

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 **17 Giugno 2016**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda una parte di un videogioco su supereroi per tablet e smartphone. Un gioco ha un nome (una stringa) e vi partecipano diversi giocatori, ciascuno dei quali partecipa esclusivamente al gioco stesso. I giocatori, che hanno un nome (una stringa) sono divisi in 2 categorie: i supereroi di cui interessa la forza (un intero), e i villan di cui interessa la cattiveria (un reale compreso tra 0 e 1). Ogni supereroe ha diversi antagonisti tra i villan, e tra questi ce ne esattamente uno che è la sua nemesi.

In questo compito siamo interessati al comportamento dei soli supereroi. Un supereroe è inizialmente in *vigile attesa*. Quando riceve un *allarme* (non importa da chi) relativo ad un villan suo antagonista, parte alla *ricerca* del villan. Quando il villan si *rivela* allora, se la sua cattiveria è minore o uguale a 0,5, cerca di convincerlo a redimersi (manda un evento *redimiti* al villan), mettendosi in *attesa della remissione* da parte del villan. Se invece la cattiveria del villan è maggiore di 0,5 allora lo *attacca* direttamente mettendosi nello stato di *combattimento*. Se, quando in *attesa della remissione*, riceve dal villan un esito *positivo*, allora lo leva dai suoi nemici, eccetto nel caso il villan sia la sua nemesi, e torna in *vigile attesa*. Se l'esito è *negativo*, allora lo *attacca* mettendosi nello stato di *combattimento*. Quando è nello stato di *combattimento*, se il villan *scappa* allora torna in *vigile attesa*; se invece il villan *controattacca* lo *attacca* nuovamente, rimanendo in *combattimento*. (Come il villan sceglie tra esito positivo e negativo o tra scappa e controattacca non interessa).

Siamo interessati alla seguente attività principale. L'attività prende come parametro di input un gioco G e concorrentemente esegue le seguenti due sottoattività: *Gioca*, e *Analizza*. La sottoattività *Gioca* avvia il gioco attivando tutti i giocatori di G mandando opportuni eventi (i dettagli non interessano). Poi si mette in attesa del comando di fine-gioco da parte dell'utente, che interrompe il gioco. La sottoattività *Analizza* verifica se per tutti i supereroi del gioco G anche la loro nemesi appartiene a G , segnalandolo in output. Una volta che tali sottoattività sono state completate, si calcola il numero medio di villan per supereroe del gioco segnalandolo in output, terminando così l'attività principale.

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 *Supereroe*, diagramma delle attività, specifica del diagramma stati e transizioni, specifica dell'attività principale e delle sottoattività NON atomiche, 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 *Giocatore*, la sottoclasse *Supereroe* con classe *SupereroeFired*, le classi JAVA per rappresentare le *associazioni* di cui queste classi hanno responsabilità.
- L'*attività principale* (NON le sue sottoattività *Gioca* e *Analizza*, né le sue sottoattività atomiche).