

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 **19 luglio 2012**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di un grafo colorato attivo e l'esecuzione delle sue componenti connesse. Una componente connessa è costituita da un nome (una stringa) e da un nodo iniziale. Ogni nodo ha un codice (una stringa) ed è collegato a un insieme di nodi successivi. Tali collegamenti sono gli archi orientati del grafo. I nodi sono partizionati in due soli tipi: *verdi* e *rossi*. Per i primi interessa la gradazione (una stringa), mentre per i secondi interessa l'intensità (un intero). Ogni nodo rosso è collegato ad esattamente un nodo successore speciale che deve essere verde.

Un nodo è inizialmente spento. Quando un nodo spento riceve l'evento *toggle*, si accende e manda a sua volta un evento *toggle* ad una selezione di nodi successivi. Per scegliere a quali nodi mandare l'evento *toggle*, conta i nodi verdi di cui è successore: se sono la maggioranza lo manda a tutti i suoi successori verdi; altrimenti lo manda a tutti i suoi successori rossi. Quando un nodo acceso riceve l'evento *toggle* si spegne.

Siamo interessati alla seguente attività che prende come parametro una componente connessa, calcola l'insieme dei codici del nodo iniziale e dei suoi successori (diretti) e lo stampa. L'attività poi procede concorrentemente con le seguenti due sottoattività, una di gioco (*i*) e una di verifica (*ii*). La sotto attività di gioco (*i*) inizia il gioco, attraverso l'inizializzazione dell'environment e l'invio al nodo iniziale della componente connessa dell'evento *toggle* e poi si mette in attesa del comando di fine esecuzione da parte dell'utente (attraverso un'opportuna attività di input/output), che interrompe l'esecuzione del gioco. La sottoattività di verifica (*ii*) calcola il numero di nodi rossi e di nodi verdi che formano la componente connessa visitandola a partire dal nodo iniziale. Una volta che tali sottoattività sono completate, stampa un messaggio di saluto riportando il numero di nodi rossi e di nodi verdi calcolato dalla precedente verifica e termina.

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 *Nodo*, 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 *Nodo* (con classe *NodoFired*), le sue *sottoclassi*, e le classi per rappresentare le *associazioni* che legano la classe *Nodo* a se stessa.
- L'*attività principale*.