

[2011-12] Monitor

Ci sono `numAgenti` agenti che si muovono nello 'spazio' di dimensione `x,y`. Inizialmente sono posizionati in **ordine inverso** nella prima riga:

A 10x10 dot grid. The top row contains the numbers 0 through 9 from left to right. The grid is used for plotting a graph.

Ogni agente si muove di una posizione (anche in diagonale) per raggiungere lo stato finale:

A blank coordinate grid with x and y axes ranging from 0 to 9. The x-axis is labeled 0 through 9, and the y-axis is labeled 0 through 9. The grid consists of 10 columns and 10 rows of points.

La gestione degli spostamenti avviene tramite un opportuno monitor Spazio con i seguenti metodi:

- `void register(int n, int x, int y):` 'registra' l'agente `n` nella posizione `x,y`. Questo metodo viene invocato una sola volta quando gli agenti vengono creati. Serve per inizializzare lo spazio nella configurazione iniziale.
- `boolean move(int n, int x, int y, int dx, int dy):` muove l'agente `n` dalla posizione `x,y` alla posizione `x+dx, y+dy`. Il valori `dx` e `dy` sono nel range `[-1,1]` in quanto gli agenti si spostano di una sola posizione. Se la posizione è occupata l'agente attende. Il metodo ritorna `false` nel caso l'agente `n` non sia nella posizione `x,y`.
- `int getAgent(int x, int y):` ritorna l'id dell'agente nella posizione `x,y`. Se la posizione è vuota ritorna `-1`. Questo metodo viene usato per stampare la situazione ed eseguire test.

Scopo della prova è implementare il monitor 'Spazio' con i tre metodi sopra descritti. Utilizzare il **programma di test** riportato qui sotto.

NOTA IMPORTANTE. COME AL SOLITO il fatto che il programma funzioni è solo una condizione necessaria: codice non commentato non verrà valutato. Inserire, inoltre, **DUE o TRE RIGHE DI SPIEGAZIONE GENERALE**

Programma di test

```
import java.util.HashSet;
import java.util.Set;

public class Test extends Thread {
    private static final int numAgenti = 10;           // numero agenti
    private static final int x=10, y=10;              // dimensione spazio di gioco

    private final int num;                            // id dell'agente
    private final Spazio s;                          // monitor spazio di gioco
    private int my_x, my_y;

    // costruttore: salva id, monitor e posizione iniziale dell'agente
    Test(int num, Spazio s) {
        this.num = num;
        this.s = s;
        this.my_x = numAgenti - 1 - num;
        this.my_y = 0;
    }

    public void run() {
        try {
            code();
        } catch (InterruptedException e) {
            System.out.println("Agente numero "+num+" interrotto!!");
        }
    }

    // codice dei thread
    void code() throws InterruptedException {
        int dx, dy, i, j, a;

        if (num == numAgenti) {
            // questo thread stampa solo la situazione e controlla interferenze
            // vedere nel ramo 'else' per il codice degli agenti
            boolean done = false;
            Set check = new HashSet ();

            // attende che tutti gli agenti siano registrati
            sleep(1000);

            // controlla la registrazione
            for (i=0; i
```