

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica ed in Ingegneria dei Sistemi Informatici
Corso di Progettazione del Software
Esame del **7 giugno 2013**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di un videogioco di combattimenti robotici. Una *Arena* ha un nome e contiene un insieme non vuoto di *Ring*. Ogni *Ring*, che ha a sua volta un nome, appartiene ad un'unica *Arena*. Su un *Ring* vengono posti almeno due *Robot*. I *Robot* hanno un nome, un intero compreso tra 0 e 99 che indica la forza nominale del *Robot* stesso, ed un intero che indica il numero di sconfitte. I *Robot* sono posti in al più un *Ring*. Il *Ring* è controllato da un *Master*, anch'esso con un nome, che si occuperà di mandare opportuni segnali ai *Robot* del *Ring* stesso. Ciascun *Master* controlla al più un *Ring*.

Il *Robot* è inizialmente in uno stato di attesa. Quando riceve dal *Master* del suo *Ring* il comando di combattere, si mette nello stato di combattimento e attacca uno dei *Robot* del suo *Ring* scelto casualmente (si assuma la funzione di scelta già realizzata). Se un *Robot* nello stato di combattimento subisce un attacco, verifica se la forza nominale del *Robot* attaccante è maggiore o uguale alla sua. Se lo è, aggiorna il numero di sconfitte, e si mette in uno stato di fuga; altrimenti, attacca a sua volta l'attaccante, rimanendo nello stato di combattimento. Quando il *Robot* è nello stato di fuga può ricevere il comando di combattere dal *Master* tornando nello stato di combattimento. Inoltre, sia quando è nello stato di combattimento che nello stato di fuga, il *Robot* può ricevere dal *Master* il comando di mettersi in attesa. Ad eccezione del numero di sconfitte, il *Robot* può essere modificato solo quando è nello stato di attesa. Il funzionamento del *Master* non ci riguarda.

Siamo interessati alla seguente attività che prende come parametro un *Ring* R . Inizialmente azzera il numero di sconfitte di tutti i *Robot* di R . Poi l'attività procede concorrentemente con le seguenti due sottoattività: (i) gioco, e (ii) analisi. La sottoattività di gioco (i) inizia con l'invio, da parte del *Master* di R , del comando di combattere a tutti i *Robot* di R (si assuma che l'environment sia già stato inizializzato all'esterno dell'attività principale). Poi si mette in attesa del comando da parte dell'utente che interrompe il gioco, riportando tutti i *Robot* di R nello stato di attesa attraverso opportuni comandi del *Master*. La sottoattività di analisi (ii) calcola il numero di tutti i *Robot* presenti nei *Ring* dell'*Arena* di R e lo stampa. Una volta che tali sottoattività sono state completate, calcola e stampa un resoconto che indica il numero di sconfitte subite da ciascun *Robot* di R .

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare la fase di 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 *Robot*, diagramma delle attività, specifica del diagramma stati e transizioni, e specifica della attività principale e delle sottoattività NON atomiche (indicando in modo esplicito quali attività atomiche sono di I/O e quali sono Task), motivando, qualora ce ne fosse bisogno, le scelte effettuate.

Domanda 2. Effettuare la fase di 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 fase di 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 *Robot* con classe *RobotFired*, e le classi per rappresentare le *associazioni* che legano la classe *Ring*.
- L'*attività principale* e le sue eventuali sottoattività non atomiche.