جامعة أم القرى
UMM AL-QURA UNIVERSITY

scanner application

شهد ساعد الصبحي

[redacted]

What is a compiler?

A compiler is a special program that translates a programming language's source code into machine code, bytecode or another programming language. The source code is typically written in a high-level, human-readable language such as Java or C++. A programmer writes the source code in a code editor or an integrated development environment (IDE) that includes an editor, saving the source code to one or more text files. A compiler that supports the source programming language reads the files, analyzes the code, and translates it into a format suitable for the target platform.

The primary function of a ***scanner*** is to combine characters from the input stream into recognizable units called tokens. A method has been presented in this paper for designing such a scanner, also frequently referred to as a lexical analyser in the current literature. The major steps involved in this design process are: identification of tokens, construction of a state diagram, building driver tables and finally writing a scanning routine. The rules for generating the driver tables are described and an algorithm for the scanner, utilizing these driver tables, is included. The method has been successfully used to build the system scanner for a user oriented plotting language. It is concluded that the method is well defined, gives rise to a modular design and as such easily lends itself to language extensions.

Programming Language

The language for which the scanner is built is a language consisting of a group of lines. The transition from one line to another is done by “enter”, and in one line the words are separated by “space”. The language contains a set of reserved words (key word) : {do,for,if,while,else,int}, and a set of arithmetic operations {+,-,*,/}.

also a set of logical operations {&&,||,==,!=,<,>}, and the definition of variables in the language consists of letters and numbers, but it must begin with a letter. The language contains positive integers.

And the comments will starts with // then accept just positive integers or letters.

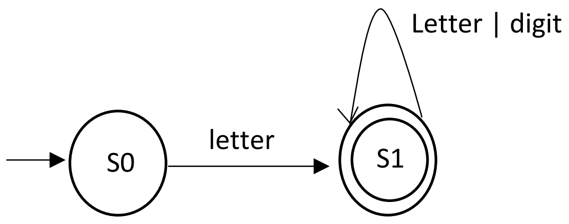
Scanner construction:

Letter: (a|b|c|….|z|A|B|C|…|Z)

Digit: (0|1|2|..|9)

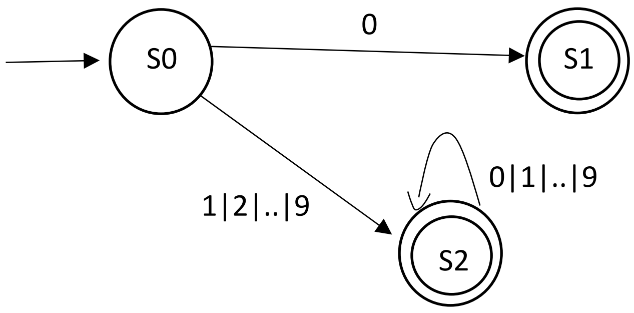
Regular Expressions of Identifier = Letter (digit | letter)*

DFA:



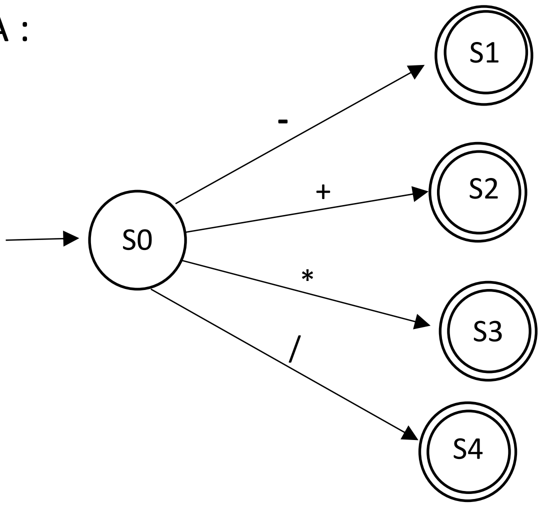
Regular Expressions of integer : (0 | <u>(1|2|..|9)(0|1|2|..|9)*</u>)

DFA:

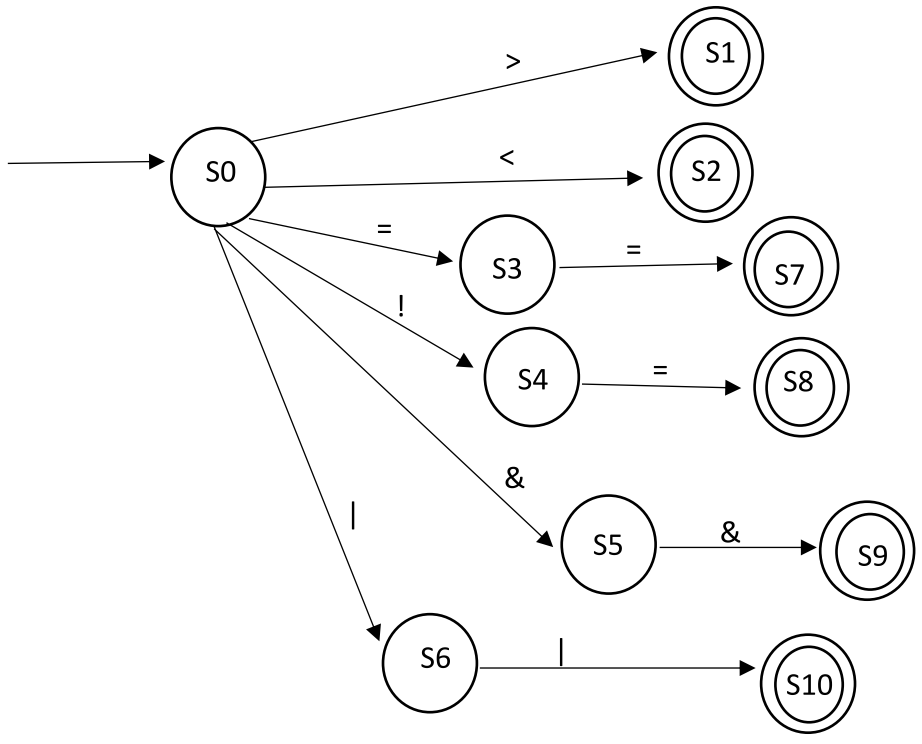


Regular Expressions of arithmetic operation: (+ | - | / | *)

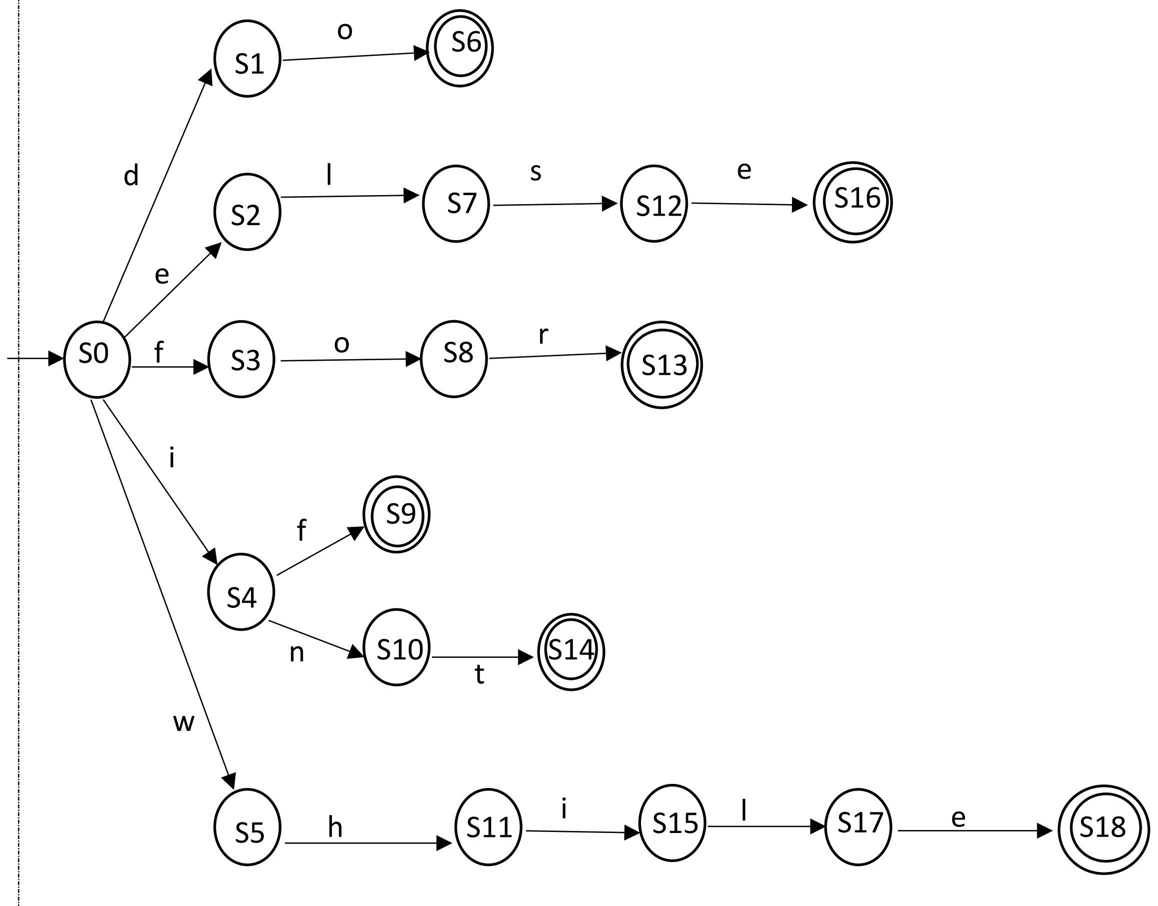
DFA :



Regular Expressions of logic operation:(&& | || | == |!= |<|>)



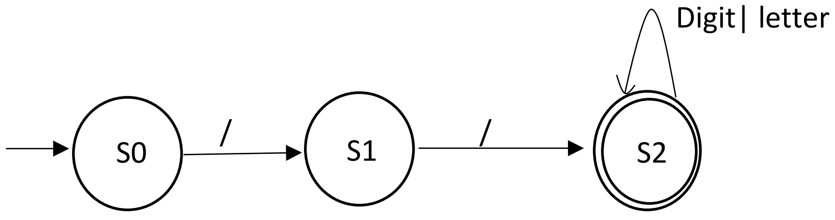
Regular Expressions of key words: (int|else|do|for|if|while)



Letter: (a|b|c|….|z|A|B|C|…|Z)

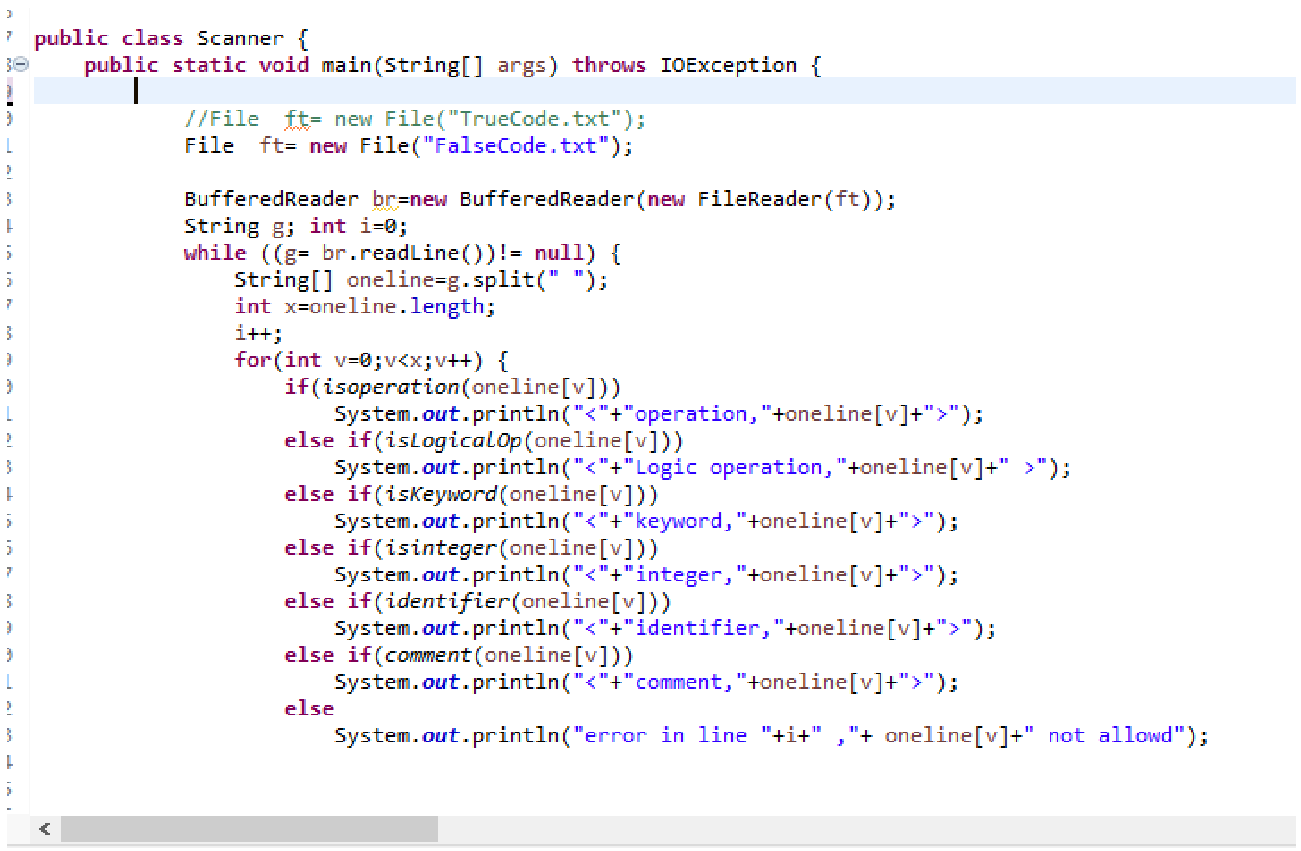
Digit: (0|1|2|..|9)

Regular Expressions of comment: //(letter|digit)*

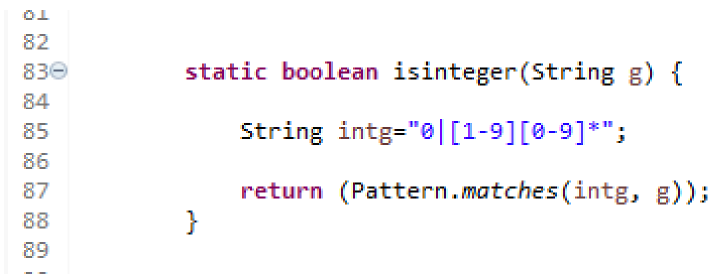


Implementation :

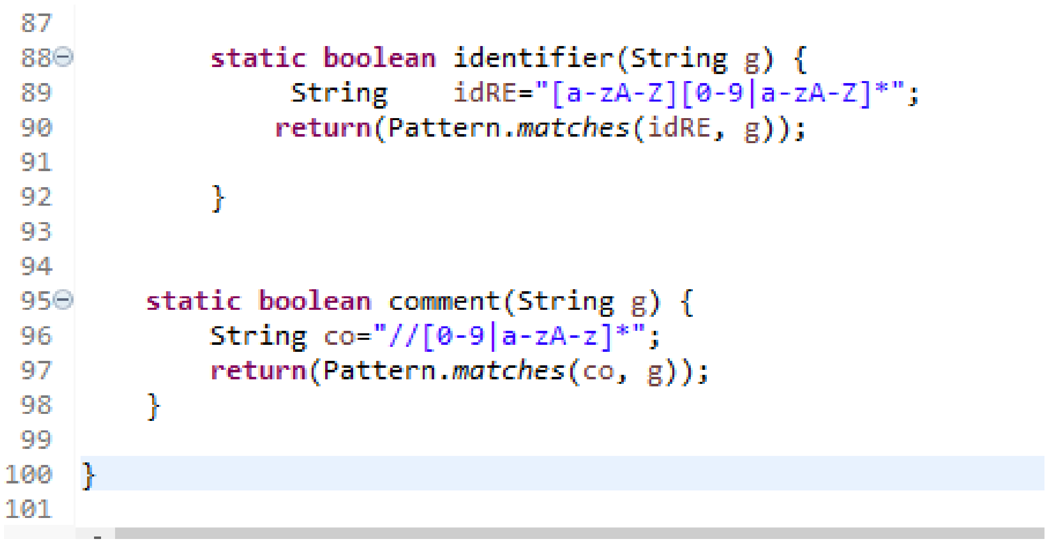
Scanner was built using java ,the program will read the first line in the text file and seprate the words in it, and print the token for each word or print error message if the word is not accepted.Then move to the second line and so on.



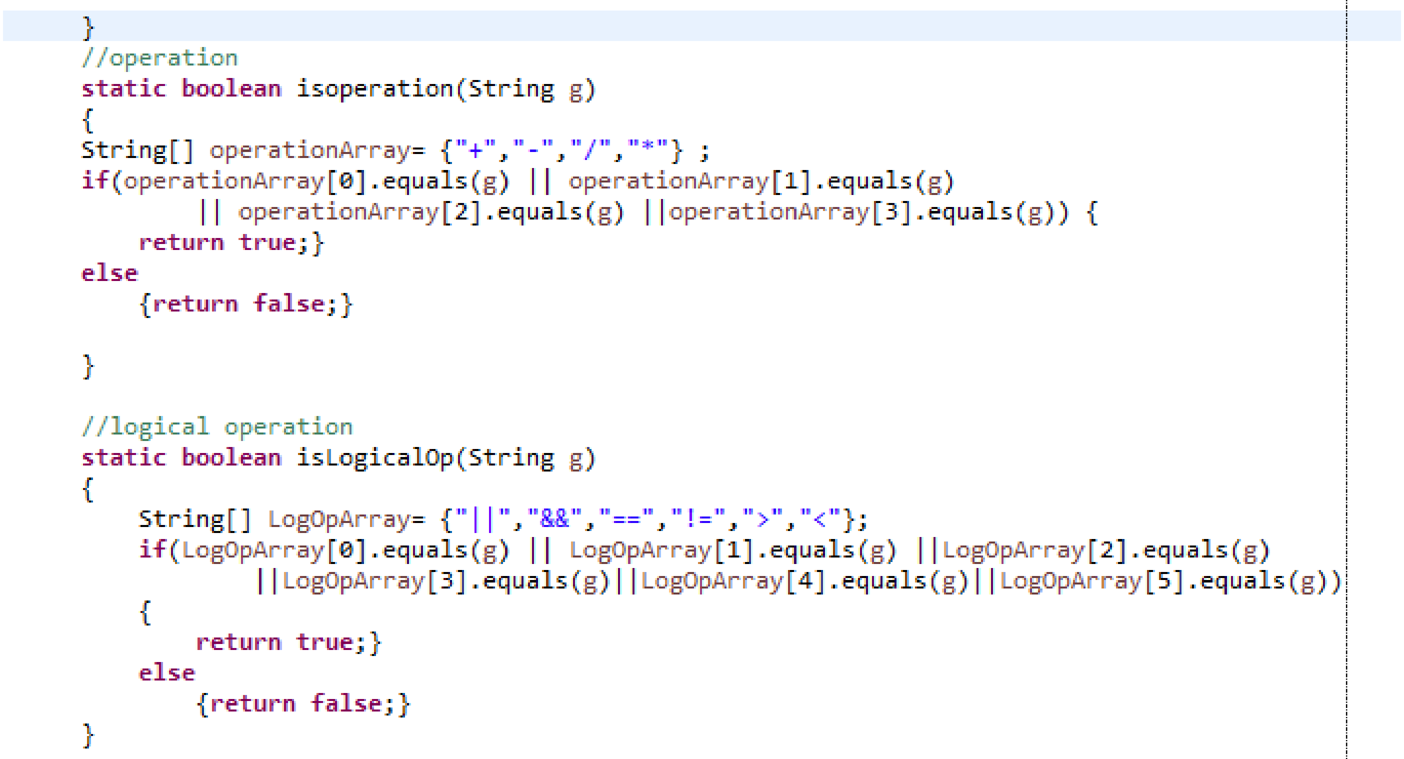
```java
public class Scanner {
    public static void main(String[] args) throws IOException {
            
            //File  ft= new File("TrueCode.txt");
            File  ft= new File("FalseCode.txt");
            
            BufferedReader br=new BufferedReader(new FileReader(ft));
            String g; int i=0;
            while ((g= br.readLine())!= null) {
                String[] oneline=g.split(" ");
                int x=oneline.length;
                i++;
                for(int v=0;v<x;v++) {
                    if(isoperation(oneline[v]))
                        System.out.println("<"+"operation,"+oneline[v]+">");
                    else if(isLogicalOp(oneline[v]))
                        System.out.println("<"+"Logic operation,"+oneline[v]+" >");
                    else if(isKeyword(oneline[v]))
                        System.out.println("<"+"keyword,"+oneline[v]+">");
                    else if(isinteger(oneline[v]))
                        System.out.println("<"+"integer,"+oneline[v]+">");
                    else if(identifier(oneline[v]))
                        System.out.println("<"+"identifier,"+oneline[v]+">");
                    else if(comment(oneline[v]))
                        System.out.println("<"+"comment,"+oneline[v]+">");
                    else
                        System.out.println("error in line "+i+" ,"+ oneline[v]+" not allowd");
```



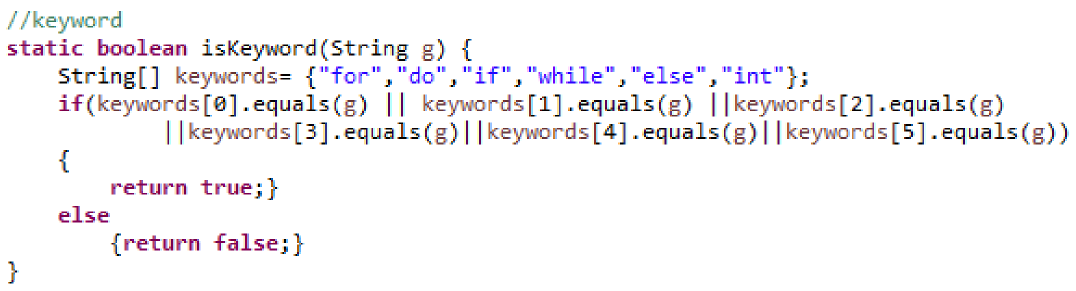
```
    static boolean isinteger(String g) {

        String intg="0|[1-9][0-9]*";

        return (Pattern.matches(intg, g));
    }

```



```

    static boolean identifier(String g) {
         String    idRE="[a-zA-Z][0-9|a-zA-Z]*";
        return(Pattern.matches(idRE, g));

    }


static boolean comment(String g) {
    String co="//[0-9|a-zA-z]*";
    return(Pattern.matches(co, g));
}

}

```

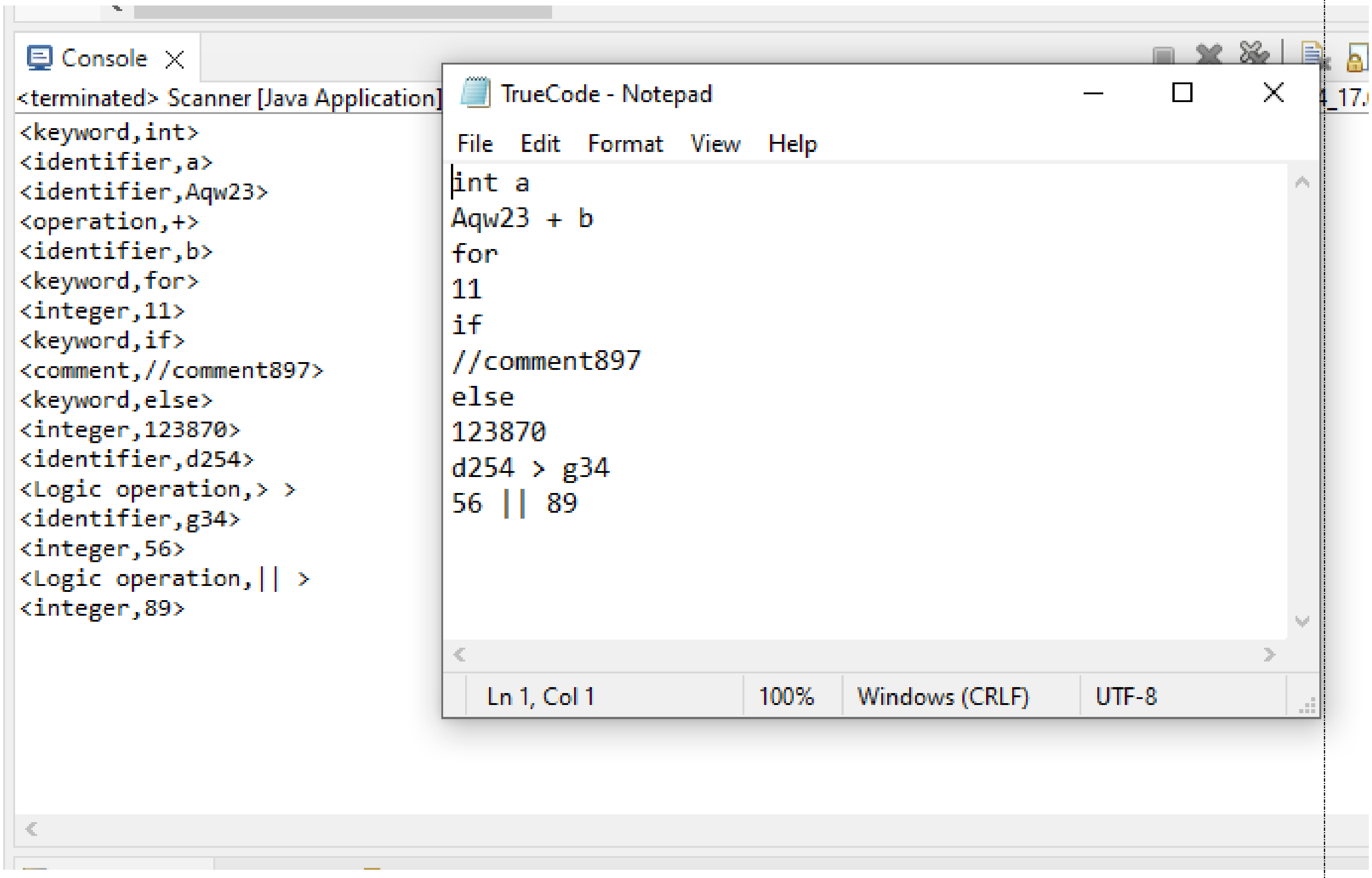


```
}
//operation
static boolean isoperation(String g)
{
String[] operationArray= {"+","-","/","*"} ;
if(operationArray[0].equals(g) || operationArray[1].equals(g)
        || operationArray[2].equals(g) ||operationArray[3].equals(g)) {
    return true;}
else
    {return false;}

}

//logical operation
static boolean isLogicalOp(String g)
{
    String[] LogOpArray= {"||","&&","==","!=",">","<"};
    if(LogOpArray[0].equals(g) || LogOpArray[1].equals(g) ||LogOpArray[2].equals(g)
            ||LogOpArray[3].equals(g)||LogOpArray[4].equals(g)||LogOpArray[5].equals(g))
    {
        return true;}
    else
        {return false;}
}
```



```
//keyword
static boolean isKeyword(String g) {
    String[] keywords= {"for","do","if","while","else","int"};
    if(keywords[0].equals(g) || keywords[1].equals(g) ||keywords[2].equals(g)
            ||keywords[3].equals(g)||keywords[4].equals(g)||keywords[5].equals(g))
    {
        return true;}
    else
        {return false;}
}
```

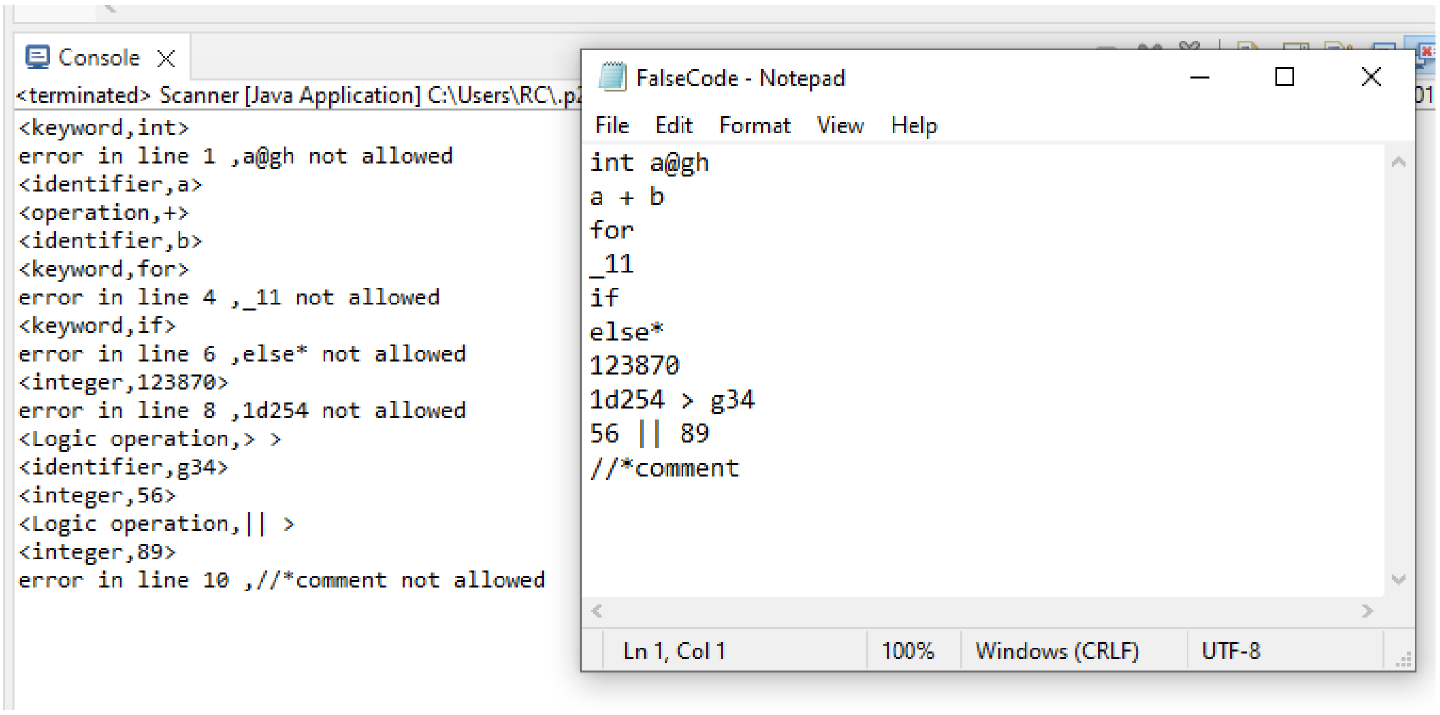
- Testing True code



```
<terminated> Scanner [Java Application]
<keyword,int>
<identifier,a>
<identifier,Aqw23>
<operation,+>
<identifier,b>
<keyword,for>
<integer,11>
<keyword,if>
<comment,//comment897>
<keyword,else>
<integer,123870>
<identifier,d254>
<Logic operation,> >
<identifier,g34>
<integer,56>
<Logic operation,|| >
<integer,89>
```

```
int a
Aqw23 + b
for
11
if
//comment897
else
123870
d254 > g34
56 || 89
```

- False code



```
<terminated> Scanner [Java Application] C:\Users\RC\.p
<keyword,int>
error in line 1 ,a@gh not allowed
<identifier,a>
<operation,+>
<identifier,b>
<keyword,for>
error in line 4 ,_11 not allowed
<keyword,if>
error in line 6 ,else* not allowed
<integer,123870>
error in line 8 ,1d254 not allowed
<Logic operation,> >
<identifier,g34>
<integer,56>
<Logic operation,|| >
<integer,89>
error in line 10 ,//*comment not allowed
```

```
int a@gh
a + b
for
_11
if
else*
123870
1d254 > g34
56 || 89
//*comment
```

References

https://www.techtarget.com/whatis/definition/compiler

https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380101203#:~:text=The%20scanner%20is%20a%20subroutine,into%20recognizable%20units%20called%20tokens.