```java
// Copyright 2013 Bill Campbell, Swami Iyer and Bahar Akbal-Delibas

package jminusminus;

/**
 * An enum of token kinds. Each entry in this enum represents the kind of a
 * token along with its image (string representation).
 *
 * When you add a new token to the scanner, you must also add an entry to this
 * enum specifying the kind and image of the new token.
 */

enum TokenKind {
    EOF("<EOF>"), ABSTRACT("abstract"), BOOLEAN("boolean"), CHAR("char"), CLASS(
            "class"), ELSE("else"), EXTENDS("extends"), FALSE("false"), IF("if"),
IMPORT(
            "import"), INSTANCEOF("instanceof"), INT("int"), NEW("new"), NULL(
            "null"), PACKAGE("package"), PRIVATE("private"), PROTECTED(
            "protected"), PUBLIC("public"), RETURN("return"), STATIC("static"),
SUPER(
            "super"), THIS("this"), TRUE("true"), VOID("void"), WHILE("while"),
PLUS(
            "+"), ASSIGN("="), DEC("--"), EQUAL("=="), GT(">"), INC("++"), LAND(
            "&&"), LE("<="), LNOT("!"), MINUS("-"), PLUS_ASSIGN("+="), STAR("*"),
LPAREN(
            "("), RPAREN(")"), LCURLY("{"), RCURLY("}"), LBRACK("["), RBRACK(
            "]"), SEMI(";"), COMMA(","), DOT("."), IDENTIFIER("<IDENTIFIER>"),
INT_LITERAL(
            "<INT_LITERAL>"), CHAR_LITERAL("<CHAR_LITERAL>"), STRING_LITERAL(
            "<STRING_LITERAL>");

    /** The token's string representation. */
    private String image;

    /**
     * Construct an instance TokenKind given its string representation.
     *
     * @param image
     *            string representation of the token.
     */

    private TokenKind(String image) {
        this.image = image;
    }

    /**
     * Return the image of the token.
     *
     * @return the token's image.
     */

    public String image() {
        return image;
    }

    /**
     * Return the string representation of the token.
     *
     * @return the token's string representation.
     */

    public String toString() {
        return image;
    }

}
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

```
62
63  /**
64   * A representation of tokens returned by the lexical analyzer method,
65   * getNextToken(). A token has a kind identifying what kind of token it is, an
66   * image for providing any semantic text, and the line in which it occurred in
67   * the source file.
68   */
69
70  class TokenInfo {
71
72      /** Token kind. */
73      private TokenKind kind;
74
75      /**
76       * Semantic text (if any). For example, the identifier name when the token
77       * kind is IDENTIFIER. For tokens without a semantic text, it is simply its
78       * string representation. For example, "+=" when the token kind is
79       * PLUS_ASSIGN.
80       */
81      private String image;
82
83      /** Line in which the token occurs in the source file. */
84      private int line;
85
86      /**
87       * Construct a TokenInfo from its kind, the semantic text forming the token,
88       * and its line number.
89       *
90       * @param kind
91       *            the token's kind.
92       * @param image
93       *            the semantic text comprising the token.
94       * @param line
95       *            the line in which the token occurs in the source file.
96       */
97
98      public TokenInfo(TokenKind kind, String image, int line) {
99          this.kind = kind;
100         this.image = image;
101         this.line = line;
102     }
103
104     /**
105      * Construct a TokenInfo from its kind, and its line number. Its image is
106      * simply its string representation.
107      *
108      * @param kind
109      *            the token's identifying number.
110      * @param line
111      *            identifying the line on which the token was found.
112      */
113
114     public TokenInfo(TokenKind kind, int line) {
115         this(kind, kind.toString(), line);
116     }
117
118     /**
119      * Return the token's string representation.
120      *
121      * @return the string representation.
122      */
123
124     public String tokenRep() {
125         return kind.toString();
126     }
127
128     /**
129      * Return the semantic text associated with the token.
130      *
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

```
131      * @return the semantic text.
132      */
133
134     public String image() {
135         return image;
136     }
137
138     /**
139      * Return the line number associated with the token.
140      *
141      * @return the line number.
142      */
143
144     public int line() {
145         return line;
146     }
147
148     /**
149      * Return the token's kind.
150      *
151      * @return the kind.
152      */
153
154     public TokenKind kind() {
155         return kind;
156     }
157
158     /**
159      * Return the token's image.
160      *
161      * @return the image.
162      */
163
164     public String toString() {
165         return image;
166     }
167
168 }
169
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder