



Tecnológico Nacional de México.
Instituto tecnológico de Nuevo
León.



Ingeniería en Sistemas Computacionales

Lenguajes autómatas II

“Reporte Proyecto 2”

Reporte presentado por:

García Hernández Julio César.

#15480089

Guadalupe, Nuevo León a 16 de Marzo del 2018,

INTRODUCCIÓN

Este segundo proyecto se basaba en 3 partes, la primera constaba en leer un archivo .txt e ir separando cada una de las palabras, al igual que los signos de puntuación, números etc. Todo esto posteriormente se almacenaba en un arreglo para así poder compararse por medio de la clase tokenizer cada una de estas palabras con una tabla de símbolos y saber si es alguna palabra reservada, algún número, operador o signo de puntuación, posteriormente si no existe en esta tabla se iría agregando.

DESCRIPCIÓN

El programa se dividía en diferentes partes, entre clases y métodos los cuales eran:

Método booleano Numero: en este método se verifica si en el txt se encontraba algún numero

Método booleano Operador: verificaba si se encontraba algún operador

Método booleano Puntuación: buscaba signos de puntuación al recorrer la tabla

Método leer: es el encargado de leer el archivo txt y separar cada elemento por medio de tokens para posteriormente guardarlos en un arreglo.

Clase Main: esta creaba la tabla de operadores buscando palabras reservadas, signos de puntuación, operadores y números, si no se encontraban en la tabla estos se iban agregando.

```
class Tokenizer {  
    Método booleano isNumero (String numero) {  
        Variable double num  
        intenta {  
            num = cambiarlo a double(el numero enviado por parámetro);  
        }  
        En caso de error(imprime el error){  
            regresa(false);  
        }  
        regresa(true);  
    }  
}
```

```
Método booleano isOperador(String operador, tabla de operadores[]){  
Ciclo para recorrer la tabla{  
Si(tabla operadores es igual a operador)){  
regresa true;  
}}  
Si no, regresa false;  
}
```

```
Método booleano isPuntuacion(String punt,tabla de signos de  
puntuación){  
Ciclo para recorrer la tabla{  
Si(tabla de signos de puntuación es igual a punt){  
regresa true;  
}}  
Si no, regresa false;  
}
```

```
Método leer(String ruta){  
intenta{  
leer el archivo de la ruta;  
String linea;  
String token;  
Nuevo arreglo llamado ar;  
Mientras existan líneas en el archivo{  
Separador(línea. Que es donde leerá el separador, los símbolos  
que separara: ";./,(/)/[/]:" , devuelve los símbolos separados);  
Mientras existan más tokens{  
Agrega el token al arreglo ar;  
}}  
En caso de error(imprime el error){  
}}}
```

```
Método principal{
```

```

Tabla operadores = {"+", "-", "*", "/"};
Tabla palabrasReservadas =
{"if", "while", "public", "for", "private", "main", "int", "float",
"double", "String"};
Tabla signosDePuntuacion = {";", ".", ",", "(", ")", "[", ""]};
Tabla Identificadores = {"01", "operador"},
{"02", "p_reservada"},
{"03", "numero"},
{"04", "espacio en blanco"},
{"05", "identificador"},
{"06", "puntuacion"}};
leer("C:\\Prueba\\archivo.txt");
isPuntuacion("", signosDePuntuacion);
isOperador("", operadores);
isNumero("");
}
}

```

CONCLUSIONES:

En este proyecto conocí la clase tokenizer, al igual que hice uso de algunos métodos que aprendí en Programación Orientada a Objetos como lo es el leer archivos txt, lamentablemente el programa no está del todo completo, me faltó implementar la parte de los cuádruplos. Son cosas que por una u otra razón son nuevas para mí.