

Informe Analizador Sintactico

Mauro Montano LU:108882

Modificaciones

Se realizaron las modificaciones indicadas en la corrección de la etapa del Analizador Lexico y se creó el analizador sintáctico que va matcheando con cada token. Luego, el resultado es solo mostrado por pantalla.

Compilar y ejecutar

El programa fue desarrollado en Eclipse y para realizar una compilación desde línea de comandos, es necesario situarse en la carpeta donde tenemos el exportable .jar y ejecutar el siguiente comando:

```
java -jar Etapa2.jar <Entrada>
```

Donde Entrada es un parámetro obligatorio, con la ruta del archivo fuente a ser evaluado en el análisis léxico incluyendo el formato de archivo. El resultado se mostrará por pantalla.

Clases nuevas

ExcepcionSintactica: Esta es la clase que extiende a la clase Excepcion, y es llamada por el analizador sintáctico para que pueda devolver los errores correspondientes, mostrando la fila donde se produce el error.

AnalizadorSintactico: En esta clase le pedimos los tokens al analizador léxico hasta que se llegue al token EOF. En el constructor se pide el primer token al analizador léxico y se lo envía al método inicial, que se encarga de llamar los métodos correspondientes y verificar el token recibido para verificar si existe un error de sintaxis. Si no existen errores de sintaxis ni léxicos entonces se muestra por pantalla que el programa es sintácticamente correcto.

Los métodos que aparecen determinan si cumplen la gramática establecida y de esta manera determinamos si existen errores de sintaxis.

También se agrega el método match, donde en el primer parámetro nos sirve para comparar con el token, y el segundo para mostrar el error correspondiente en caso de que no se dé el matcheo.

Se agrega el método contenido, que determina si el token actual corresponde a algunos de los tokens pasados por parámetro a este método y finalmente otros métodos más correspondientes al nombre de las gramáticas para resolver los tokens.

Gramática

Gramática sin extensión BNF

1. $\langle \text{Inicial} \rangle \rightarrow \langle \text{ListaClases} \rangle$
2. $\langle \text{ListaClases} \rangle \rightarrow \langle \text{Clase} \rangle \langle \text{ListaClases} \rangle \mid \langle \text{Clase} \rangle$
3. $\langle \text{Clase} \rangle \rightarrow \text{class } \text{idClase} \langle \text{Herencia} \rangle \{ \langle \text{ListaMiembros} \rangle \}$
4. $\langle \text{Herencia} \rangle \rightarrow \text{extends } \text{idClase} \mid \epsilon$
5. $\langle \text{ListaMiembros} \rangle \rightarrow \langle \text{Miembro} \rangle \mid \langle \text{ListaMiembros} \rangle \mid \epsilon$
6. $\langle \text{Miembro} \rangle \rightarrow \langle \text{Atributo} \rangle \mid \langle \text{Constructor} \rangle \mid \langle \text{Metodo} \rangle$
7. $\langle \text{Atributo} \rangle \rightarrow \langle \text{Visibilidad} \rangle \langle \text{Tipo} \rangle \langle \text{ListaDecAtrs} \rangle ;$
8. $\langle \text{Metodo} \rangle \rightarrow \langle \text{FormaMetodo} \rangle \langle \text{TipoMetodo} \rangle \text{idMetVar} \langle \text{ArgsFormales} \rangle \langle \text{Bloque} \rangle$
9. $\langle \text{Constructor} \rangle \rightarrow \text{idClase} \langle \text{ArgsFormales} \rangle \langle \text{Bloque} \rangle$
10. $\langle \text{Visibilidad} \rangle \rightarrow \text{public} \mid \text{private}$
11. $\langle \text{Tipo} \rangle \rightarrow \langle \text{TipoPrimitivo} \rangle \mid \text{idClase}$
12. $\langle \text{TipoPrimitivo} \rangle \rightarrow \text{boolean} \mid \text{char} \mid \text{int} \mid \text{String}$
13. $\langle \text{ListaDecAtrs} \rangle \rightarrow \text{idMetVar} \mid \text{idMetVar} , \langle \text{ListaDecAtrs} \rangle$
14. $\langle \text{FormaMetodo} \rangle \rightarrow \text{static} \mid \text{dynamic}$
15. $\langle \text{TipoMetodo} \rangle \rightarrow \langle \text{Tipo} \rangle \mid \text{void}$
16. $\langle \text{ArgsFormales} \rangle \rightarrow \{ \langle \text{ListaArgsFormalesOVacio} \rangle \}$
17. $\langle \text{ListaArgsFormalesOVacio} \rangle \rightarrow \langle \text{ListaArgsFormales} \rangle \mid \epsilon$
18. $\langle \text{ListaArgsFormales} \rangle \rightarrow \langle \text{ArgFormal} \rangle \mid \langle \text{ArgFormal} \rangle , \langle \text{ListaArgsFormales} \rangle$
19. $\langle \text{ArgFormal} \rangle \rightarrow \langle \text{Tipo} \rangle \text{idMetVar}$
20. $\langle \text{Bloque} \rangle \rightarrow \{ \langle \text{ListaSentencias} \rangle \}$
21. $\langle \text{ListaSentencias} \rangle \rightarrow \langle \text{Sentencia} \rangle \langle \text{ListaSentencias} \rangle \mid \epsilon$
22. $\langle \text{Sentencia} \rangle \rightarrow ; \mid \langle \text{Asignacion} \rangle ; \mid \langle \text{Llamada} \rangle ; \mid \langle \text{Tipo} \rangle \langle \text{ListaDecVars} \rangle ; \mid \text{if} (\langle \text{Expresion} \rangle) \langle \text{Sentencia} \rangle \mid \text{if} (\langle \text{Expresion} \rangle) \langle \text{Sentencia} \rangle \text{else} \langle \text{Sentencia} \rangle \mid \text{while} (\langle \text{Expresion} \rangle) \langle \text{Sentencia} \rangle \mid \langle \text{Bloque} \rangle \mid \text{return} \langle \text{ExpresionOVacio} \rangle ;$
23. $\langle \text{Asignacion} \rangle \rightarrow \langle \text{Acceso} \rangle \langle \text{TipoDeAsignacion} \rangle \langle \text{Expresion} \rangle$
24. $\langle \text{TipoDeAsignacion} \rangle \rightarrow = \mid += \mid -=$
25. $\langle \text{Llamada} \rangle \rightarrow \langle \text{Acceso} \rangle$
26. $\langle \text{ExpresionOVacio} \rangle \rightarrow \langle \text{Expresion} \rangle \mid \epsilon$
27. $\langle \text{Expresion} \rangle \rightarrow \langle \text{Expresion} \rangle \langle \text{OperadorBinario} \rangle \langle \text{ExpresionUnaria} \rangle \mid \langle \text{ExpresionUnaria} \rangle$
28. $\langle \text{OperadorBinario} \rangle \rightarrow \mid \mid \&\& \mid = \mid != \mid < \mid > \mid <= \mid >= \mid + \mid - \mid * \mid / \mid \%$
29. $\langle \text{ExpresionUnaria} \rangle \rightarrow \langle \text{OperadorUnario} \rangle \langle \text{Operando} \rangle \mid \langle \text{Operando} \rangle$
30. $\langle \text{OperadorUnario} \rangle \rightarrow + \mid - \mid !$
31. $\langle \text{Operando} \rangle \rightarrow \langle \text{Literal} \rangle \mid \langle \text{Acceso} \rangle$
32. $\langle \text{Literal} \rangle \rightarrow \text{null} \mid \text{true} \mid \text{false} \mid \text{intLiteral} \mid \text{charLiteral} \mid \text{stringLiteral}$
33. $\langle \text{Acceso} \rangle \rightarrow \langle \text{Primario} \rangle \langle \text{Encadenado} \rangle$

34. $\langle \text{Primario} \rangle \rightarrow \langle \text{AccesoThis} \rangle \mid \langle \text{AccesoVar} \rangle \mid \langle \text{AccesoEstatico} \rangle \mid \langle \text{AccesoConstructor} \rangle \mid \langle \text{AccesoMetodo} \rangle \mid (\langle \text{Expresion} \rangle)$
35. $\langle \text{AccesoThis} \rangle \rightarrow \text{this}$
36. $\langle \text{AccesoVar} \rangle \rightarrow \text{idMetVar}$
37. $\langle \text{AccesoEstatico} \rangle \rightarrow \text{static idClase} . \langle \text{AccesoMetodo} \rangle$
38. $\langle \text{AccesoConstructor} \rangle \rightarrow \text{new idClase} \langle \text{ArgsActuales} \rangle$
39. $\langle \text{AccesoMetodo} \rangle \rightarrow \text{idMetVar} \langle \text{ArgsActuales} \rangle$
40. $\langle \text{ArgsActuales} \rangle \rightarrow (\langle \text{ListaExpsOVacio} \rangle)$
41. $\langle \text{ListaExpsOVacio} \rangle \rightarrow \langle \text{ListaExps} \rangle \mid \epsilon$
42. $\langle \text{ListaExps} \rangle \rightarrow \langle \text{Expresion} \rangle \mid \langle \text{Expresion} \rangle, \langle \text{ListaExps} \rangle$
43. $\langle \text{Encadenado} \rangle \rightarrow \langle \text{VarOMetodoEncadenado} \rangle \langle \text{Encadenado} \rangle \mid \epsilon$
44. $\langle \text{VarOMetodoEncadenado} \rangle \rightarrow \langle \text{VarEncadenada} \rangle \mid \langle \text{MetodoEncadenado} \rangle$
45. $\langle \text{VarEncadenada} \rangle \rightarrow . \text{idMetVar}$
46. $\langle \text{MetodoEncadenado} \rangle \rightarrow . \text{idMetVar} \langle \text{ArgsActuales} \rangle$

Gramática final

Gramática final del resultado de aplicar recursión y factorización a izquierda

1. $\langle \text{Inicial} \rangle \rightarrow \langle \text{ListaClases} \rangle$
2. $\langle \text{ListaClases} \rangle \rightarrow \langle \text{Clase} \rangle \langle \text{ListaClasesF} \rangle \mid \langle \text{Clase} \rangle$
3. $\langle \text{ListaClasesF} \rangle \rightarrow \langle \text{ListaClases} \rangle \mid \epsilon$
4. $\langle \text{Clase} \rangle \rightarrow \text{class idClase} \langle \text{Herencia} \rangle \{ \langle \text{ListaMiembros} \rangle \}$
5. $\langle \text{Herencia} \rangle \rightarrow \text{extends idClase} \mid \epsilon$
6. $\langle \text{ListaMiembros} \rangle \rightarrow \langle \text{Miembro} \rangle \mid \langle \text{ListaMiembros} \rangle \mid \epsilon$
7. $\langle \text{Miembro} \rangle \rightarrow \langle \text{Atributo} \rangle \mid \langle \text{Constructor} \rangle \mid \langle \text{Metodo} \rangle$
8. $\langle \text{Atributo} \rangle \rightarrow \langle \text{Visibilidad} \rangle \langle \text{Tipo} \rangle \langle \text{ListaDecAtrs} \rangle ;$
9. $\langle \text{Metodo} \rangle \rightarrow \langle \text{FormaMetodo} \rangle \langle \text{MetodoF} \rangle$
10. $\langle \text{MetodoF} \rangle \rightarrow \langle \text{TipoMetodo} \rangle \text{idMetVar} \langle \text{ArgsFormales} \rangle \langle \text{Bloque} \rangle$
11. $\langle \text{Constructor} \rangle \rightarrow \text{idClase} \langle \text{ArgsFormales} \rangle \langle \text{Bloque} \rangle$
12. $\langle \text{Visibilidad} \rangle \rightarrow \text{public} \mid \text{private}$
13. $\langle \text{Tipo} \rangle \rightarrow \langle \text{TipoPrimitivo} \rangle \mid \text{idClase}$
14. $\langle \text{TipoPrimitivo} \rangle \rightarrow \text{boolean} \mid \text{char} \mid \text{int} \mid \text{String}$
15. $\langle \text{ListaDecAtrs} \rangle \rightarrow \text{idMetVar} \langle \text{ListaDecAtrsF} \rangle$
16. $\langle \text{ListaDecAtrsF} \rangle \rightarrow , \langle \text{ListaDecAtrsF} \rangle \mid \epsilon$
17. $\langle \text{FormaMetodo} \rangle \rightarrow \text{static} \mid \text{dynamic}$
18. $\langle \text{TipoMetodo} \rangle \rightarrow \langle \text{Tipo} \rangle \mid \text{void}$
19. $\langle \text{ArgsFormales} \rangle \rightarrow (\langle \text{ListaArgsFormalesOVacio} \rangle)$
20. $\langle \text{ListaArgsFormalesOVacio} \rangle \rightarrow \langle \text{ListaArgsFormales} \rangle \mid \epsilon$
21. $\langle \text{ListaArgsFormales} \rangle \rightarrow \langle \text{ArgFormal} \rangle \langle \text{ListaArgsFormalesAux} \rangle$
22. $\langle \text{ListaArgsFormalesAux} \rangle \rightarrow \langle \text{ListaArgsFormales} \rangle \mid \epsilon$
23. $\langle \text{ArgFormal} \rangle \rightarrow \langle \text{Tipo} \rangle \text{idMetVar}$
24. $\langle \text{Bloque} \rangle \rightarrow \{ \langle \text{ListaSentencias} \rangle \}$
25. $\langle \text{ListaSentencias} \rangle \rightarrow \langle \text{Sentencia} \rangle \langle \text{ListaSentencias} \rangle \mid \epsilon$

26. $\langle \text{Sentencia} \rangle \rightarrow ; \mid \langle \text{Asignacion} \rangle ; \mid \langle \text{Llamada} \rangle ; \mid \langle \text{Tipo} \rangle \langle \text{ListaDecVars} \rangle ; \mid \text{if} (\langle \text{Expresion} \rangle) \langle \text{Sentencia} \rangle \langle \text{SentenciaFif} \rangle \mid \text{while} (\langle \text{Expresion} \rangle) \langle \text{Sentencia} \rangle \mid \langle \text{Bloque} \rangle \mid \text{return} \langle \text{SentenciaFRet} \rangle$
27. $\langle \text{SentenciaFif} \rangle \rightarrow \epsilon \mid \text{else} \langle \text{Sentencia} \rangle$
28. $\langle \text{SentenciaFRet} \rangle \rightarrow \langle \text{ExpresionOVacio} \rangle ; \mid ;$
29. $\langle \text{Asignacion} \rangle \rightarrow \langle \text{Acceso} \rangle \langle \text{TipoDeAsignacion} \rangle \langle \text{Expresion} \rangle$
30. $\langle \text{TipoDeAsignacion} \rangle \rightarrow = \mid += \mid -=$
31. $\langle \text{Llamada} \rangle \rightarrow \langle \text{Acceso} \rangle$
32. $\langle \text{ExpresionOVacio} \rangle \rightarrow \langle \text{Expresion} \rangle \mid \epsilon$
33. $\langle \text{Expresion} \rangle \rightarrow \langle \text{ExpresionUnaria} \rangle \langle \text{ExpresionAux} \rangle$
34. $\langle \text{ExpresionAux} \rangle \rightarrow \langle \text{OperadorBinario} \rangle \langle \text{ExpresionUnaria} \rangle \langle \text{ExpresionAux} \rangle \mid \epsilon$
35. $\langle \text{OperadorBinario} \rangle \rightarrow \mid \mid \mid \&\& \mid == \mid != \mid < \mid > \mid <= \mid >= \mid + \mid - \mid * \mid / \mid \%$
36. $\langle \text{ExpresionUnaria} \rangle \rightarrow \langle \text{ExpresionUnariaAux} \rangle \langle \text{Operando} \rangle$
37. $\langle \text{ExpresionUnariaAux} \rangle \rightarrow \langle \text{OperadorUnario} \rangle \mid \epsilon$
38. $\langle \text{OperadorUnario} \rangle \rightarrow + \mid - \mid !$
39. $\langle \text{Operando} \rangle \rightarrow \langle \text{Literal} \rangle \mid \langle \text{Acceso} \rangle$
40. $\langle \text{Literal} \rangle \rightarrow \text{null} \mid \text{true} \mid \text{false} \mid \text{intLiteral} \mid \text{charLiteral} \mid \text{stringLiteral}$
41. $\langle \text{Acceso} \rangle \rightarrow \langle \text{Primario} \rangle \langle \text{Encadenado} \rangle$
42. $\langle \text{Primario} \rangle \rightarrow \langle \text{AccesoThis} \rangle \mid \langle \text{AccesoVarOMet} \rangle \mid \langle \text{AccesoEstatico} \rangle \mid \langle \text{AccesoConstructor} \rangle \mid (\langle \text{Expresion} \rangle)$
43. $\langle \text{AccesoThis} \rangle \rightarrow \text{this}$
44. $\langle \text{AccesoVarOMet} \rangle \rightarrow \text{idMetVar} \langle \text{ArgsActuales} \rangle$
45. $\langle \text{AccesoEstatico} \rangle \rightarrow \text{static idClase} . \langle \text{AccesoMetodo} \rangle$
46. $\langle \text{AccesoConstructor} \rangle \rightarrow \text{new idClase} \langle \text{ArgsActuales} \rangle$
47. $\langle \text{ArgsActuales} \rangle \rightarrow (\langle \text{ListaExpsOVacio} \rangle)$
48. $\langle \text{ListaExpsOVacio} \rangle \rightarrow \langle \text{ListaExps} \rangle \mid \epsilon$
49. $\langle \text{ListaExps} \rangle \rightarrow \langle \text{Expresion} \rangle \langle \text{ListaExpsF} \rangle$
50. $\langle \text{ListaExpsF} \rangle \rightarrow \langle \text{ListaExpsF} \rangle \mid \epsilon$
51. $\langle \text{Encadenado} \rangle \rightarrow \langle \text{VarOMetodoEncadenado} \rangle \mid \epsilon$
52. $\langle \text{VarOMetodoEncadenado} \rangle \rightarrow . \text{idMetVar} \langle \text{VarOMetodoEncadenadoF} \rangle \langle \text{Encadenado} \rangle$
53. $\langle \text{VarOMetodoEncadenadoF} \rangle \rightarrow \langle \text{ArgsActuales} \rangle$

Casos de prueba

Test validos

- **Constructor:** Es un constructor dentro de una clase.
- **Declaraciones:** Se comprueba la declaración de variables y asignaciones.
- **Llamadas:** Se comprueba que las llamadas y el anidamiento de llamadas funcionan correctamente.
- **Argumentos:** Se comprueba que la detección de argumentos es válida, además de la herencia.
- **MetAtribConst:** Se encuentran métodos, atributos y constructores dentro de una clase. Para comprobar su correcta detección.

- **IfElse:** Se encuentran if y else anidados, y argumentos complejos en los condicionales y algunas sentencias dentro de los bloques contenidos en los condicionales.
- **Whiles:** Se comprueba el correcto funcionamiento de whiles anidados.
- **ClaseCompleta:** Es una clase donde se comprueban que funcione la herencia, constructor, métodos, this, new, y encadenados.
- **ProgramaPrueba:** Es un programa de prueba sintácticamente válido con clases dentro, donde se comprueban la mayoría de las gramáticas.

Test Inválidos

- **ArchivoVacío:** Se encuentra un archivo vacío.
- **ArgumentosSinComa:** La separación de argumentos no es con coma.
- **ArgumentosConPuntoYComa:** La separación de argumentos es con punto y coma.
- **ArgumentosDobleComa:** Se encuentra una doble coma al final de los argumentos.
- **Atributos:** Se define un atributo sin identificador.
- **Clase:** La clase comienza con clas faltándole una “s” mas.
- **ErrorLexico:** Esta es una clase que contiene un solo | en la comparación dentro del if, por lo que es un error Léxico y muestra correctamente el error.
- **If:** Un if sin paréntesis que cierra.
- **Else:** Se encuentra un condicional con un else de más.
- **Expresiones:** En una asignación aparece una expresión donde unos de los números tiene un operador unario no válido, en este caso un *.
- **Extends:** El extends extiende a un identificador y no una clase.
- **Metodo:** Se define un método que tiene el nombre de una clase.
- **Miembro:** El miembro no es un atributo, ni un constructor, ni un método.
- **While:** Se define un while sin condición.
- **SinModificador:** Se declara un atributo sin el modificador de acceso.
- **ProgramaPrueba:** Es un programa de prueba sintácticamente inválido donde se prueban la mayoría todas las gramáticas, pero se encuentra con un modificador de visibilidad en un lugar incorrecto de la clase.