

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica ed in Ingegneria dei Sistemi Informatici
Corso di Progettazione del Software
Esame del **4 febbraio 2016**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda l'attività giornaliera di gestione di pacchi presso un centro di smistamento. Un pacco ha un codice (una stringa), un cliente destinatario ed un cliente mittente. Un cliente è caratterizzato da un nome ed un indirizzo (una stringa). I pacchi sono smistati da operatori, tra cui uno che ne è il responsabile. Ogni operatore può essere responsabile di al più 50 pacchi. Gli operatori sono caratterizzati da un codice (una stringa). Alcuni pacchi sono pesanti e in tal caso possono essere o non essere approvati per lo smistamento (un booleano).

L'*operatore* è inizialmente in uno stato di riposo. Quando è a riposo e *riceve* un pacco (evento), ne diventa il responsabile; se il pacco non è pesante lo smista mandando un evento *smistato* (con payload il pacco) in broadcast; se invece è pesante manda evento *verifica* (con payload il pacco) ad un altro operatore (come scelto non interessa) per essere o meno approvato, mettendosi in attesa della risposta. Quando l'operatore contattato gli *comunica* se il pacco è approvato o no: se il pacco è stato approvato lo smista e lancia un evento in broadcast *smistato* (con payload il pacco), altrimenti lo scarta lanciando un evento in broadcast *scartato* (con payload il pacco); in entrambi i casi torna a riposo. Infine se quando è a riposo riceve un evento *verifica* (con payload un pacco) da un altro operatore, approva o meno il pacco (i dettagli non interessano) e *comunica* (con payload il pacco) l'esito della verifica all'operatore che gliel'ha richiesta.

Siamo interessati alla seguente attività principale che prende come parametro l'insieme di pacchi e di operatori e attiva concorrentemente le seguenti due sottoattività: (i) esercizio e (ii) analisi. La sottoattività di esercizio (i) manda in esecuzione tutti gli operatori passando loro i pacchi in modo che questi li smistino (i dettagli non interessano). La sottoattività di analisi (ii) calcola la percentuale di pacchi pesanti presenti nell'insieme e la manda in output. Una volta che tali sottoattività sono state completate, l'attività principale manda in output un messaggio di saluto e termina.

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 (inclusi vincoli non esprimibili in UML), diagramma stati e transizioni per la classe *Operatore*, diagramma delle attività, specifica del diagramma stati e transizioni, e specifica della attività principale e delle sottoattività NON atomiche (indicando in modo esplicito quali attività atomiche sono di I/O e quali sono Task), motivando, qualora ce ne fosse bisogno, le scelte effettuate.

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 solo le responsabilità sulle associazioni del diagramma delle classi.

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:

- La classe *Operatore* con classe *OperatoreFired*, e le classi per rappresentare tutte le *associazioni* che la legano con le altre classi.
- L'*attività principale* e le sue eventuali sottoattività NON atomiche.