

SAPIENZA Università di Roma
Facoltà di Ingegneria – Corso di Laurea in Ingegneria Informatica
Corso di Progettazione del Software
Esame del **13 Luglio 2009**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di dipendenti di un'azienda. Ciascun dipendente ha un nome ed una email, e presta normalmente servizio in una sede di lavoro, della quale interessa l'indirizzo. Sono poi di interesse le giornate lavorative, con la data a cui fanno riferimento e eventualmente una nota (una stringa) che riporti particolarità della giornata. Durante ogni giornata, al fine di monitorarne l'effettiva presenza in sede, i dipendenti che lavorano in quella giornata devono timbrare il loro cartellino, sia in ingresso che in uscita. Ciascuna timbratura, sia di ingresso che di uscita, avviene in una certa ora e presso una delle sedi di lavoro (non obbligatoriamente quella presso cui il dipendente lavora normalmente), ed è ovviamente legata al dipendente che la effettua e alla giornata a cui si riferisce. Di ciascuna timbratura di uscita interessa anche memorizzare la timbratura di ingresso a cui è associata¹. Si osservi che i dipendenti possono effettuare più timbrature di ingresso e di uscita durante una giornata (ad esempio, per uscire a prendere un caffè, il dipendente deve timbrare in uscita e nuovamente al rientro).

Siamo interessati a progettare l'attività di controllo sui dipendenti, che prende come parametro un insieme non vuoto D di dipendenti, ed un insieme non vuoto G di giornate lavorative e procede come segue. Inizialmente scarta i dipendenti in D che non hanno lavorato in nessuna giornata di G . Poi partiziona l'insieme risultante dei dipendenti in coloro che sono "regolari" (insieme DR) e coloro che non lo sono (insieme DA): un dipendente è regolare se in ciascuna giornata di G in cui ha lavorato, ogni sua timbratura di ingresso ha una timbratura di uscita a cui è associata, altrimenti è "anomalo". Successivamente, ed in modo concorrente, l'attività opera sui due insiemi: (i) iterativamente viene letto da input una sede e viene calcolato per quella sede quanti dipendenti, che lavorano normalmente in quella sede, sono presenti in DR , ripetendo questi passi finché da input non vengono più richieste nuove sedi; concorrentemente (ii) viene calcolato per ciascun dipendente in DA il numero di giornate in G per il quale almeno una timbratura di ingresso non ha alcuna timbratura di uscita associata. Al termine di queste due sottoattività, si produce un report in output con i risultati ottenuti.

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare la fase di analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo del diagramma delle classi, diagramma delle attività, specifica delle attività atomiche che operano sul diagramma delle classi (i task), motivando, qualora ce ne fosse bisogno, le scelte effettuate. La specifica delle attività di I/O non è richiesta.

Domanda 2. Effettuare la fase di progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio definire le responsabilità su tutte le associazioni del diagramma delle classi ed il progetto dell'algoritmo dell'attività atomica (task) di divisione in "dipendenti regolari" e "dipendenti anomali".

Domanda 3. Effettuare la fase di realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- Le classi *Dipendente* e *Timbratura*, e le eventuali *associazioni* che le legano (non le eventuali sottoclassi).
- L'attività principale, le eventuali sottoattività non atomiche, l'attività atomica (task) di divisione in "dipendenti regolari" e "dipendenti anomali". Le altre sottoattività non vanno realizzate.

¹Si faccia attenzione che data una timbratura d'uscita deve riferirsi ad una timbratura d'ingresso effettuata dallo stesso dipendente nella stessa giornata per la stessa sede con un orario successivo. Questo vincolo non è rappresentabile nel Diagramma delle Classi UML, quindi lo si esprima con un opportuno commento nel diagramma.