Bruno Henrique de Carvalho Scaglione - 10335812

PMR 3402
Sistemas Embarcados
Parser Código G

Brasil

3 de fevereiro de 2021

1 Código Escrito

Função main que vai ler arquivo .txt com código G e vai criar um arquivo output.txt com as coordenadas

main.py

```
import sys
from antlr4 import *
from gcode2Lexer import gcode2Lexer
from gcode2Parser import gcode2Parser
from Txtgcode2Listener import Txtgcode2Listener

def main(argv):
  # txt com código G
  input = FileStream(argv[1])
  lexer = gcode2Lexer(input)
  stream = CommonTokenStream(lexer)
  parser = gcode2Parser(stream)
  tree = parser.prg()
  output = open("output.txt","w")

  htmlgcode2 = Txtgcode2Listener(output)
  walker = ParseTreeWalker()
  walker.walk(htmlgcode2, tree)

  output.close()

if __name__ == '__main__':
  main(sys.argv)
```

Listener personalizado

Txtgcode2Listener.py

```
import sys
from antlr4 import *
from gcode2Parser import gcode2Parser
from gcode2Listener import gcode2Listener

class Txtgcode2Listener(gcode2Listener) :

  def __init__(self, output):
    self.output = output
    # self.output.write('<html><head><meta charset="UTF-8"/></head><body>')

  # Exit a parse tree produced by gcode2Parser#coordx.
  def exitCoordx(self, ctx):
    txt = ctx.getText()
    number = txt.strip("X")
    number = number + ' '
    self.output.write(number)

  # Exit a parse tree produced by gcode2Parser#coordy.
  def exitCoordy(self, ctx):
    txt = ctx.getText()
    number = txt.strip("Y")
    number = number + '\n'
    self.output.write(number)
```

Vai testar dois casos. Uma linha válida e outra inválida. Obterá um relatório de teste.

testgcode2Parser.py

```
from antlr4 import *
from gcode2Lexer import gcode2Lexer
from gcode2Parser import gcode2Parser
from Txtgcode2Listener import Txtgcode2Listener
from gcode2ErrorListener import gcode2ErrorListener
import unittest
import io

class Testgcode2Parser(unittest.TestCase):
  def setup(self, text):
    lexer = gcode2Lexer(InputStream(text))
    stream = CommonTokenStream(lexer)
    parser = gcode2Parser(stream)

    self.output = io.StringIO()
    self.error = io.StringIO()

    parser.removeErrorListeners()
    errorListener = gcode2ErrorListener(self.error)
    parser.addErrorListener(errorListener)

    self.errorListener = errorListener

    return parser

  def test_valid_gcode(self):
    # exemplo de linha que é válida
    parser = self.setup("N001 G00 X10 Y10\n")
    tree = parser.statement()

    Txtgcode2 = Txtgcode2Listener(self.output)
    walker = ParseTreeWalker()
    walker.walk(Txtgcode2, tree)

    # let's check that there aren't any symbols in errorListener
    self.assertEqual(len(self.errorListener.symbol), 0)

  def test_invalid_gcode(self):
    # exemplo de linha que não é válida
    parser = self.setup("N001 G00 X10 Y10X")
    tree = parser.statement()

    Txtgcode2 = Txtgcode2Listener(self.output)
    walker = ParseTreeWalker()
    walker.walk(Txtgcode2, tree)

    # let's check the symbol in errorListener
    self.assertEqual(self.errorListener.symbol[0], 'X')

if __name__ == '__main__':
  unittest.main()
```

Captura erros e guarda numa lista

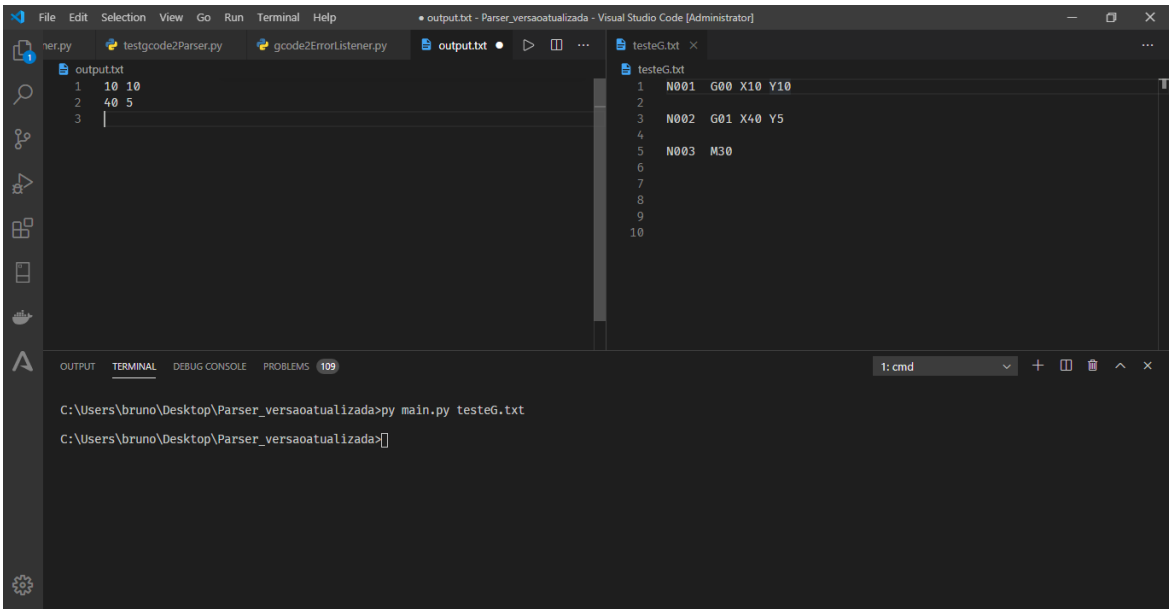
gcode2ErrorListener.py

```
import sys
from antlr4 import *
from gcode2Parser import gcode2Parser
from gcode2Listener import gcode2Listener
from antlr4.error.ErrorListener import *
import io

class gcode2ErrorListener(ErrorListener):
  def __init__(self, output):
    self.output = output
    self._symbol = []

  def syntaxError(self, recognizer, offendingSymbol, line, column, msg, e):
    self.output.write(msg)
    self._symbol.append(offendingSymbol.text)

  @property
  def symbol(self):
    return self._symbol
```

2 Programa Rodando

Escrever as coordenadas no txt

Figura 1 – py main.py testeG.txt



Capturar erros

Figura 2 – python -m unittest testgcode2Parser.py