

Corso di Progettazione del SoftwareEsame del **27 Luglio 2018***Tempo a disposizione: 3 ore*

Requisiti. L'applicazione da progettare riguarda un sistema di diffusori audio e video per domotica. Una installazione domotica ha un indirizzo (una stringa) e un insieme di reti. Le reti hanno un nome e appartengono a una sola installazione. Inoltre le reti sono costituite da uno o più diffusori e tra questi esattamente uno svolge il ruolo di master. Ciascun diffusore appartiene esclusivamente ad una rete. I diffusori hanno un nome e sono partizionati in diffusori audio e diffusori video. Dei diffusori audio interessano i watt nominali (un intero) e dei diffusori video interessano i pollici (un reale).

Siamo interessati al comportamento dei diffusori. Un diffusore è inizialmente in *standby*. Se, nello stato di *standby*, riceve il comando di *diffondere* con payload un contenuto di tipo audio o video, verifica se egli stesso lo può diffondere, cioè se esso stesso è del tipo (audio o video secondo quanto richiesto nel payload) giusto, e se lo è lo diffonde mettendosi in *diffusione*. Altrimenti sceglie uno dei diffusori della sua rete del tipo giusto (si assume di avere una ausiliaria funzione di scelta già definita) e gli passa il comando di diffondere con il rispettivo payload, mettendosi nello stato di attesa. Se arriva il comando di *stop*, se è nello stato di *diffusione* smette di diffondere il contenuto e torna in *standby*; se invece è nello stato di *attesa*, passa il comando *stop* al diffusore scelto e torna in *standby*.

Siamo interessati alla seguente attività principale. L'attività prende in input una installazione *I* e verifica che ogni rete abbia un congruo numero di diffusori audio e video (i dettagli non interessano). Se la verifica non va a buon fine, l'attività termina segnalando in output un errore. Altrimenti concorrentemente esegue le seguenti due sottoattività: (i) funzionamento e (ii) analisi. La sottoattività di funzionamento (i) avvia l'installazione portando tutti i diffusori di tutte le reti dell'installazione *I* nello stato di *standby*. Poi si mette in attesa del segnale di input di fine funzionamento che interrompe il funzionamento stesso. La sottoattività di analisi (ii) calcola il numero di diffusori audio e video di ciascuna rete dell'installazione *I* mandando un report con questi dati in output. Una volta che tali sottoattività sono state completate, l'attività principale manda un segnale di output con il numero medio di diffusori audio e video per rete dell'installazione e 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 *Diffusore*, diagramma delle attività, specifica del diagramma stati e transizioni, e specifica dell'attività principale e delle sottoattività NON atomiche (cioè NON va riportata la specifica del Task e dei segnali di I/O), motivando, 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. 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 di progetto. È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- La classe **Diffusore** con classe **DiffusoreFired**, le eventuali sottoclassi, e le classi JAVA per rappresentare le *associazioni* di cui la classe **Diffusore** ha responsabilità. Qualora sia necessario realizzare più classi **TipoLink** e **Manager** è obbligatorio riportarne solo una di entrambe.
- L'*attività principale* e le sue eventuali sottoattività NON atomiche.