



INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ingeniería en sistemas computacionales

Lenguaje y Autómatas 2

Proyecto 2

Catedrático

Juan Pablo Rosas Baldazo

Alumno

Rafael Salazar Rodríguez

16/03/2018

Introducción

Este proyecto consiste básicamente en abrir y leer un archivo y de ese modo utilizando tokens dividir el contenido del mismo por signos especiales que declararemos en el guardar cada una de ellas dentro de un array para comparar con una tabla y determinar a qué tipo de símbolo pertenece. Finalmente se crean cuádruplas a partir de cadenas las cuales deben en si representar una operación de este tipo.

Pseudocódigo

Clase Tokenizer.java

```
clase 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){
        }
    }
}

cuadruplor (String not_pref){
    int i=0;
    char item1, item2, operador;

```

```

String[] operando = {"+", "-", "*", "/"};
while (i<= not_pref.length()) {
    item1 = not_pref.charAt(i);
    for(int j=0;j<4;j++){
        if(operando[j].equals(item1)){
            operador = item1;
            item2 = not_pref.charAt(i+1);
        }
    }
}

```

```

if(item1==operando){
    item2 = not_pref[i+i];
    if(item2==operando){
        if(operador !=null)
            {Agregar cuádruplo}
    }
    else{operador=item[i]}
}
else{operador = item[i]}
i++;
}
}

```

```

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("");

}

}

```

Conclusión

En esta unidad conocimos las funciones y métodos que nos permitían abrir, leer archivos utilizando la ruta en la cual están almacenados, separar el contenido de estos a través de signos y comparando si estos pertenecen a un cierto grupo de símbolos con arreglos. Si bien el programa no funciona del todo bien se logró aprender conocimientos que no tenía.

Bibliografía

<https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html>