Building a Scanner

Bibek Dhungana, Kiriti Aryal, Arogya Bhatta

Texas Tech University

CS3361 Concepts of Programming Languages

Dr. Yuanlin Zhan, Texas Tech University

Computer Science,

March 27 , 2022

Introduction

The primary task of this project is to build a scanner and implement it using any programming language. Scanner (also known as tokenization) is the first step of compilation where the input source file is taken as input and the valid token in the programming language is identified. This process should throw an error if invalid tokens are identified. We need to be varied and clear about the input and output of specific tasks.

For the Scanner:

Input: the file written in specified programming language(We use input.txt for this demo).

Output: tokens in a given programming language.

(Note: The valid token in one programming language might be invalid in another programming language, this process of scanning is compiler dependent.)

There were multiple choices for us to build a scanner. We chose java in this assignment because java has very rich features in string handling issues and a lot of optimization is done by java jit compiler leading to efficient code.

Data Structure Used in this program:

The primary data structure used in these projects is arrays. We did not use sophisticated features from the java collection api. Rather we build our scanner using single dimensional and two dimensional arrays as our data structure. In java, arrays are also treated as objects, so this was one of the reasons for choosing arrays as data structure.

Apart from this, we have also used a predefined class in java File class to read the context of the input file that needs to be scanned.

Testing the Scanner

The first step of this process is compilation.Since, all the Scanner related properties and methods are located in Scanner class. Those methods are tested in a file ScannerTest.

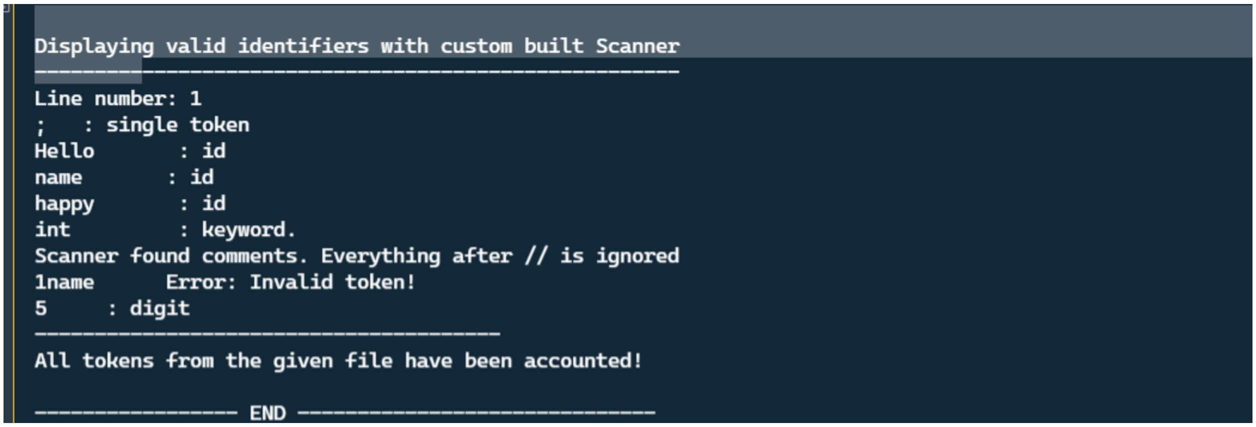
Compilation:

Javac *.java   //compiling all the java files in current working directory

Execution:

Java ScannerTest.java   //Executing the .class file using java virtual machine.

Below is screen shot of the working scanner code written in java programing language:



```
Displaying valid identifiers with custom built Scanner
------------------------------------------------------
Line number: 1
;   : single token
Hello       : id
name       : id
happy       : id
int         : keyword.
Scanner found comments. Everything after // is ignored
1name      Error: Invalid token!
5     : digit
---------------------------------------
All tokens from the given file have been accounted!

----------------- END ------------------------------
```

Pseudocode:

Contents:

```
// this function finds valid single token
    findValidToken(data);

    // this function finds identifiers
    findId(data);

    // this function finds reserved keyword
    findKeyword(data);

    // this function checks for the assignment
    findAssignment(data);

    // this function checks if there are any comments
    findComment(data);

    // this checks for floating point numbers
    findFloatingPointNumber(data);

    // this function checks for digits
    findDigit(data);
```

Main Function:

```
while( !End of line){

Object1.Scanner(data);          //creating scanner class and constructor is invoked.

 }
```

For reading the file.

Using while loop:

```
while (getfileData= getfile.readline() != NULL){

            Code to access each word of the text file

}
```

Reading and comparing each word from the text file:

```
ReadAndCompare{

tokens(data);

            decimal(data);

            id(data);

            keyword(data);

 Assign(data);
```

```
         comment(data);
        Digit(data);
 }
```

Assign Function:

```
nexchar: next to prechar
for(i=0 ; i= data.length){
                For(j=0 ; j= data[i].length){
                        If(prechar = data[i].charAt(j) = “:” and nexchar = “=” )
                                Print assign(:=);
```

Checking Digits

```
D={0,1,2,3,4,5,6,7,8,9}
        while(  i <= data.length)

       If(prechar = data[i].charAt(0)  = D)
                If(prechar=D && nexchar is  D until word ends)
         Println( this is a Digit)
```

Checking for decimal function:

```
while(i <= data.length)
  If(prechar = Digit)
        If (nexchar = Digit) move to next
         Else If(nexchar = ‘.’ Move to next )
Println(this is a decimal number)
```

Function to check keyword:

```
letter={a,b,c,d,….,A,B,C,D,….}
 while( i <= data.length) {
     do{
             If(data[i] =  string/ int/ double ……)
} while(prechar != letter) }
      Println( this is a keyword)
```

Function to check id:

```
while( i <= data.length)
                 do{
             If(data[i] = keyword, then data[i] is a keyword
                          Else data is an ID } while(prechar != letter)
```

Checking tokens:

Tokens function:

```
   Singletoken={*,/,+,….}
  For( i=0 to data.length){
            For(j=0 to data[i].length){
                    If(prechar = data[i].charAt(j) = singletoken)
                            Print(this is a token);
```

Conclusion

Thus, the scanner is built in the Java programming language. This scanner uses various concerts used in the class like regular expressions to determine if the identifiers are valid.It also uses various other computing constructs like abstraction, decomposition and data manipulation.