

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 **20 febbraio 2020**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda una parte del sistema di gestione del traffico aereo. Degli aerei interessa il peso in kg e la portata del serbatoio in litri. Gli aerei possono essere solo aerei passeggeri o cargo. Un aereo ha due piloti assegnati e un responsabile delle comunicazioni. Di questi interessa rispettivamente il numero di ore di volo e le qualifiche (una stringa). Ogni pilota è sempre assegnato a un aereo. Nel sistema sono inserite anche altre persone che non sono di nessuna di queste due categorie. Di tutti interessa la città in cui abitano. Un aereo può trovarsi in un aeroporto, oppure in nessuno (se è in volo). Di ogni aeroporto interessa conoscere la città in cui si trova e il codice IATA (una stringa).

In questo compito siamo interessati al comportamento degli aerei. Un aereo inizialmente si trova in un hangar. Quando riceve l'evento PARTENZA si sposta sulla pista e attende l'evento DECOLLO da parte della torre di controllo. Con l'evento PARTENZA viene anche comunicato l'aeroporto di destinazione. Quando l'aereo riceve l'evento DECOLLO lascia l'aeroporto in cui si trova, e passa allo stato di volo. Se non ci sono problemi durante il volo, l'aereo ATTERRA alla sua destinazione prevista; questo viene comunicato all'aereo con un apposito evento (senza payload). Situazioni di emergenza che rendono impossibile l'arrivo alla destinazione prevista (es. emergenza medica o problema tecnico) sono segnalate all'aereo con l'evento EMERGENZA. In questo caso, viene selezionato l'aeroporto più vicino (assumere che esista un metodo ausiliario predefinito `AeroportoVicino()`), a cui viene chiesta una pista per un atterraggio di emergenza. L'evento ACCETTATO indica che l'aereo può atterrare dove richiesto, per cui vola verso l'aeroporto scelto. Invece, in caso di impossibilità di atterraggio nell'aeroporto scelto all'aereo arriva un evento RIFIUTATO, e l'aereo cerca un altro aeroporto usando lo stesso metodo `AeroportoVicino()`.

Siamo interessati alla seguente attività principale. L'attività prende come parametro di input un aeroporto A e un insieme di aerei I , e controllo che siano tutti presenti nell'aeroporto stesso. In caso negativo, termina con un messaggio di errore. In caso positivo, esegue concorrentemente le seguenti due sottoattività: *Vola* e *Monitora*. La sottoattività *Vola* avvia i voli degli aerei I (i dettagli non interessano); poi si mette in attesa del comando di fine voli e termina. La sottoattività *Monitora* richiede periodicamente i dati di tutti gli aerei e produce un report, che è una stringa (i dettagli non interessano). Una volta che tali sottoattività sono state completate, viene stampato il report e l'attività principale termina.

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare l'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 *Aereo*, diagramma delle attività, specifica del diagramma stati e transizioni, specifica delle signature di tutte attività e segnali di I/O, motivando, qualora ce ne fosse bisogno, le scelte effettuate.

Domanda 2. Effettuare il 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 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 *Aereo*, la classe *AereoFired*, le eventuali superclassi e sottoclassi, e le classi JAVA per rappresentare la *associazione* fra *Aereo* e *Pilota*.
- l'*attività principale* e la sottoattività *Monitora*; non realizzare le altre attività, atomiche o meno.