Logo Interpreter

di Arlind Canga

Implementazione media

Interfacce e Responsabilità

Durante la progettazione ho individuato 3 macro categorie di responsabilità:

- Una **View** che permette di interagire con l'utente e spedisce l'input al **Controller.**
- **Controller** che formatta in maniera corretta i comandi per poi passarli al **Model**.
- **Model** comprende tutti i componenti che costituiscono la logica del progetto. Prende l'input formattato dal Controller ed esegue i vari comandi.

Model

Come già detto in precedenza, qui ho raggruppato le componenti che costituiscono la logia del programma individuando le responsabilità quali:

- Un'executor (Interfaccia IManager), si occupa di interpretare i comandi che arrivano dal Controller e li indirizza alle varie componenti.
- Un'area di disegno che funge da contenitore delle figure e delle linee che si vengono a disegnare.
- Un Cursore che effettua i spostamenti in base ai comandi che l'executor lancia;
- Regole per la valutazione delle figure chiuse.

Le interfacce presenti in Model sono:

- **Shape**: offre una “descrizione” generale alle varie figure che si vengono a creare nell’area di disegno.
- **Environment**: estende Shape e rappresenta l’area di disegno; si occupa della gestione dei segmenti e di eventuali figure chiuse che si vengono a creare.
- **Line**: rappresenta un segmento nell’area di disegno.
- **Polygon**: rappresenta eventuali figure chiuse che si verranno a creare nell’area di disegno. Viene implementata da una classe astratta (**abstractClosedPolygon**) che a sua volta viene estesa dalla classe **trianglePolygon** che, sovrascrivendo gli opportuni metodi della classe padre, offre le regole per individuare un Polygono di 3 lati. Questo meccanismo rende piu’ facile implementare nuovi tipi di poligoni.
- **Turtle**: interfaccia che rappresenta il Cursore. Gestisce i movimenti di quest’ultimo e le varie personalizzazioni che offre (settaggio Colori, dimensioni penna, ecc); sfrutta delle classi di comodo per gestire l’angolazione del Cursore e la gestione delle sue coordinate.
- **factoryCreator:** e’ un’interfaccia con due metodi statici utili alla creazione di Linee e Poligoni. Questa interfaccia e’ pensata per rendere piu semplice l’implementazione di nuove figure.

Inoltre Model detiene le classi utili alla creazione di token (comandi) ed altre classi di comodo.

Controller

Qui risiedono le interfacce e le classi che hanno il compito di prendere l’input dell’utente, verificare la correttezza dei comandi (presi da un file o da linea di comando), creare una lista di token e spedirli al Model.

Sono presenti 3 componenti:

- Un Parser che ha la responsabilità di trasformare una stringa in una lista di token ;

- Un sender che prende la lista dei token e li spedice al model.
- Un reader per leggere file.

Le interfacce presenti sono 2:

- **IParseCMDS** che rappresenta il parser;
- **IAddresser** che rappresenta il sender.

View

In questo blocco ho raccolto due classi:

- Main del progetto (utile all'avvio del programma)
- userInterpreter che permette all'utente di inserire comandi, in seguito li trasforma in token (tramite il parser del Controller) e li da in pasto al sender.