

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 **22 Settembre 2017**  
*Tempo a disposizione: 3 ore*

**Requisiti.** L'applicazione da progettare riguarda un sistema robotico per l'esplorazione automatizzata di Marte. Un' esplorazione ha un nome (una stringa) e un insieme di basi. Delle basi interessano le coordinate geografiche (due reali) e una descrizione (una stringa). Una base ha un insieme di Transporter, ciascuno con un nome e con le coordinate geografiche in cui è situato. Ogni Transporter appartiene esattamente a una base. Alcuni Transporter sono HeavyTransporter. Degli HeavyTransporter interessa il carico trasportabile in kg (un reale) e raggio di azione dalla base (un reale). Ogni HeavyTransporter ha a bordo uno o più Rover. Ogni Rover è a bordo di esattamente un HeavyTransporter. Dei Rover interessa il raggio di azione dal HeavyTransporter su cui sono a bordo.

In questo compito siamo interessati al comportamento degli HeavyTransporter. Un HeavyTransporter è inizialmente alla base. Quando riceve un comando *esplorazione* dalla propria base con payload le coordinate da raggiungere: (1) se la distanza è inferiore al suo raggio di azione, si reca presso le coordinate desiderate; (2) se la distanza è inferiore alla somma della suo raggio di azione e del raggio di azione dal suo Rover che può andare più lontano, allora si reca nella coordinate a distanza massima dalla base lungo la retta che passa per le coordinate desiderate e per le coordinate della base e fa partire uno dei Rover il cui raggio di azione è superiore alla distanza dal punto corrente alle coordinate desiderate;<sup>1</sup> (3) se la distanza è superiore, ignora il comando. Quando riceve il comando *rientrare alla base*: nel caso (1), torna alla base; nel caso (2), manda un comando di *ritornare* al Rover scelto restando in attesa del segnale *rientrato* dal Rover, e quando riceve tale segnale torna alla base; nel caso (3), ignora il comando.

Siamo interessati alla seguente attività principale. L'attività prende come parametro di input una esplorazione  $E$  e concorrentemente esegue le seguenti due sottoattività: *Esplora* e *Verifica*. La sottoattività *Esplora* avvia l'esplorazione attivando tutte le basi associate ad  $E$  e tutti i Transporter in esse, mandando opportuni eventi (i dettagli non interessano); poi si mette in attesa del comando di fine esplorazione da parte dell'utente, che interrompe l'esplorazione. La sottoattività *Verifica* calcola per ciascuna base dell'esplorazione  $E$  quanti HeavyTransporter sono disponibili, segnalandolo in output. Una volta che tali sottoattività sono state completate, si segnala in output la fine dell'esplorazione, 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 *HeavyTransporter*, 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 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 *HeavyTransporter*, con classe *HeavyTransporterFired*, le eventuali superclassi, e le classi JAVA per rappresentare le *associazioni* su cui *HeavyTransporter* ha responsabilità.
- L'*attività principale*, le sottoattività *Esplora* e *Verifica*, ma NON le sottoattività atomiche.

---

<sup>1</sup>Le funzioni ausiliarie per gestire coordinate, raggi di azione, distanze, per la verifica del raggio di azione massimo raggiungibile dai Rover a bordo e per la selezione dei Rover vanno dichiarate ma non definite (cioè va scritta la segnatura ma non il corpo).