

Un centro di ricerca sta sviluppando un software di simulazione a supporto di un laboratorio di virologia, con lo scopo di studiare gli effetti delle regole per il distanziamento sociale. Nello specifico si vuole studiare quanti contatti avvengono tra persone che seguono le regole (persone definite *attente*) rispetto alle persone che non lo fanno (definite *distratte*).

Allo stato corrente, il codice prevede la presenza in un solo tipo di persone, senza alcun particolare comportamento: si muovono confinate all'interno di un ambiente (modellato dalla classe `dist.sim.Ambiente`) e si spostano occupando delle celle scelte casualmente, anche coincidenti, disposte su posizioni (modellate dalla classe `dist.sim.Coordinate`) di un piano cartesiano.

DOMANDA 1 (5%)

Modificare il codice della classe `dist.sim.Coordinate` affinché i test presenti nelle classi `CoordinateTest` comincino ad avere successo. Già dopo aver effettuato questa correzione, è possibile verificare il corretto funzionamento dell'intera simulazione eseguendo il metodo `main()` della classe `dist.Main`.

*(N.B. Per una più agevole comprensione della descrizione che segue, si consiglia di provare ad eseguire il metodo `main()` della classe `dist.sim.Main`, osservare l'animazione della simulazione già dopo aver risposto a questa prima domanda, e premere il tasto **ESCape** dopo qualche secondo per fermare la simulazione. La simulazione stampa, a fine esecuzione, alcune statistiche. Queste saranno oggetto delle domande successive: è possibile premere il tasto **ESCape** per anticipare la fine della simulazione e la stampa delle statistiche in qualsiasi momento, anche senza attendere la terminazione dell'intera simulazione.)*

Ciascuna persona possiede una posizione di partenza e ad ogni passo della simulazione (vedi il metodo `dist.pers.Persona.mossa()`) si sposta in una cella adiacente. Nel codice fornito la scelta della cella su cui spostarsi è, al momento, ancora casuale. Quando più persone occupano la medesima posizione viene modellato il contatto avvenuto tra di esse tramite la creazione di oggetti di tipo `dist.sim.Contatto`, oggetti che vengono successivamente conservati per il calcolo delle statistiche finali (vedi metodo `dist.sim.Simulatore.simula()`). Tutte le persone che sono coinvolte in tale contatto vengono registrate nell'oggetto stesso di tipo `Contatto`.

Un progettista esperto fa notare che, nell'ottica di modellare i diversi tipi di comportamenti delle persone, conviene introdurre alcune nuove classi (**Attenta** e **Distratta**), per poi ristrutturare il codice trasformando la classe **Persona** in una classe astratta che le generalizzi.

DOMANDA 2 (50%)

Pertanto il progettista esperto suggerisce di ristrutturare l'applicazione come segue:

- a) **(30%)** Rifattorizzare la classe `dist.pers.Persona`, rendendola astratta. Introdurre contestualmente due nuove classi `dist.pers.Attenta` e `dist.pers.Distratta` che la estendano. Assegnare alle persone *attente* l'icona bianca e alle persone *distratte* l'icona rossa: (vedi anche `dist.gui.CostantiGUI`). Ogni persona deve possedere un intero identificatore ("**id**") progressivo base 0 assegnato sulla base della propria tipologia, ed incrementato ogni qualvolta una nuova persona di quella tipologia viene creata: a supporto e precisazione di questo requisito completare e ricorrere all'esecuzione dei test di unità presenti nella classe `dist.persone.PersonaTest`. Verificare ed eventualmente correggere il codice principale affinché questi test abbiano sempre successo. Implementare le modifiche suggerite dal progettista esperto, quindi verificare il funzionamento dell'applicazione anche dopo le modifiche. Per far ciò è necessario modificare la popolazione iniziale della simulazione per farvi apparire tutte le tipologie, in egual numero, come specificato dalla costante `dist.sim.CostantiSimulazione.NUMERO_INIZIALE_PER_TIPOLOGIA`.
- b) **(15%)** Assegnare ad ogni tipo di persona una nuova strategia per decidere il proprio movimento. Modificare il metodo `mossa()` di modo che le persone *attente* si muovano sempre in una casella adiacente, ma solo se non già occupata (le informazioni riguardo le celle della simulazione e gli eventuali occupanti si trovano nella classe `dist.sim.Ambiente`). Le persone *distratte* invece continueranno a muoversi sempre in una casella adiacente della corrente, scelta in maniera completamente casuale.

Le domande che seguono richiedono il completamento del corpo di alcuni metodi nella classe `dist.stats.Statistiche`: questi sono dedicati al calcolo delle statistiche prodotte al termine di ciascuna simulazione. Sono anche forniti a supporto dei metodi di stampa dei loro risultati per facilitarne la verifica del corretto funzionamento. Si suggerisce di studiare il sorgente della classe `dist.stat.Statistiche` ed in particolare il metodo `stampaFinale()` per i dettagli.

DOMANDA 3 (25%)

Dopo aver completato il punto precedente:

- completare il metodo `Map< Object, List< Object > > produciStatistica (Set< Object >)` nella classe `Statistiche`. Questo metodo deve scandire l'insieme degli oggetti di tipo `Contatto` che riceve come parametro, e generare una mappa che metta in relazione ogni **Persona** con la lista dei contatti in cui ha preso parte. Al fine di consultare meglio le statistiche, i contatti di ogni persona devono essere ordinati a seconda del numero crescente/decrecente di persone che hanno preso parte / passo in cui il contatto è avvenuto crescente/decrecente]. È necessario modificare la segnatura del metodo, sostituendo i riferimenti a `Object` con l'opportuno tipo. È possibile, se ritenuto necessario, modificare anche le classi `Contatto` e `Persona`.
- completare il corrispondente test-case `testProduciStatistica()` all'interno della classe `StatisticheTest`

DOMANDA 4 (25%)

Scrivere una o più classi di test (posizionandole corrispondentemente alla classe sotto test, e denominandole di conseguenza: ovvero all'interno della directory `src/test/` e chiamandola, ad esempio, `dist.pers.NomeClasseTest`), con test-case *minimali* per verificare il corretto funzionamento dei metodi che si occupano della scelta della prossima posizione di tutte le tipologie di persone. È possibile, ma solo se ritenuto conveniente e senza compromettere il resto del progetto, modificare anche il codice della classi oggetto della precedente domanda 2 per favorirne la *testabilità*.