

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea in Ingegneria Informatica ed Automatica
Corso di Progettazione del Software
Esame del **21 gennaio 2019**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda una parte di un videogioco di fantascienza. Un gioco è costituito da diversi giocatori. Ciascun giocatore appartiene ad un solo gioco e ha un nome (una stringa). I giocatori sono divisi in 3 categorie: i droidi, i robot, gli umani. Dei droidi interessa il modello (una stringa), dei robot l'anno di produzione (un intero), e degli umani una descrizione testuale (una stringa). Un droide ed un robot formano un equipaggio di cui interessa un codice identificativo. Ogni droide fa parte di diversi equipaggi. invece ogni robot fa parte esattamente di un equipaggio. Un equipaggio pilota esattamente una astronave ed una astronave ha un solo equipaggio. Delle astronavi interessa una descrizione testuale (una stringa). Gli umani possono essere capitani di una astronave. Ogni astronave può avere un numero arbitrario di capitani.

Siamo interessati al comportamento dei droidi. Un droide è inizialmente alla *base*. Se riceve il comando di *volare* con payload un umano ed una astronave del cui equipaggio lui è il droide, chiede all'umano di salire sull'astronave come capitano, ponendosi in *attesa del capitano*. Quando l'umano gli comunica *capitano-salito* si mette *in volo*. Cosa fa quando in volo non interessa, eccetto che quando riceve il comando di *rientro* torna alla *base*. Il comportamento degli altri tipi di giocatore non interessa.

Siamo interessati alla seguente attività principale. L'attività prende in input un gioco G e verifica che il numero di droidi, robot, e umani di G sia congruo (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)* gioco e *(ii)* analisi. La sottoattività di gioco *(i)* avvia il gioco attivando tutti i giocatori di G mandando opportuni eventi (i dettagli non interessano). Poi si mette in attesa del segnale di fine-gioco che interrompe il gioco stesso. La sottoattività di analisi *(ii)* calcola un report sul gioco G (i dettagli non interessano). Una volta che tali sottoattività sono state completate, l'attività principale manda un segnale di output con il report calcolato nella sottoattività di analisi e 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 vincoli non esprimibili in UML); diagramma stati e transizioni per la classe *Droide*; 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.

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 *Droide* con classe *DroideFired*, le eventuali superclassi, e le classi JAVA per rappresentare le *associazioni* di cui la classe *Droide* e le sue superclassi hanno responsabilità.
- L'*attività principale* e le sue eventuali sottoattività NON atomiche.