

Esercitazione 5 – Il robot

FASE 1 (minimo per la sufficienza)

Giorgio ha appena costruito un nuovo robot programmabile per il suo laboratorio di robotica, in grado di rispondere a ben cinque comandi di basso livello:

- 'R' (right): gira di 90° a destra;
- 'L' (left): gira di 90° a sinistra;
- 'F' (forward): avanza di 1 nella direzione in cui è orientato;
- 'B' (backward): indietreggia di 1 rispetto alla direzione in cui è orientato;
- 'X' (explode): attiva l'autodistruzione ed esplode.

Il Robot non ha sensori per cui deve memorizzare la sua posizione e la direzione a partire da quando viene acceso, che per lui corrisponde all'origine (0, 0) e orientato verso est (coordinate x crescenti).

Si chiede di realizzare la classe Robot in grado di eseguire i comandi elementari e di restituire la posizione corrente e l'orientamento corrente. La sua interfaccia è:

```
public class Robot{
    public Robot(String nome);
    public void right();
    public void left();
    public void forward();
    public void forward(int n);
    public void backward();
    public void backward(int n);
    public void explode();
    public int getX();
    public int getY();
    public int getDirection();    //0 = est,, 1 = nord, 2 =ovest, 3=sud
    public boolean isExploded();
    public String toString();    // Robot <nome> in (4, -2), direction: Ovest
                                // se esploso si aggiunge la dicitura EXPLODED in fondo.
}
```

FASE 2

A partire dai comandi elementari è possibile scrivere e memorizzare nel Robot dei comandi di alto livello, costituiti a sequenze di comandi elementari. Ad esempio, è possibile definire il comando "Quadrato1" costituito dalla sequenza FRFRFRFR che fa "disegnare" al Robot un quadrato di lato 1 e lo riposiziona nel punto di partenza con lo stesso orientamento.

Si intende quindi aggiungere alla classe Robot un metodo addCommand(String nome, String sequenza) in grado di memorizzare un nuovo comando di alto livello. Allo scopo di ridurre il consumo di memoria i comandi nuovi vengono memorizzati tutti in una unica lunga sequenza (il "programma") e sono identificati dalla posizione iniziale della sequenza nel programma e dalla lunghezza.

I nuovi comandi di alto livello sono memorizzati aggiungendo pezzi al "programma" solo se la sequenza non è già compresa. Se non è compresa, inoltre, si verifica che almeno la sua parte iniziale non sia già la parte terminale del "Programma", nel qual caso si memorizza solo la parte mancante (ad esempio, se il "Programma" contiene RFFRFRFL e il nuovo comando "Flip" ha sequenza LFFF, visto che LFFF non è presente ma il Programma termina con la prima parte, LF, a partire dalla posizione 6, si aggiunge al Programma il pezzo mancante FF e il comando "Flip" viene memorizzato con le "coordinate" posizione 6 e lunghezza 4.

A questo punto aggiungere anche il metodo execute(String nome) che esegue un comando se questo esiste.

L'interfaccia completa con codice di prova e casi di test minimali (solo per la prima parte) sono allegati.

Completare la classe.

Valutazione:

Fase 1 (minimo per la sufficienza): commenti completi, passaggio di tutti i test unit

Fase 2: voti >= 6 a seconda delle funzionalità realizzate e della correttezza del codice.