

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 **12 gennaio 2016**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di "contatti" presso un call-center. Un contatto ha un codice (una stringa), un cliente coinvolto, ed un "esito" che può assumere solo 3 valori: "positivo", "negativo" e "indeterminato". Un cliente è caratterizzato da un nome ed un insieme di recapiti telefonici (stringhe). Un contatto è effettuato da esattamente un operatore che è caratterizzato da un codice. Ad un contatto sono poi associati un insieme non vuoto di servizi offerti, ciascuno con uno sconto specifico (valido solo per il contatto stesso) sul costo nominale del servizio. Un servizio è caratterizzato da un nome, una descrizione ed un costo nominale.

L'*operatore* è inizialmente in uno stato di riposo. Quando è a riposo e riceve un comando (evento) *contatta*, con parametro un oggetto contatto con esito "indeterminato", chiama il cliente proponendogli i servizi e gli sconti associati, e rimane in attesa. Nello stato di attesa l'operatore può ricevere tre comandi: (1) *annulla* che lo riporta a riposo registrando l'esito "negativo" per il contatto; (2) *accettato*, mandato dal cliente contattato, che lo riporta a riposo registrando in questo caso "positivo"; (3) *indeciso*, ancora mandato solo dal cliente contattato, che lo lascia in attesa lasciando l'esito del contatto "indeterminato". Il funzionamento del cliente, ecc., anche esso basato su eventi, non è di interesse.

Siamo interessati alla seguente attività principale che prende come parametro l'insieme di contatti e li partiziona in due insiemi, i contatti con esito indeterminato e gli altri. Dopodichè attiva concorrentemente le seguenti due sottoattività: (i) esercizio e (ii) analisi. La sottoattività di esercizio (i) manda in esecuzione tutti gli operatori dei contatti con esito indeterminato passando loro i contatti stessi (i dettagli non interessano). Poi si mette in attesa del segnale di I/O di chiusura che interrompe le attività gli tutti i contatti ancora aperti dando opportunamente il comando *annulla* a tutti gli operatori ancora in attesa. La sottoattività di analisi (ii) calcola il numero di contatti con esito positivo e il numero di contatti con esito negativo presenti nel secondo insieme. Una volta che tali sottoattività sono state completate, stampa 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 le *associazioni* che la legano con le altre classi.
- L'*attività principale* e le sue eventuali sottoattività NON atomiche.