# main()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 **public** **static** **void** main(String[] args) **throws** IOException {  2 String fname = **null**;  3 **if** (args.length == 0) { /\* if not given filename,take as '""' \*/  4 fname = **new** String();  5 } **else** **if** (args.length == 1) {  6 fname = args[1];  7 } **else** {  8 System.***out***.print("Error!,please give the token stream\n");  9 System.*exit*(0);  }  10 Printtokens t = **new** Printtokens();  11 BufferedReader br = t.open\_token\_stream(fname); /\* open token stream \*/  12 String tok = t.get\_token(br);  13 **while** (tok != **null**) { /\* take one token each time until eof \*/  14 t.print\_token(tok);  15 tok = t.get\_token(br);  }    16 System.*exit*(0);  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3 | 2 | 3 | | 2 | 4 | 4 | 4 | | 3 | 5 | 5 | 5 | | 4 | 6 | 6 | 6 | | 5 | 8, 9 | 8 | 9 | | 6 | 10, 11, 12 | 10 | 12 | | 7 | 13 | 13 | 13 | | 8 | 14, 15 | 14 | 15 | | 9 | 16 | 16 | 16 | |

True

True

True

False

False

False

# open\_character\_stream()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 BufferedReader open\_character\_stream(String fname) {  2 BufferedReader br = **null**;  3 **if** (fname == **null**) {  4 br = **new** BufferedReader(**new** InputStreamReader(System.***in***));  5 } **else** {  6 **try** {  7 FileReader fr = **new** FileReader(fname);  8 br = **new** BufferedReader(fr);  9 } **catch** (FileNotFoundException e) {  10 System.***out***.print("The file " + fname +" doesn't exists\n");  11 e.printStackTrace();  }  }  12 **return** **null**;  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3 | 2 | 3 | | 2 | 4 | 4 | 4 | | 3 | 6, 7, 8, 9, 10, 11 | 6 | 11 | |

True

False

# get\_char()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 **int** get\_char(BufferedReader br){  2 **int** ch = 0;  3 **try** {  4 br.mark(3);  5 ch= br.read();  6 } **catch** (IOException e) {  7 e.printStackTrace();  }  8 **return** ch;  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3, 4, 5, 6, 7 | 2 | 7 | | 2 | 8 | 8 | 8 | |

# unget\_char()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 **char** unget\_char (**int** ch,BufferedReader br) {  2 **try** {  3 br.reset();  4 } **catch** (IOException e) {  5 e.printStackTrace();  }  6 **return** 0;  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3, 4, 5 | 2 | 5 | | 2 | 6 | 6 | 6 | |

# open\_token\_stream()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 BufferedReader open\_token\_stream(String fname)  {  2 BufferedReader br;  3 **if**(fname.equals(**null**))  4 br=open\_character\_stream(**null**);  5 **else**  6 br=open\_character\_stream(fname);  7 **return** br;  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3 | 2 | 3 | | 2 | 4 | 4 | 4 | | 3 | 6 | 6 | 6 | |

True

False

# get\_token()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 String get\_token(BufferedReader br)  {  2 **int** i=0,j;  3 **int** id=0;  4 **int** res = 0;  5 **char** ch = '\0';    6 StringBuilder sb = **new** StringBuilder();  7 **try** {  8 res = get\_char(br);  9 **if** (res == -1) {  10 **return** **null**;  }  11 ch = (**char**)res;  12 **while**(ch=='\t'||ch=='\n' || ch == '\r') /\* strip all blanks until meet characters \*/  {  13 res = get\_char(br);  14 ch = (**char**)res;  }    15 **if**(res == -1)**return** **null**;  16 sb.append(ch);  17 **if**(*is\_spec\_symbol*(ch)==**true**)**return** sb.toString();  18 **if**(ch =='"') 19 id=2; /\* prepare for string \*/  20 **if**(ch ==59) 21 id=1; /\* prepare for comment \*/    22 res = get\_char(br);  23 **if** (res == -1) {  24 unget\_char(ch,br);  25 **return** sb.toString();  }  26 ch = (**char**)res;  27 **while** (*is\_token\_end*(id,res) == **false**)/\* until meet the end character \*/  {  28 sb.append(ch);  29 br.mark(4);  30 res = get\_char(br);  31 **if** (res == -1) {  32 **break**;  }  33 ch = (**char**)res;  }    34 **if**(res == -1) /\* if end character is eof token \*/  35 { unget\_char(ch,br); /\* then put back eof on token\_stream \*/  36 **return** sb.toString();  }    37 **if**(*is\_spec\_symbol*(ch)==**true**) /\* if end character is special\_symbol \*/  38 { unget\_char(ch,br); /\* then put back this character \*/  39 **return** sb.toString();  }  40 **if**(id==1) /\* if end character is " and is string \*/  {  41 sb.append(ch);  42 **return** sb.toString();  }  43 **if**(id==0 && ch==59)  /\* when not in string or comment,meet ";" \*/  44 { unget\_char(ch,br); /\* then put back this character \*/  45 **return** sb.toString();  }  46 } **catch** (IOException e) {  47 e.printStackTrace();  }    48 **return** sb.toString(); /\* return nomal case token \*/  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2, 3, 4, 5, 6, 7, 8, 9 | 2 | 9 | | 2 | 11 | 11 | 11 | | 3 | 12 | 12 | 12 | | 4 | 13, 14 | 13 | 14 | | 5 | 15 | 15 | 15 | | 6 | 16, 17 | 16 | 17 | | 7 | 18 | 18 | 18 | | 8 | 19 | 19 | 19 | | 9 | 20 | 20 | 20 | | 10 | 21 | 21 | 21 | | 11 | 22, 23 | 22 | 23 | | 12 | 24 | 24 | 24 | | 13 | 26 | 26 | 26 | | 14 | 27 | 27 | 27 | | 15 | 28, 29, 30, 31 | 28 | 31 | | 16 | 33 | 33 | 33 | | 17 | 34 | 34 | 34 | | 18 | 35 | 35 | 35 | | 19 | 37 | 37 | 37 | | 20 | 38 | 38 | 38 | | 21 | 40 | 40 | 40 | | 22 | 41 | 41 | 41 | | 23 | 43 | 43 | 43 | | 24 | 44 | 44 | 44 | | 25 | 46, 47 | 44 | 45 | |

True

False

True

False

True

False

True

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

True

False

# is\_token\_end()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 **static** **boolean** is\_token\_end(**int** str\_com\_id, **int** res)  {  2 **if**(res==-1) 3 **return**(**true**); /\* is eof token? \*/  4 **char** ch = (**char**)res;  5 **if**(str\_com\_id==1) /\* is string token \*/  6 { **if**(ch=='"' | ch=='\n' || ch == '\r') /\* for string until meet another " \*/  7 **return** **true**;  8 **else**  9 **return** **false**;  }  10 **if**(str\_com\_id==2) /\* is comment token \*/  11 { **if**(ch=='\n' || ch == '\r' || ch=='\t') /\* for comment until meet end of line \*/  12 **return** **true**;  13 **else**  14 **return** **false**;  }  15 **if**(*is\_spec\_symbol*(ch)==**true**) 16 **return** **true**; /\* is special\_symbol? \*/  17 **if**(ch ==' ' || ch=='\n'|| ch=='\r' || ch==59) 18 **return** **true**;    19 **return** **false**; /\* other case,return FALSE \*/  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2 | 2 | 2 | | 2 | 4, 5 | 4 | 5 | | 3 | 6 | 6 | 6 | | 4 | 10 | 10 | 10 | | 5 | 11 | 11 | 11 | | 6 | 15 | 15 | 15 | | 7 | 17 | 17 | 17 | |

True

False

True

False

True

False

True

False

# token\_type()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 static int token\_type(String tok)  {  2 if(is\_keyword(tok)) 3 return(keyword);  4 if(is\_spec\_symbol(tok.charAt(0))) 5 return(spec\_symbol);  6 if(is\_identifier(tok)) 7 return(identifier);  8 if(is\_num\_constant(tok)) 9 return(num\_constant);  10 if(is\_str\_constant(tok)) 11 return(str\_constant);  12 if(is\_char\_constant(tok)) 13 return(char\_constant);  14 if(is\_comment(tok)) 15 return(comment);  16 return(error); /\* else look as error token \*/  } | |  |  |  |  | | --- | --- | --- | --- | | Block | Lines | Entry | Exit | | 1 | 2 | 2 | 2 | | 2 | 3 | 3 | 3 | | 3 | 4 | 4 | 4 | | 4 | 5 | 5 | 5 | | 5 | 6 | 6 | 6 | | 6 | 7 | 7 | 7 | | 7 | 8 | 8 | 8 | | 8 | 9 | 9 | 9 | | 9 | 10 | 10 | 10 | | 10 | 11 | 11 | 11 | | 11 | 12 | 12 | 12 | | 12 | 13 | 13 | 13 | | 13 | 14 | 14 | 14 | | 14 | 15 | 15 | 15 | | 15 | 16 | 16 | 16 | |

False

False

True

False

True

False

False

False

False

True

True

True

True

True