Shawn Zhong & Ziyi Zhang

CS 536: Intro to PLs and Compilers

Program 2

Feb. 17, 2019

How to invoke

Run `make && make test` in the command line.

If there is no output after `java -cp ./deps:. P2` and `diff allTokens.in allTokens.out`, then all tests are passed

Others

To know more about the details of the program, see the comments of each class and methods.

You can also see our javadocs pages starting from the next page.

Class P2

java.lang.Object
    P2

```
public class P2
extends java.lang.Object
```

This program is to be used to test the Carrot scanner. This version is set up to test all tokens, but more code is needed to test other aspects of the scanner (e.g., input that causes errors, character numbers, values associated with tokens).

Field Summary

Fields

Modifier and Type	Field and Description
`private static java.lang.String[]`	**`genericTokenInputs`**
`private static java.lang.String[]`	**`idTokenInputs`**
`private static java.lang.String[]`	**`intLitTokenInputs`**
`private static java.lang.String[]`	**`strLitTokenInputs`**
`private static java.lang.String[][]`	**`tokenInputs`**

Constructor Summary

Constructors

Constructor and Description
`P2``()`

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and Description
`private static java.lang.String`	**`addQuotes`**`(java.lang.Object str)` Add Quotes to the string
`private static void`	**`checkIdTokenVal`**`(java.lang.String input, java.lang.String expectedValue, java.lang.String expectedErrorMessage)` Check if Id TokenVal matches as expected

private static void	**checkIntLitTokenVal**(java.lang.String input, java.lang.Integer expectedValue, java.lang.String expectedErrorMessage) Check if Integer TokenVal matches as expected
private static void	**checkStrLitTokenVal**(java.lang.String input, java.lang.String expectedValue, java.lang.String expectedErrorMessage) Check if String TokenVal matches as expected
private static void	**compareTokenVal**(java.lang.String input, **TokenVal** actualTokenval, **TokenVal** expectedTokenVal) This methods compares two TokenVals and print their difference, including the char number, line number, token type and token value
private static **TokenVal**	**getTokenAndCheckErrorMessage**(java.lang.String input, java.lang.String expectedErrorMessage) This method check if the Error message matches as expected and return the TokenVal
private static **TokenVal**[]	**getTokens**(java.lang.String[] input, java.lang.String delim) This methods takes in a list of input, combine them with delimiter and return a list of TokenVals
static void	**main**(java.lang.String[] args) The main function
private static java.lang.String	**parseErrorMessage**(java.io.ByteArrayOutputStream buffer) Parse Error Message
private static void	**printTestMessage**(java.lang.String message, java.lang.Object expected, java.lang.Object actual) Print out Test Message
private static void	**testAllTokens**() testAllTokens
private static void	**testCharNum**() Test if Character Number is calculated correctly
private static void	**testErrors**() Test if the error thrown matches the expected error
private static void	**testLineNum**() Test if Line Number is calculated correctly
private static void	**testVal**() Test if the value of the token matches the expected value

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

genericTokenInputs

```
private static final java.lang.String[] genericTokenInputs
```

strLitTokenInputs

```
private static final java.lang.String[] strLitTokenInputs
```

intLitTokenInputs

```
private static final java.lang.String[] intLitTokenInputs
```

idTokenInputs

```
private static final java.lang.String[] idTokenInputs
```

tokenInputs

```
private static final java.lang.String[][] tokenInputs
```

Constructor Detail

P2

```
public P2()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
                 throws java.io.IOException
```

The main function

Throws:

`java.io.IOException`

testVal

```
private static void testVal()
                     throws java.io.IOException
```

Test if the value of the token matches the expected value

Throws:

`java.io.IOException`

testErrors

```
private static void testErrors()
                         throws java.io.IOException
```

Test if the error thrown matches the expected error

Throws:

`java.io.IOException`

checkStrLitTokenVal

```
private static void checkStrLitTokenVal(java.lang.String input,
                                        java.lang.String expectedValue,
                                        java.lang.String expectedErrorMessage)
                                 throws java.io.IOException
```

Check if String TokenVal matches as expected

Throws:

`java.io.IOException`

checkIntLitTokenVal

```
private static void checkIntLitTokenVal(java.lang.String input,
                                        java.lang.Integer expectedValue,
                                        java.lang.String expectedErrorMessage)
                                 throws java.io.IOException
```

Check if Integer TokenVal matches as expected

Throws:

`java.io.IOException`

checkIdTokenVal

```
private static void checkIdTokenVal(java.lang.String input,
                                    java.lang.String expectedValue,
                                    java.lang.String expectedErrorMessage)
                             throws java.io.IOException
```

Check if Id TokenVal matches as expected

Throws:

java.io.IOException

compareTokenVal

```
private static void compareTokenVal(java.lang.String input,
                                    TokenVal actualTokenval,
                                    TokenVal expectedTokenVal)
```

This methods compares two TokenVals and print their difference, including the char number, line number, token type and token value

getTokens

```
private static TokenVal[] getTokens(java.lang.String[] input,
                                    java.lang.String delim)
                             throws java.io.IOException
```

This methods takes in a list of input, combine them with delimiter and return a list of TokenVals

Throws:

java.io.IOException

getTokenAndCheckErrorMessage

```
private static TokenVal getTokenAndCheckErrorMessage(java.lang.String input,
                                                     java.lang.String expectedErrorMessage)
                                              throws java.io.IOException
```

This method check if the Error message matches as expected and return the TokenVal

Throws:

java.io.IOException

printTestMessage

```
private static void printTestMessage(java.lang.String message,
                                     java.lang.Object expected,
                                     java.lang.Object actual)
```

Print out Test Message

addQuotes

```
private static java.lang.String addQuotes(java.lang.Object str)
```

Add Quotes to the string

parseErrorMessage

```
private static java.lang.String parseErrorMessage(java.io.ByteArrayOutputStream buffer)
```

Parse Error Message

testCharNum

```
private static void testCharNum()
                           throws java.io.IOException
```

Test if Character Number is calculated correctly

Throws:

`java.io.IOException`

testLineNum

```
private static void testLineNum()
                           throws java.io.IOException
```

Test if Line Number is calculated correctly

Throws:

`java.io.IOException`

testAllTokens

```
private static void testAllTokens()
                            throws java.io.IOException
```

testAllTokens

Open and read from file allTokens.txt For each token read, write the corresponding string to allTokens.out If the input file contains all tokens, one per line, we can verify correctness of the scanner by comparing the input and output files (e.g., using a 'diff' command).

Throws:

`java.io.IOException`