

MINI-CAD

Scopo del progetto è la realizzazione di un mini interprete di comandi da integrare nell'applicazione per la manipolazione di oggetti grafici bidimensionali vista a lezione. Ai fini del progetto, si assume che esistano solo i tre tipi di oggetti grafici già realizzati: rettangoli, cerchi ed immagini.

L'interprete, la cui grammatica EBNF è riportata di seguito, supporta i seguenti comandi, alcuni dei quali sono reversibili (se ne può richiedere l'*undo*).

- Creazione di un nuovo oggetto grafico

Esempi:

- **create circle (5.0) (3.1,4.5)**
crea un cerchio di raggio 5.0 in posizione (3.1,4.5) e ne restituisce l'ID
- **create img ("./pippo.png") (6.1,4.6)**
crea un cerchio un'immagine a partire dal contenuto del file "./pippo.png" in posizione (6.1,4.6) e ne restituisce l'ID

- Rimozione di un oggetto o un gruppo di oggetti

Esempio:

- **del id1**
rimuove l'oggetto (o il gruppo di oggetti) identificato dall'ID id1

- Spostamento di un oggetto o un gruppo di oggetti

Esempi:

- **mv id1 (5.9,8.2)**
sposta l'oggetto (o il gruppo di oggetti) identificato dall'ID id1 nella posizione (5.9,8.2)
- **mvoff id1 (5.9,8.2)**
sposta l'oggetto (o il gruppo di oggetti) identificato dall'ID id1 nella posizione ottenuta sommando (5.9,8.2) alla posizione corrente

- Ridimensionamento di un oggetto o un gruppo di oggetti

Esempio:

- **scale id1 2.0**
ridimensiona l'oggetto (o il gruppo di oggetti) identificato dall'ID id1 con un fattore di scala pari a 2.0

- Visualizzazione delle proprietà di un oggetto, un gruppo di oggetti o di un tipo di oggetti

Esempi:

- **ls id1**
visualizza le proprietà dell'oggetto identificato da id1 o l'elenco degli oggetti parte del gruppo identificato da id1
- **ls circle**
visualizza l'elenco degli oggetti di tipo circle
- **ls all**
visualizza l'elenco di tutti gli oggetti
- **ls groups**

visualizza l'elenco di tutti i gruppi di oggetti

- Creazione di un gruppo di oggetti

Esempio:

- **grp id1, id2, id3**

crea un nuovo gruppo che contiene 3 i elementi identificati da id1, id2 e id3.

Ciascuno di essi può essere l'identificativo di un oggetto o di un gruppo. Il comando visualizza l'identificativo generato per il gruppo creato.

- Rimozione di un gruppo di oggetti

Esempio:

- **ungrp gid**

rimuove il gruppo identificato da gid (**senza** rimuovere i suoi componenti!)

- Calcolo area (perimetro)

Esempi:

- **area id1**

calcola l'area dell'oggetto identificato da id1 o la somma delle aree degli oggetti parte del gruppo identificato da id1

- **perimeter rectangle**

calcola la somma dei perimetri degli oggetti di tipo rectangle

- **area all**

calcola la somma delle aree di tutti gli oggetti.

La scelta dei comandi di cui supportare l'operazione di undo deve essere opportunamente motivata. Si suggerisce di utilizzare la classe StreamTokenizer per l'implementazione dell'analizzatore lessicale. Come spunto di partenza si possono utilizzare le classi viste a lezione per la realizzazione la realizzazione dell'interprete per le combinazioni.

Grammatica

<cmd> ::= <create> | <remove> | <move> | <scale> | <list> | <group> | <ungroup> | <area> | <perimeter>

<create> ::= new <typeconstr> <pos>

<remove> ::= del <objID>

<move> ::= mv <objID> <pos> | mvoff <objID> <pos>

<scale> ::= scale <objID> <posfloat>

<list> ::= ls <objID> | ls <type> | ls all | ls groups

<group> ::= grp <listID>

<ungroup> ::= ungrp <objID>

<area> ::= area <objID> | area <type> | area all

<perimeter> ::= perimeter <objID> | perimeter <type> | perimeter all

<pos> ::= (<posfloat> , <posfloat>)

<typeconstr> ::= circle (<posfloat>) | rectangle <pos> | img (<path>)

<type> ::= circle | rectangle | img

<listID> ::= <objID> { , <objID> }

Simboli terminali (token)

new

del

mv

mvoff

scale

ls

all

groups

grp

ungrp

area

perimeter

circle

rectangle

img

(

)

,

<posfloat>:= numero floating point **non negativo**

<objID>:= un identificatore

<path>:= un percorso valido di file

Nello sviluppo del progetto si devono utilizzare i Design Pattern ritenuti più adeguati motivandone opportunamente la scelta. Le fasi del processo di sviluppo devono essere documentate ricorrendo, ove necessario, all'uso di diagrammi UML.

Si richiede inoltre di effettuare il testing di uno o più moduli significativi impiegando un opportuno criterio e sfruttando le funzionalità offerte dal framework Junit.