SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica
Corso di Progettazione del Software
Esame del **20 luglio 2022**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione dei vari sistemi informatici (embedded) presenti all'interno di una automobile moderna (ad esempio l'ABS). Di ogni sistema informatico interessa il nome, i sensori a cui è collegato (in numero qualsiasi) e gli attuatori a cui è collegato (in numero qualsiasi). Esistono soltanto due tipi di sistema informatico: i sistemi di sicurezza ed i sistemi di informazione/intrattenimento (detti sistemi di infotainment). Dei sistemi di sicurezza interessano la data di ultima manutenzione ed il nome dell'ultimo meccanico che ha revisionato il sistema. Dei sistemi di infotainment interessa la versione del software. Ogni sensore è collegato ad almeno un sistema informatico ed ogni attuatore è collegato ad almeno un sistema informatico. Poichè diversi sistemi informatici possono cercare di usare contemporaneamente gli attuatori, un valore di priorità (un intero) è necessario per indicare se un sistema informatico ha più diritto di un altro su uno specifico attuatore. E' inoltre di interesse memorizzare all'interno della scatola nera dell'automobile gli eventi che interessano l'automobile. Ogni evento ha un timestamp (modellato come un long) ed una descrizione. Degli eventi provenienti da sensore interessa il sensore che lo ha generato, mentre degli eventi provenienti dagli attuatori interessa memorizzare l'attuatore che lo ha generato.

Siamo interessati al comportamento dei sistemi informatici. Un sistema informatico è inizialmente in stato *spento*. Ad un certo punto, quando l'automobile viene accesa, esso passa in stato *acceso* se tutti i sensori e gli attuatori collegati sono in stato *ready* (si supponga di potere accedere allo stato di sensori ed attuatori), altrimenti va in stato *errore*. L'evento di accensione fornisce come argomento un insieme di sistemi informatici di infotainment; non appena il sistema informatico va in stato di errore (qualunque sia il motivo) viene visualizzata una notifica sui sistemi di infotainemnt indicati dall'evento di accensione (si supponga che i sistemi di infotainment forniscano un metodo apposito). In particolare, un sistema informatico può finire in stato *errore* anche a partire dallo stato acceso, se il sistema va in errore durante dopo l'accensione. Al momento dello spegnimento dell'automobile, il sistema informatico torna in stato *spento*.

Siamo interessati all'attività principale di marcia dell'automobile. L'attività principale prende come input la lista di sistemi informatici da avviare. All'inizio dell'attività principale, l'attività atomica di avvio automobile invia l'evento di avvio a tutti i sistemi informatici passati come input. A questo punto, l'attività principale controlla lo stato di ogni sistema informatico. Per ogni sistema informatico in stato di *errore*, l'attività chiede all'automobilista se continuare con l'accensione o terminare l'attività. Una volta verificati tutti i sistemi informatici, ricevendo conferma a proseguire per ognuno di quelli in stato di errore, si avviano due attività complesse concorrenti. La prima sottoattività complessa contiene una attività atomica di raccolta degli eventi nella scatola nera (i dettagli non sono di interesse) che termina alla ricezione dell'evento di spegnimento dell'automobile. L'altra sottoattività complessa contiene una attività atomica che controlla continuamente lo stato dei sistemi informatici e termina se si trova un sistema informatico in errore oppure dopo avere ricevuto un evento di spegnimento dell'automobile (i dettagli dell'attività atomica non sono di interesse). Una volta concluse le attività in parallelo, il contenuto della scatola nera viene inviato nel cloud (attività atomica di cui non interessano i dettagli).

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 *SistemaInformatico*; 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, ed 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:

- La classe *SistemaInformatico* (ed eventuali sottoclassi) con la classe *SistemaInformaticoFired*, le classi Java per rappresentare le associazioni di cui la classe *SistemaInformatico* ha responsabilità.
- L'attività principale e le sue eventuali sottoattività NON atomiche.