Indice

1. Manuale utente
2. Presentazione grammatica
   1. Configurazione
   2. Lista movimenti
   3. Uscita
3. Java
   1. Gestione dati e metodi
   2. Gestione errori
4. Testing del tool
   1. Codice corretto
   2. Codice errato
5. Manuale utente

GCODEParser è un JAR eseguibile dopo aver installato JDK o JRE sul proprio computer windows. Il tool permette di estrapolare le informazioni da un file scritto in GCode e di visualizzare i comandi del programma con una breve descrizione e una rappresentazione grafica.



Cliccato i tasto «Sfoglia» il tool ci permette di scegliere un file, contenente il Gcode, da qualsiasi posizione all’interno della memoria di massa della nostra macchina

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamenteInfine si possono visualizzare i comandi e la rappresentazione grafica a schermo.

1. Presentazione grammatica

La grammatica utilizzata permette di riconoscere programmi scritti in GCode è la seguente:



Abbiamo scelto di dividere il programma in tre parti:

* Configurazione
* Lista movimenti
* Uscita (termine operazione)
  1. Configurazione

Nonostante il GCode sia un linguaggio abbastanza libero per quanto riguarda la struttura del codice, la nostra grammatica descrive una struttura meno dinamica.

La configurazione è descritta come segue:

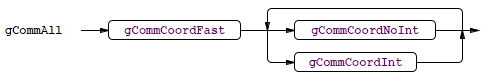
Seguendo lo schema, la grammatica è formata da:

* Selezione coordinate assolute o relative (GCODESCOORD)
* Selezione macchina e utensile (TCODES)
* Cambio utensile (MCODES)

La grammatica continua con:

* Selezione unità di misura velocità di movimento (GCODESF)
* Selezione unità di misura velocità di taglio (GCODESS)
* Velocità di movimento (FCODES)
* Velocità di taglio (SCODES)
* Configurazioni aggiuntive (mConfig)
  1. Lista movimenti

La lista dei movimenti, che descrivono il lavoro svolto dalla macchina, è così strutturata:



Il movimento che l’utensile può effettuare è di tre tipi:

* Lineare veloce (gCommCoordFast)
* Lineare (gCommCoordNoInt)
* Circolare (gCommCoordInt)

Il movimento lineare veloce viene utilizzato per raggiungere velocemente il punto iniziale della lavorazione.



È formato da:

* Il codice univoco «G00» (GCODESFAST)
* Coordinate x (XCOORD) e y (YCOORD) nel piano

Il movimento lineare viene utilizzato per l’operazione di taglio che congiunge due punti nel piano.



È formato da:

* Il codice univoco «G01» (GCODESINT)
* Coordinate x (XCOORD) e y (YCOORD) nel piano del punto da raggiungere

Il movimento circolare viene utilizzato per l’operazione di taglio che congiunge due punti nel piano tramite interpolazione.

È formato da:

* Il codice univoco «G02» (rotazione oraria) oppure «G03» (rotazione antioraria) (GCODESINTCIRC)
* Coordinate x (XCOORD) e y (YCOORD) nel piano del punto da raggiungere
* Coordinate i (ICOORD) e j (JCOORD) del centro dell’arco di circonferenza
  1. Uscita

il termine delle operazioni viene effettuato nell’ultima parte del programma:



Viene effettuato un movimento lineare veloce (gCommCoordFast) lontano dal pezzo lavorato e vengono specificati alcuni codici per spegnere la macchina e concludere la lavorazione (mConfig).

1. Java
   1. Gestione dati e metodi

L’interfaccia è stata implementata come componente separato dal launcher, utilizzando componenti sia di Java Swing che AWT. Di seguito una breve documentazione dei metodi dell’interfaccia:



Costruttore che si occupa di impostare e mostrare tutti i componenti dell’interfaccia.



Si occupa di mostrare le informazioni ottenute dall’handler.



Mostra il selettore del file da leggere



Metodo lanciato dal listener sul pulsante «Sfoglia», a sua volta si ricollega alla classe ParserLauncher.java per lanciare il selettore del file e le operazioni di visualizzazione delle informazioni

Proseguiamo con i metodi del Launcher:

Imposta l’oggetto da dare dal parser, restituendo tutto il file selezionato.

Metodo richiamato dall’interfaccia che avvia il processo di selezione del file e riconoscimento.



Istanzia l’interfaccia

La classe che si occupa di creare la lista dei movimenti e della configurazione è l’Handle, i cui metodi più significativi sono i seguenti:



Metodo che riceve il codice del tipo di movimento lineare, le coordinate del punto finale e lo aggiunge all’arrayList dei movimenti.



Metodo che riceve il codice del tipo di movimento circolare, le coordinate del punto finale, le coordinate del centro e lo aggiunge all’arrayList dei movimenti.



Metodo che riconosce tutti i codici forniti nella configurazione.

* 1. Gestione degli errori

Vengono effettuati controlli sulla presenza e correttezza dei codici e si controlla che ogni campo contenente le informazioni del brano sia di lunghezza pari a 30 caratteri.

Sfruttiamo il riconoscimento delle eccezioni di ANTLR per capire che tipo di errore riceviamo durante il parsing

Alcuni esempi:

* Codice errato
* Formato coordinate errato

1. Testing del tool
   1. Codice corretto

Esempio di codice corretto:

Immagine che contiene testo, tavolo

Descrizione generata automaticamente

Il tool restituisce la configurazione corrispondente alle prime due righe di codice:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamenteLa lista dei movimenti da effettuare:

La configurazione di uscita:

Immagine che contiene testo

Descrizione generata automaticamente

E infine una rappresentazione grafica del lavoro della macchina:

Immagine che contiene piazza

Descrizione generata automaticamente

* 1. Codice errato

Immagine che contiene testo, interni, screenshot

Descrizione generata automaticamenteNel caso in cui il programma presenti errori, viene stampata una lista di errori e avvisi dopo aver effettuato il parsing:

In questo caso ci sono errori lessicali (codice errato) e sintattici (formato errato) e anche un avviso che riguarda un possibile errore nel taglio circolare. Infatti se l’utensile non è in grado di collegare due punti tramite interpolazione, il tool avvisa l’utente del possibile sbaglio.