

Token.java

```
1  /* Generated By:JavaCC: Do not edit this line. Token.java Version 3.0 */
2  package jminusminus;
3
4  /**
5   * Describes the input token stream.
6   */
7
8  public class Token {
9
10     /**
11      * An integer that describes the kind of this token. This numbering
12      * system is determined by JavaCCParser, and a table of these numbers is
13      * stored in the file ...Constants.java.
14      */
15     public int kind;
16
17     /**
18      * beginLine and beginColumn describe the position of the first character
19      * of this token; endLine and endColumn describe the position of the
20      * last character of this token.
21      */
22     public int beginLine, beginColumn, endLine, endColumn;
23
24     /**
25      * The string image of the token.
26      */
27     public String image;
28
29     /**
30      * A reference to the next regular (non-special) token from the input
31      * stream. If this is the last token from the input stream, or if the
32      * token manager has not read tokens beyond this one, this field is
33      * set to null. This is true only if this token is also a regular
34      * token. Otherwise, see below for a description of the contents of
35      * this field.
36      */
37     public Token next;
38
39     /**
40      * This field is used to access special tokens that occur prior to this
41      * token, but after the immediately preceding regular (non-special) token.
42      * If there are no such special tokens, this field is set to null.
43      * When there are more than one such special token, this field refers
44      * to the last of these special tokens, which in turn refers to the next
45      * previous special token through its specialToken field, and so on
46      * until the first special token (whose specialToken field is null).
47      * The next fields of special tokens refer to other special tokens that
48      * immediately follow it (without an intervening regular token). If there
49      * is no such token, this field is null.
50      */
51     public Token specialToken;
52
53     /**
54      * Returns the image.
55      */
56     public String toString()
57     {
58         return image;
59     }
60
61     /**
62      * Returns a new Token object, by default. However, if you want, you
63      * can create and return subclass objects based on the value of ofKind.
64      * Simply add the cases to the switch for all those special cases.
65      * For example, if you have a subclass of Token called IDToken that
66      * you want to create if ofKind is ID, simply add something like :
```

```
67      *
68      *      case MyParserConstants.ID : return new IDToken();
69      *
70      * to the following switch statement. Then you can cast matchedToken
71      * variable to the appropriate type and use it in your lexical actions.
72      */
73      public static final Token newToken(int ofKind)
74      {
75          switch(ofKind)
76          {
77              default : return new Token();
78          }
79      }
80
81 }
82
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder