The StreamTokenizer class takes an input stream and parses it into "tokens", allowing the tokens to be read one at a time. The parsing process is controlled by a table and a number of flags that can be set to various states. The StreamTokenizer can recognize identifiers, numbers, quoted strings, and various comment styles.

A typical application first constructs an instance of this class, sets up the syntax tables (the default setting is usually sufficient), and then repeatedly loops calling the nextToken() method in each iteration of the loop until it returns the value TT_EOF.

This is an example:

Creation of a StreamTokenizer object

```
try {
  Reader r = new BufferedReader(new InputStreamReader(new FileInputStream("nomeFile")));
   StreamTokenizer st = new StreamTokenizer(r);
    } catch(IOException e) {
      System.out.println("File not found");
      System.exit(1);
    }
```

Use a StreamTokenizer object

```
 try {
      while(st.nextToken() != StreamTokenizer.TT_EOF) {
        String s;
        switch(st.ttype) {
          case StreamTokenizer.TT_EOL:


        s = new String("EOL");// an "end of line" is found and then changed in a string
        break;
      case StreamTokenizer.TT_NUMBER: // a number is found. Its value is stored in nval
        s = Double.toString(st.nval);
        break;
      case StreamTokenizer.TT_WORD: // a string is found. Its value is stored in sval
        s = st.sval;
        break;
      default: // a character is found. Its value is stored in ttype
        s = String.valueOf((char)st.ttype);
    }
    ........
  }
} catch(IOException e) {
  System.out.println(
    "st.nextToken() unsuccessful");
}
```