

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 **15 settembre 2022**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di impianti d'allarme. Un impianto d'allarme è caratterizzato da un codice di sblocco e da un insieme di numeri da chiamare in caso di allarme. Ogni impianto di allarme contiene almeno un sensore. Di ogni sensore interessa conoscere il numero di matricola, la marca, il modello, una descrizione e l'impianto di allarme su cui è installato. Esistono soltanto due categorie di sensori, i sensori digitali ed i sensori meccanici. Dei sensori digitali interessa il numero di tecnologie diverse impiegate. Dei sensori meccanici interessa il tempo di vita utile (in anni) previsto. L'applicazione deve inoltre memorizzare gli eventi di allarme che si sono verificati. Di ogni evento di allarme interessano il timestamp (un intero lungo), l'impianto di allarme che lo ha generato e l'insieme dei sensori (almeno uno) coinvolti, mantenendo l'ordine in cui sono scattati. Si noti che tutti i sensori collegati ad un evento di allarme devono essere installati sullo stesso impianto di allarme a cui fa capo l'evento di allarme. Alcuni eventi di allarme sono falsi allarmi. Per questi ultimi interessa memorizzare il nome della persona che ha annullato l'allarme. Infine, di ogni impianto di allarme interessano gli eventuali eventi di allarme generati.

Siamo interessati al comportamento dei sensori. Un sensore è inizialmente nello stato *non configurato*. Da questo stato può ricevere un evento di configurazione contenente come parametri due pattern di luci, ognuno rappresentato per semplicità da una stringa. Questi due pattern rappresentano le luci da accendere in caso di chiusura e in caso di apertura. Alla ricezione di questo evento, il sensore passa allo stato *aperto* o *chiuso* a seconda dello stato dei contatti del sensore, e visualizza il pattern di luci corrispondente. Nello stato *chiuso*, il sensore può ricevere un evento di apertura, in corrispondenza del quale passa nello stato *aperto* mostrando il pattern di luci di apertura. Nello stato *aperto* può ricevere un evento di chiusura che lo riporta nello stato *chiuso*, mostrando il pattern di luci di chiusura. Si supponga che lo stato dei contatti di un sensore possa essere verificato chiamando un metodo `String statoContatti()` e che l'accensione di luci sia possibile tramite un metodo `void accendiLuci(String pattern)` entrambi disponibili sulla classe *SensoreAllarme* e di cui non interessa il corpo.

Siamo interessati all'attività principale di funzionamento di un impianto di allarme. Questa attività prende come parametro il riferimento ad un impianto di allarme. Inizialmente l'attività richiede all'utente l'inserimento del codice di sblocco. Se questo è corretto si prosegue con l'attività principale, altrimenti si termina visualizzando un messaggio di errore. Se il codice di sblocco è corretto, a tutti i sensori è inviato un evento di configurazione (attività atomica). A questo punto, due sottoattività complesse vengono eseguite in parallelo. La prima sottoattività controlla continuamente lo stato dei sensori. Se almeno un sensore è in stato *aperto*, viene generato un evento di allarme riportante l'insieme dei sensori aperti e termina. Se nessun sensore è aperto si ritorna a controllare lo stato dei sensori. La seconda sottoattività complessa resta in attesa finché l'utente inserisce il codice di sblocco corretto. Quando entrambe le sottoattività sono concluse, l'attività principale termina.

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