

requisiti che implicano responsabilità

Specifica

Dati

L'applicazione da progettare riguarda la gestione da parte di una compagnia di trasporto locale di una flotta di autobus. Di ogni **autobus** sono di interesse la **targa** (una stringa), il **modello** (una stringa) e un **timestamp** indicante l'istante da cui è prevista la prossima manutenzione. **Un autobus si può trovare in una autorimessa**. Di ogni **autorimessa** si vogliono conoscere l'**indirizzo** (una stringa), il numero di **posti** totali (un intero) e i timestamp di prossima manutenzione degli autobus presenti. Alcune autorimesse sono anche **officine**. Di queste ultime è di interesse conoscere il numero di **capannoni** (un intero). La compagnia di trasporto vuole anche tenere traccia dei dipendenti. Di ogni **dipendente** interessa il **nome** (una stringa). Esistono diverse categorie disgiunte di dipendenti. In particolare, degli **autisti** interessa conoscere il numero di **rottture** che si sono verificate mentre guidavano, mentre dei **meccanici** interessa l'**officina** in cui lavorano (**esattamente una**). Ogni **autista** è autorizzato ad entrare in almeno una autorimessa.

Comportamento

Operazioni che implicano responsabilità

Siamo interessati al comportamento degli autobus. Inizialmente, un autobus si trova in una autorimessa (che deve essere nota al momento della nascita) nello stato inautorimessa. Nello stato inautorimessa, l'autobus può ricevere, da parte di un autista, un messaggio di uscita. **Se l'autista è autorizzato all'accesso all'autorimessa, l'autobus va in stato inservizio e viene aggiornato lo stato dell'associazione con l'autorimessa**. Nello stato inservizio, l'autobus può ricevere un messaggio di entrata da parte di un autista (che deve essere lo stesso che ha inviato uscita) che ha come parametro una autorimessa e che, se l'autista è autorizzato per l'autorimessa porta l'autobus nello stato inautorimessa, aggiorna l'associazione con autorimessa e invia un messaggio di entrata sia all'autista che all'autorimessa. Nello stato inservizio l'autobus può ricevere un messaggio di rottura, che aggiorna il numero di rotture dell'autista e porta l'autobus in stato rottura. Dallo stato rottura, un messaggio trasporto, con parametro una officina, aggiorna l'associazione con autorimessa e porta l'autobus in stato inmanutenzione. Dallo stato inmanutenzione, un messaggio fine porta l'autobus in stato inautorimessa aumentando il timestamp di prossima manutenzione di tre mesi nel futuro (otto milioni di secondi). Dallo stato inautorimessa, se l'autorimessa è anche una officina, l'autobus va in stato inmanutenzione alla ricezione di un messaggio di periodica.

Attività

Siamo interessati all'attività principale di gestione della manutenzione. L'attività viene inizializzata con un insieme di autobus e un insieme di officine. All'inizio, l'attività chiede all'utente di inserire due parametri t1 e t2 che vengono usati dopo. L'attività avvia a questo punto tre sottoattività complesse in parallelo. Nella prima viene ripetuta ogni t1 secondi una attività atomica, di cui non interessano i dettagli, che verifica quali **autobus attualmente in autorimessa hanno necessità di manutenzione**. Nella seconda sottoattività viene ripetuta ogni t2 secondi una attività atomica, di cui non interessano i dettagli, che controlla lo stato degli autobus e in caso di rottura assegna gli autobus a una officina casuale per la manutenzione. La terza sottoattività esegue una sottoattività di attesa di un input dall'utente per chiudere l'attività principale. Alla ricezione del comando dell'utente, la terza sottoattività interrompe le altre due.

Domande

Domanda 1

Basandosi sui requisiti riportati sopra, effettuare l'analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo di: diagramma delle classi (inclusi eventuali vincoli non esprimibili in UML); diagramma stati e transizioni per la classe Autobus; diagramma delle attività; specifica del diagramma stati e transizioni; segnatura dell'attività principale, sottoattività non atomiche, atomiche e segnali di

input/output. Si noti che NON è richiesta la specifica delle attività. Motivare, qualora ce ne fosse bisogno, le scelte di progetto.

Domanda 2

Effettuare il progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare definire SOLO le responsabilità sulle associazioni del diagramma delle classi. Per le responsabilità utilizzare la seguente notazione: M per molteplicità, O per operazioni, e R per requisiti.

Domanda 3

Effettuare la realizzazione, producendo un programma Java e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare, realizzare in Java SOLO i seguenti aspetti dello schema concettuale:

- Per l'esame si supponga che i timestamp siano memorizzati e passati come parametro nella forma di interi lunghi così come restituiti da `Java.lang.System.currentTimeMillis()`.
- Si implementi questo meccanismo di interruzione con una modalità a piacimento.
 - La classe *Autobus* con la classe *AutobusFired*, le classi Java per rappresentare l'associazione a responsabilità doppia tra *Autobus* e *Autorimessa*.
 - L'attività principale e le due sottoattività complesse *verifica* e *attesa*. Si richiede inoltre l'implementazione della attività atomica relativa all'attività complessa *verifica*.

Diagrammi

diagramma dei dati (classi)

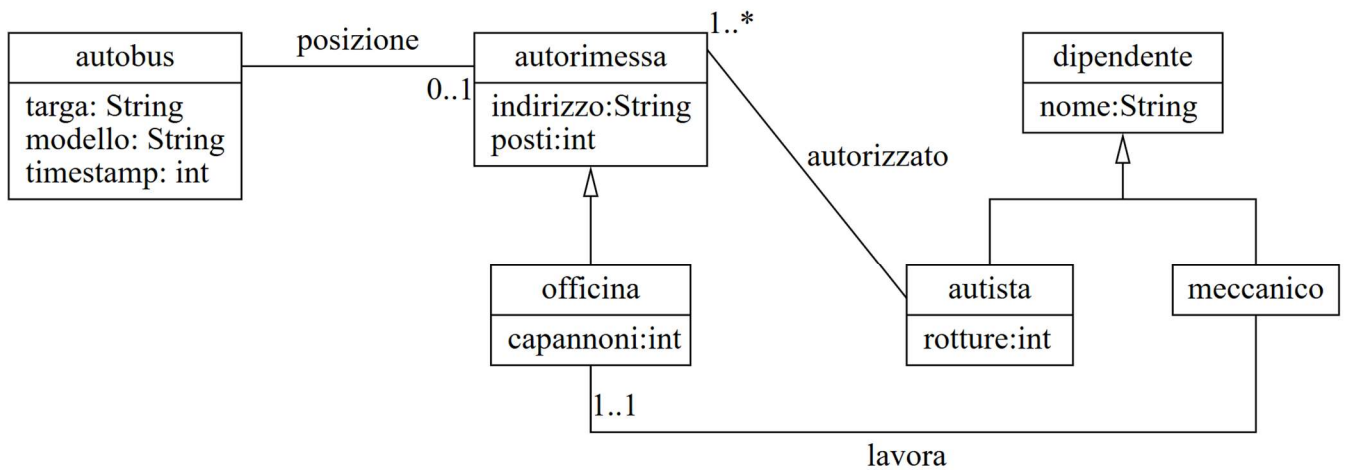


Diagramma del comportamento (stati e transizioni)

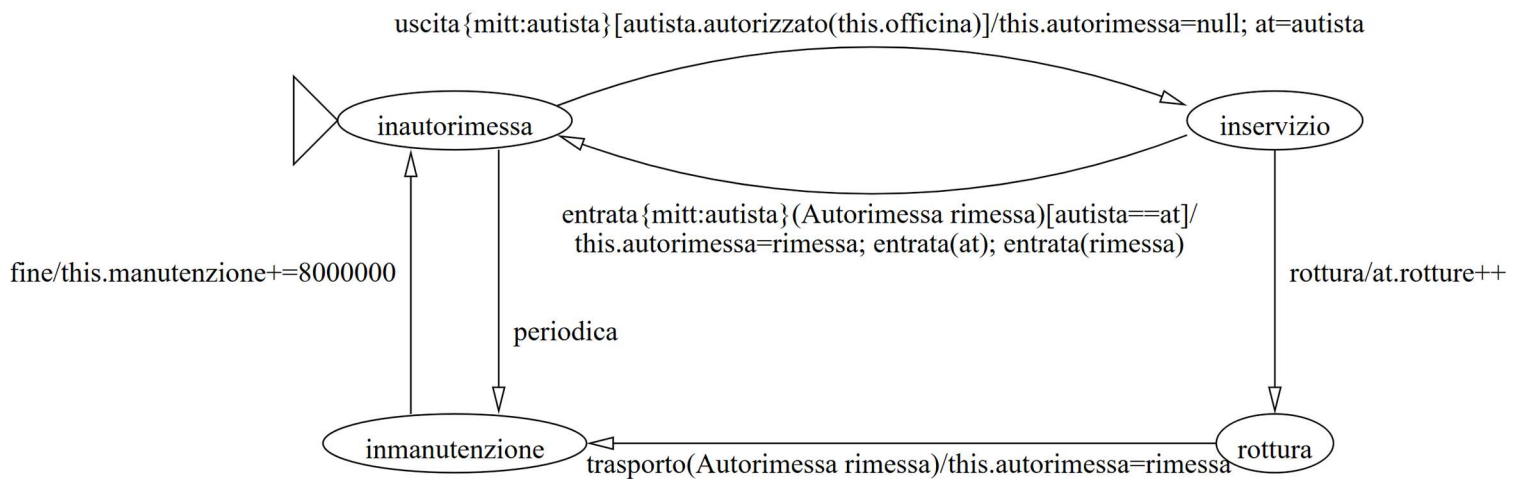


Diagramma delle attività

