

Corso di Progettazione del SoftwareEsame del **19 Luglio 2011***Tempo a disposizione: 3 ore*

Requisiti. L'applicazione da progettare riguarda un sistema domotico (automazione della propria abitazione e/o ufficio). Un dispositivo è caratterizzato da un nome, una locazione (una stringa) ed un insieme ordinato non vuoto di operazioni che esso offre. Un'operazione è caratterizzata da un nome, da un input opzionale e da esattamente un output. Un input è caratterizzato da un nome, dal suo tipo (una stringa), dal valore che assume (una stringa) e dal valore di default che assume in eventuali operazioni di verifica (una stringa). Un output è caratterizzato da un nome, dal suo tipo (una stringa), dal valore che assume (una stringa) e dal valore corretto che dovrebbe assumere se l'operazione funzionasse correttamente (una stringa). Si noti che la stessa operazione può essere offerta da vari dispositivi ed è di interesse data una operazione conoscere quali dispositivi la offrono. Ad esempio, il dispositivo *lampadina 27*, ubicato nella *stanza da letto*, offre due operazioni: (i) *accendi*, senza input e con output *esito* di tipo *booleano*, che ha valore *non specificato* in questo momento e che assume come valore corretto *true*, e (ii) *spegni*, senza input e con output *esito* di tipo *booleano*, che ha valore *false* in questo momento e che dovrebbe assumere come valore corretto *true*.

Siamo interessati a progettare l'attività di test & reset dell'ambiente domotico (casa/ufficio), che prende come parametro un insieme di dispositivi. L'attività inizia stampando la cardinalità di tale insieme di dispositivi, poi procede eseguendo concorrentemente la seguente sottoattività complessa per ognuno dei dispositivi dell'insieme: per ogni operazione del dispositivo, (i) prende il valore di default dell'input e lo passa come parametro ad una attività atomica di I/O *simulazione di chiamata* (se l'operazione non ha input passa la stringa vuota). Questa attività atomica di I/O, il cui funzionamento interno esula dagli scopi del compito d'esame, restituisce una stringa in output; (ii) si esegue il task di *verifica*, in cui si controlla che la stringa di output restituita corrisponda effettivamente al valore corretto che l'output dell'operazione dovrebbe assumere. Se la verifica non va a buon fine, (iii) si *visualizza* il nome dell'operazione che non si comporta correttamente. Una volta che tutte le sottoattività concorrenti sono completate, si visualizza un messaggio di saluto 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, diagramma delle attività, specifica delle attività complesse, 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 *operazione* e le eventuali classi per rappresentare e gestire le associazioni che la legano alla classe *Dispositivo*.
- L'attività principale con tutte le eventuali sottoattività complesse, e l'attività atomica di *verifica* escluse le attività atomiche.