

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 **9 luglio 2013**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di cene presso un ristorante con camerieri robotici. Una tavolata ha una breve descrizione (una stringa), un cliente responsabile, con nome e recapito telefonico, e un insieme non vuoto di tavoli, ciascuno con un numero di posti. Una tavolata ha inoltre un attributo booleano che indica se gli è stato portato il conto. Ad una tavolata è servita da esattamente un cameriere robotico, caratterizzato da un codice. Un cameriere robotico può servire al più 15 tavolate. Ad una tavolata sono poi associati i piatti ordinati, ciascuno con la quantità ordinata. I piatti sono caratterizzati da nome e costo unitario.

Il *cameriere* è inizialmente in uno stato di attesa. Quando è in attesa e riceve da una tavolata che serve una *chiamata* se pone nello stato di richiesta. Nello stato di richiesta il cameriere può ricevere due comandi, *ordine* e *conto*, dalla tavolata che ha inviato la chiamata. Il comando *ordine* include un insieme non vuoto di piatti ciascuno con la propria quantità e a fronte di tale comando il cameriere aggiorna opportunamente i piatti ordinati, con le quantità, associati alla tavolata. Invece a fronte del comando *conto* il cameriere porta il conto aggiornando l'attributo booleano corrispondente nella tavolata. In entrambi i casi il cameriere ritorna nello stato di attesa. Il funzionamento della tavolata, anche esso basato su eventi, non è di interesse.

Siamo interessati alla seguente attività principale che prende come parametro l'insieme di tutte le tavolate dell'applicazione. L'attività inizia attivando concorrentemente le seguenti due sottoattività: (i) esercizio, e (ii) analisi. La sottoattività di esercizio (i) manda in esecuzione tutte le tavolate e i camerieri. Poi si mette in attesa del comando di chiusura della serata da parte dell'utente che interrompe l'esercizio facendo richiedere il conto a tutte le tavolate che non l'hanno ancora fatto. La sottoattività di analisi (ii) calcola il numero medio di tavolate servite dai camerieri e lo stampa. Una volta che tali sottoattività sono state completate, calcola il conto di tutte le tavolate e poi li stampa.

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