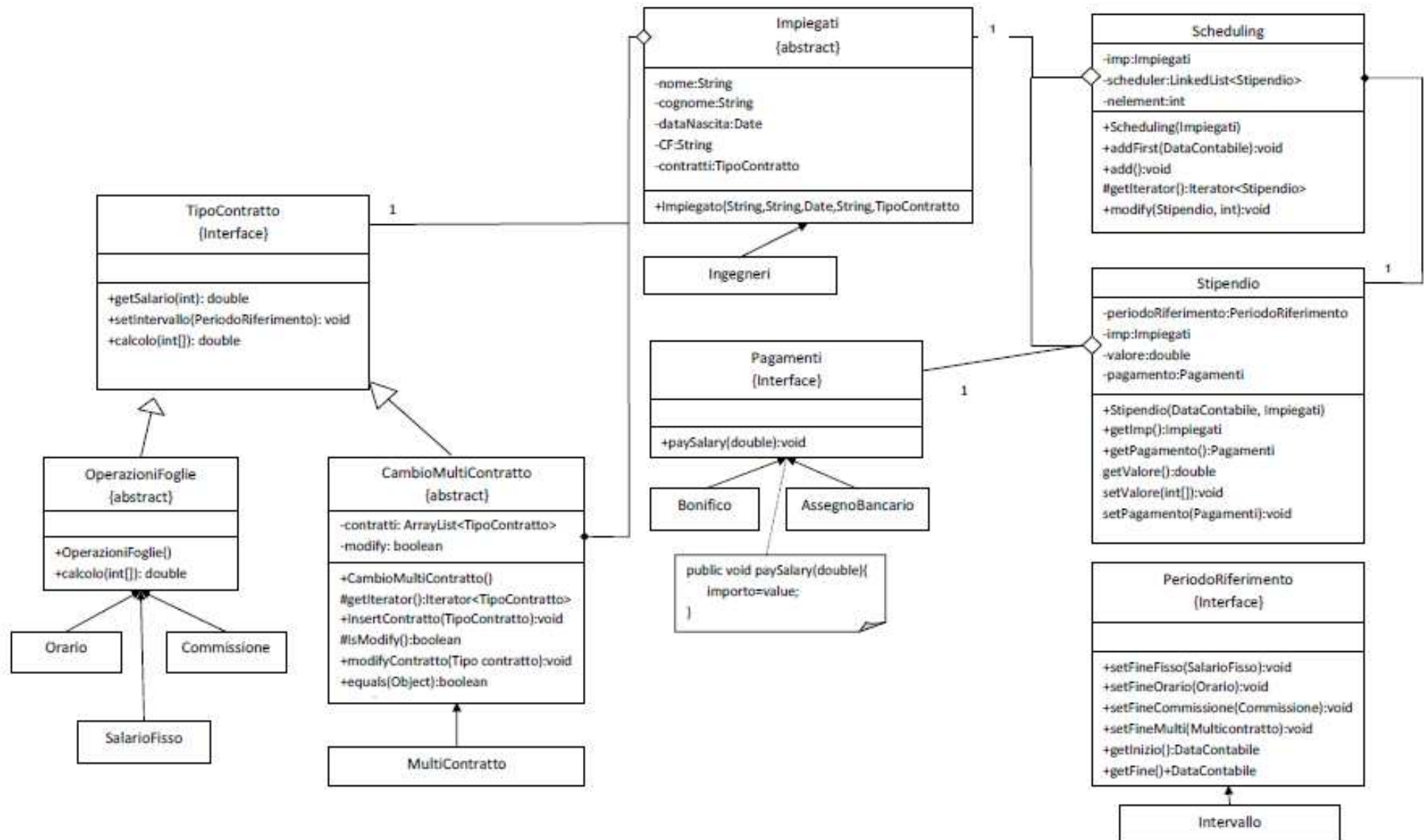
Durante la realizzazione del progetto abbiamo fatto riferimento al seguente modello per creare un framework più realistico. Consideriamo il caso di un'azienda sviluppatrice di software. Nell'azienda ci sono diversi tipi di impiegati: gli ingegneri, i quali progettano il software; i programmatori, che lo sviluppano; i tester, che ne verificano le funzionalità. Il gruppo che lavora ad un progetto percepisce uno stipendio base e, qualche volta, una commissione su specifici lavori extra (ad es. la consulenza di un esperto esterno).

Specifiche del sistema

Durante la realizzazione del progetto abbiamo adottato le seguenti specifiche:

- Per risolvere il problema relativo ai contratti che ricadono nell'intersezione di due categorie abbiamo previsto una classe Multicontratto. Possiamo creare anche oggetti di classe che si basano su altri oggetti di Multicontratto; infatti nel modello che abbiamo utilizzato per la realizzazione del progetto si prevede che un impiegato (ingegnere, programmatore...) possa essere coinvolto in diversi progetti con paghe diverse che potrebbero richiedere alcuni lavori extra;
- Un impiegato durante la sua carriera lavorativa percepisce diversi compensi. Per questo abbiamo previsto la possibilità di visualizzare per ogni impiegato tutti gli stipendi percepiti con i rispettivi pagamenti (assegno o bonifico);
- Gli impiegati che hanno un contratto a salario fisso, commissione oppure che ricadono nell'intersezione di questi due vengono pagati ogni mese invece quelli con contratto ad ora sono pagati ogni settimana;
- Un impiegato può essere pagato con un bonifico oppure con un assegno bancario;
- Ogni tipo di impiegato (ingegneri, programmatori,...) ha un tipo di contratto e di paga definita per default;

Diagramma delle classi in U.M.L.



Design del progetto

Il progetto è stato organizzato in quattro diversi package: Contratti, Impiegati, Pagamenti e Stipendi. Questi ospitano le classi che gestiscono rispettivamente: i contratti che differenziano i vari impiegati a seconda della categoria/tipo di contratto assegnatoli, le mansioni assegnate ai vari impiegati (ingegnere, programmatore...), il pagamento degli stipendi e la gestione e memorizzazione dei vari stipendi degli impiegati.

Nel progetto ci sono le seguenti principali classi astratte:

1. l'interfaccia TipoContratto comune a tutte le diverse categorie/tipi di contratto che possono essere assegnate a ciascun impiegato. Questa interfaccia definisce l'operazione comune a tutte le classi che la implementano: il metodo che calcola l'importo dello stipendio sulla base delle ore lavorate;
2. la classe astratta Impiegati che definisce le operazioni e le informazioni comuni per ogni tipo di mansione che può essere associata agli impiegati;
3. l'interfaccia Pagamenti comune a tutti i tipi di pagamenti che possono essere utilizzati per il pagamento dello stipendio;

Nel package Contratti abbiamo utilizzato il design pattern Composite per la gestione delle diverse categorie/tipologie di contratto associabili a ciascun impiegato. Le classi che gestiscono le seguenti tipologie di contratto possono essere viste come foglie di un Composite: il tipo di contratto a salario fisso (SalarioFisso), il tipo di contratti a paga oraria (Orario), il tipo di contratti su commissione (Commissione); come composto abbiamo la classe MultiContratto la quale gestisce i tipi di contratto che ricadono tra l'intersezione delle tipologie elencate sopra.

Ogni stipendio ha associato il suo periodo di riferimento, cioè un intervallo settimanale o mensile a cui lo stipendio si riferisce. Il periodo di riferimento dipende dal tipo di contratto assegnato all'impiegato. Per la gestione dei periodi di riferimento associati a ciascuno stipendio in base al tipo di contratto dell'impiegato abbiamo utilizzato il design patter Visitor. In particolare, la classe PeriodoRiferimento è l'interfaccia Visitor del pattern, la classe Intervallo (che implementa PeriodoRiferimento) è il ConcreteVisitor, mentre dall'interfaccia TipoContratto viene richiesto di implementare il metodo che accetta il visitatore.

Descrizione della parte implementata e dei test

Per quanto riguarda la parte implementata bisogna dire che il metodo getSalario della classe MultiContratto calcola l'importo totale dello stipendio sulla base dei contratti a lui associati. Tale importo è ottenuto aggiungendo al totale complessivo l'importo derivante dal contratto in esame e dal numero di ore lavorate, cioè per ogni tipo di contratto di MultiContratto si calcola il valore associato al tipo di contratto in base al numero di ore passate e si aggiunge al totale dei precedenti. Inoltre nella classe Scheduling abbiamo due costruttori: il primo, in genere, viene utilizzato la prima volta che viene inserito uno stipendio (richiede di inserire la data d'inizio del periodo di riferimento); il secondo viene utilizzato tutte le volte successive, la data di inizio del periodo di riferimento dipende dalla data di fine del periodo di riferimento associata allo stipendio precedente. Infine, bisogna sottolineare che gli stipendi non ancora pagati sono caratterizzati dal campo relativo al pagamento nullo.

Per ogni classe abbiamo testato le funzionalità più importanti: per le classi che implementano l'interfaccia TipoContratto abbiamo testato il metodo che calcola il valore degli stipendi, inoltre per le due classi che

gestiscono i tipi di contratti composti abbiamo testato i metodi per l'aggiunta e la modifica dei contratti; per la classi Stipendi abbiamo verificato che per ogni oggetto di Stipendio creato venisse restituito il totale complessivo dello stipendio adeguato al numero di ore lavorate; per la classe Scheduling abbiamo testato i metodi di aggiunta, modifica e selezione.