Técnicas de Compilación

Profesor: Eschoyez Maximiliano
Alumnos: Cantarelli Sofía, Figueroa Javier
Tema:Trabajo Práctico Nro. 2
Fecha de Entrega: 10/06/2020

Indice

Consigna

Dado un archivo de entrada en lenguaje C, se debe generar como salida el Árbol Sintáctico (ANTLR) correcto y mostrar por consola o archivo el contenido de la Tabla de Símbolos de cada contexto. Esto es válido únicamente cuando el archivo de entrada es correcto.

Para esta versión se deberá realizar un control de errores básicos y generar un reporte de lo encontrado para poder analizar archivos con posibles errores de codificación. El reporte de errores podrá realizarse por consola o archivo de texto. Los errores a detectar deben ser los siguientes:

Errores sintácticos comunes:

- Falta de un punto y coma,
- Falta de apertura de paréntesis,
- Formato incorrecto en lista de declaración de variables

Errores semánticos comunes:

- Doble declaración del mismo identificador,
- Uso de un identificador no declarado,
- Uso de un identificador sin inicial izar,
- Identificador declarado pero no usado,
- Tipos de datos incompatibles.

Cada error reportado debe indicarse si es sintáctico o semántico.

Acrónimos

- errorPYC: Es un error que falta un “;”.
- errorPAFor: Es un error que falta un “(“ en el ciclo for.
- errorPAWhile: Es un error que falta un “(“ en el ciclo while.
- errorPAIf : Es un error que falta un “(“ en el ciclo if.
- errorPADefFuncion: Es un error en la apertura de “(“ cuando se define la función.
- errorPALlamadaFuncion: Es un error en la apertura “(“ cuando se llama la función.
- errorPCFor:Es un error que falta un “)“ en el ciclo for.
- errorPCWhile:Es un error que falta un “)“ en el ciclo while.
- errorPCIf:Es un error que falta un “)“ en el ciclo if.
- errorPCLlamadaFuncion:Es un error en la apertura “)“ cuando se llama la función.
- errorPCDefFuncion:Es un error en la apertura de “)“ cuando se define la función.

Solución

La idea principal para abordar la solución a la problemática abordada es la generación de una aplicación que posea un generador Parser, utilizando ANTLR para poder construir y recorrer árboles de sintaxis;la corrección de errores básicos, lo generamos en un reporte por consola para analizar el archivo y sus posibles errores sintácticos o semánticos. Ademas utilizamos un Listener.java para analizar y recorrer la estructura del árbol, reconoce cada entrada y componente, y muestra los tipos de errores.
Como indica la consigna, reportamos los errores de sintaxis o semantica y finaliza el programa,
La aplicación reconoce una serie de símbolos e identifica estructuras de control como if, while y for; además de asignaciones, declaraciones, funciones y llamadas a funciones,ademas agregamos nuevas funciones y/o funcionalidades para los errores;en el acrónimo explica cada una de las mismas.

Estructura principal :

reglas.g4

```
prog : instrucciones EOF;

instrucciones : instruccion instrucciones
              |
              ;

instruccion : bloque
            | declaracionDatos PYC
            | asignacionDatos PYC
            | expresion PYC
            | definicionFuncion
            | llamadaFuncion
            | cicloFor
            | cicloWhile
            | cicloIf
            | returnDatos PYC
            | PYC
            | errorPYC
            ;
```

Ademas incluimos una tabla de símbolos “TablaSimbolos.java” que , donde se asocia la ubicación o contexto,el tipo de dato, el ámbito de la variable , constante o procedimientos y se utiliza un Map para mantener los datos durante el proceso de compilacion. Un “ID.java” para saber la ubicación y los tipos de datos,si la variable fue declarada o inicializada. “Funcion.java” para saber los argumentos de cada variable y por ultimo un “Variable.java” para saber si la variable fue utilizada.

Por ultimo el archivo “MiListener.java” contiene la visualización de los errores semánticos y sintácticos que va a mostrarse por consola una vez ejecutado el codigo.c.

Códigos Utilizados

Codigo.c

1. ```
   int a();

   int main(){
      a(a);
   }
   ```

2. ```
   int main(){
      int x = 2*3+9-6;
   }
   ```

3. ```
   int main(){
      int x=2;
      int sum;
     if(x <= 3){
         sum=x+2;
      }
   }
   ```

4. ```
   int main(){

      int multi;
      for(int x=0;x<10;x++){
         multi = 2 * x;
      }

   }
   ```

5. ```
   int main(){
      int x=6;
     int resta;
      while (x !=5 )
      {
         resta = x - 3;

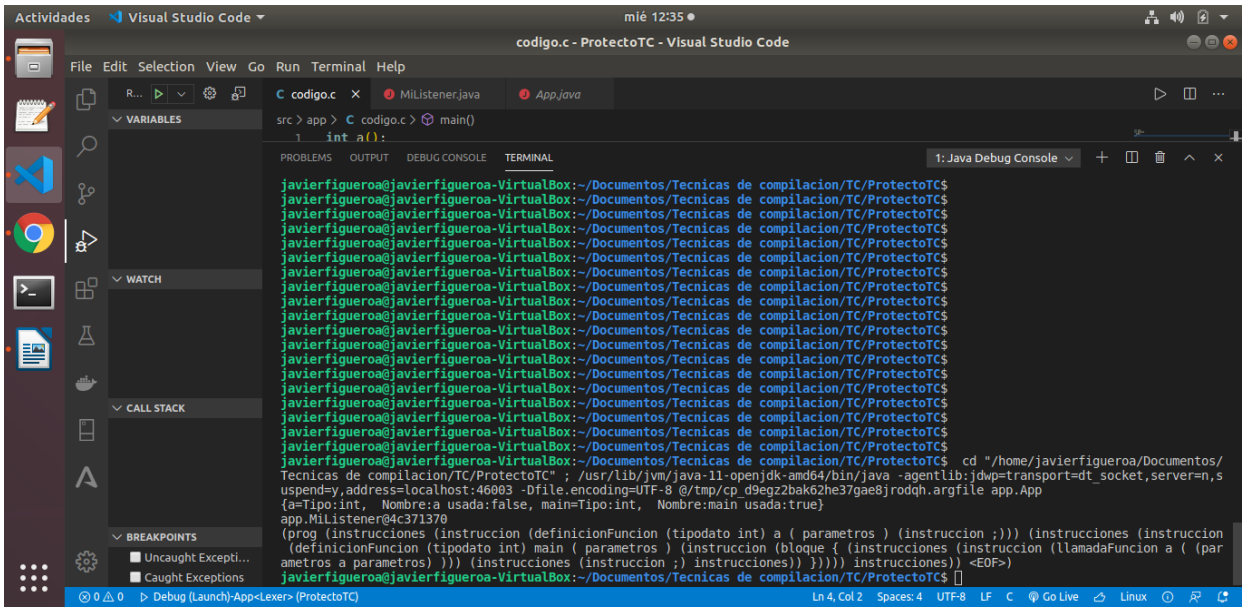
      }
   }
   ```
   ```

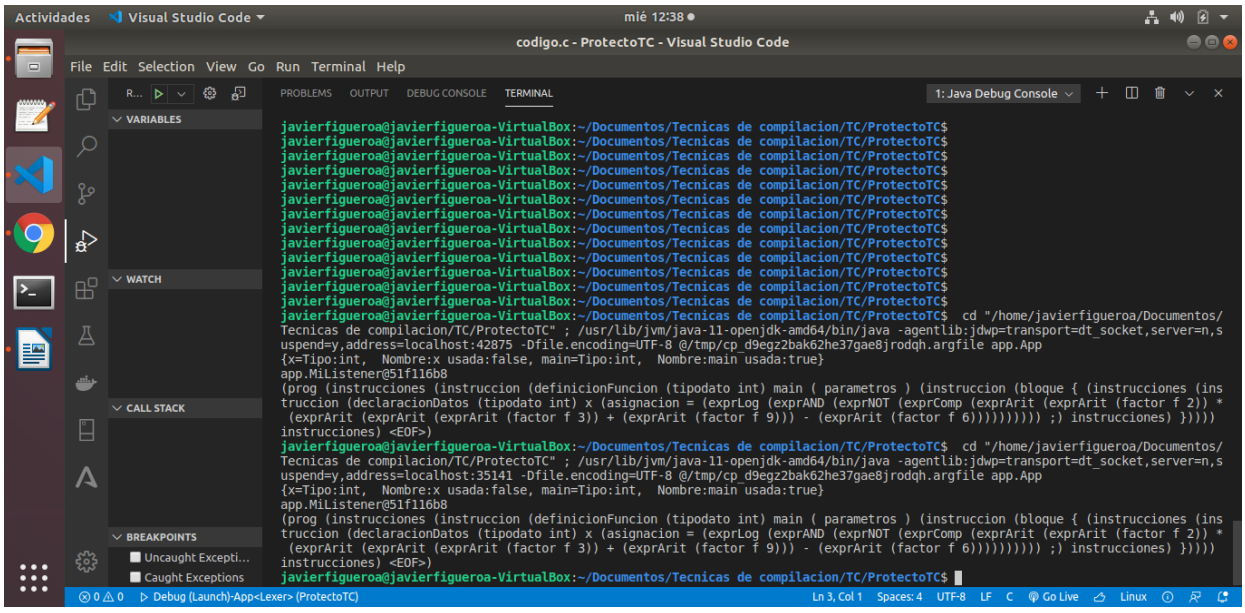
Bibliografía

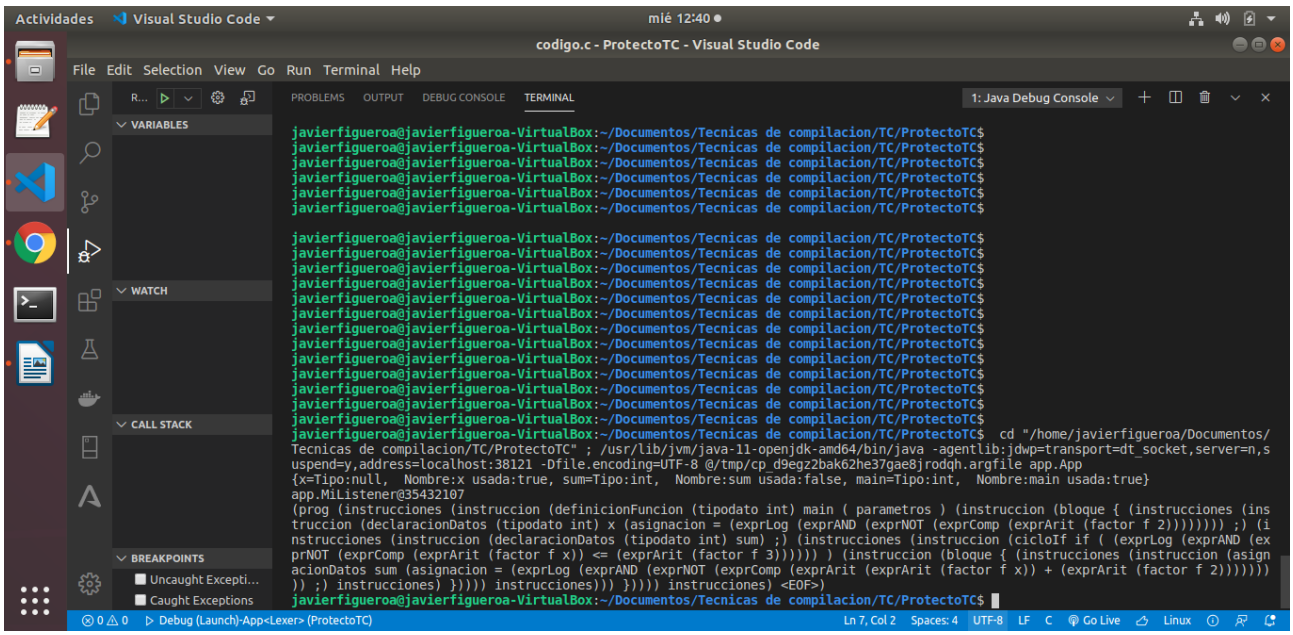
- http://lsi.vc.ehu.es/asignaturas/FdIc/labs/a1/htm/asig.html
- http://www.lcc.uma.es/~galvez/ftp/libros/Compiladores.pdf
- http://oa.upm.es/32288/1/PFC_JOSE_BARBERA_TORRALVO.pdf
- http://www.lsi.us.es/~troyano/documentos/guia.pdf
- https://books.google.com.ar/books/about/The_Definitive_ANTLR_4_Reference.html?id=SBXuLwEACAAJ&redir_esc=y
- https://campodeencinos.wordpress.com/2012/01/30/implantacion-de-la-tabla-de-simbolos-usando-el-metodo-de-hash-en-java/

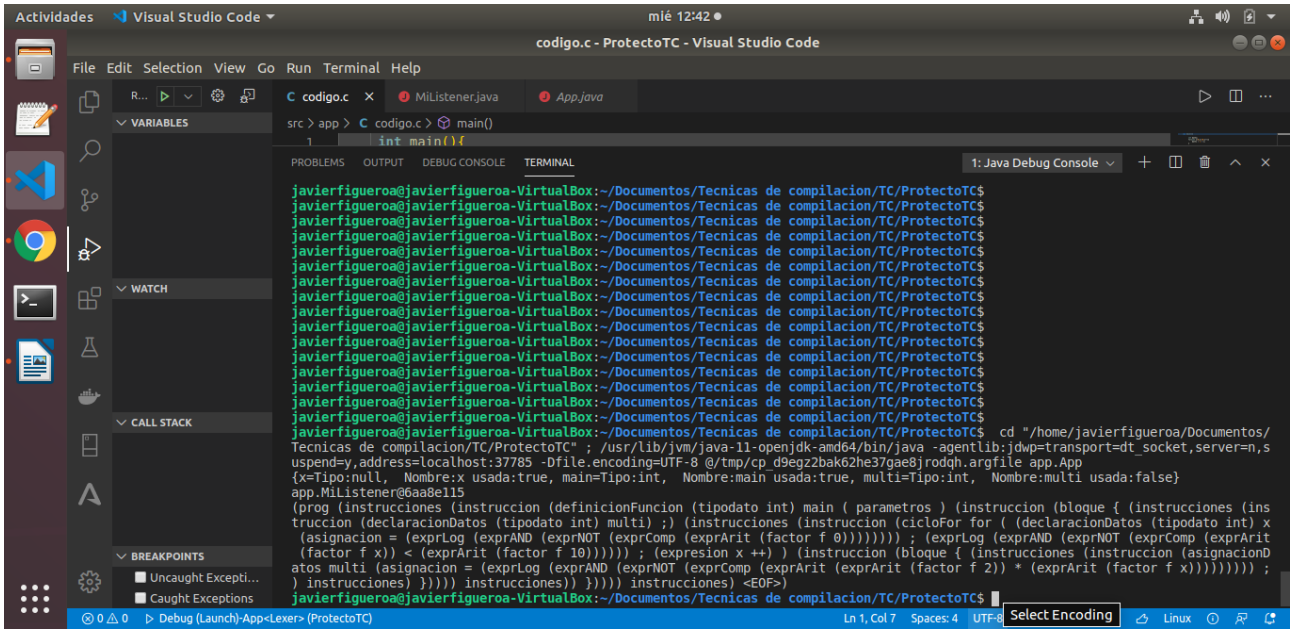
Anexo

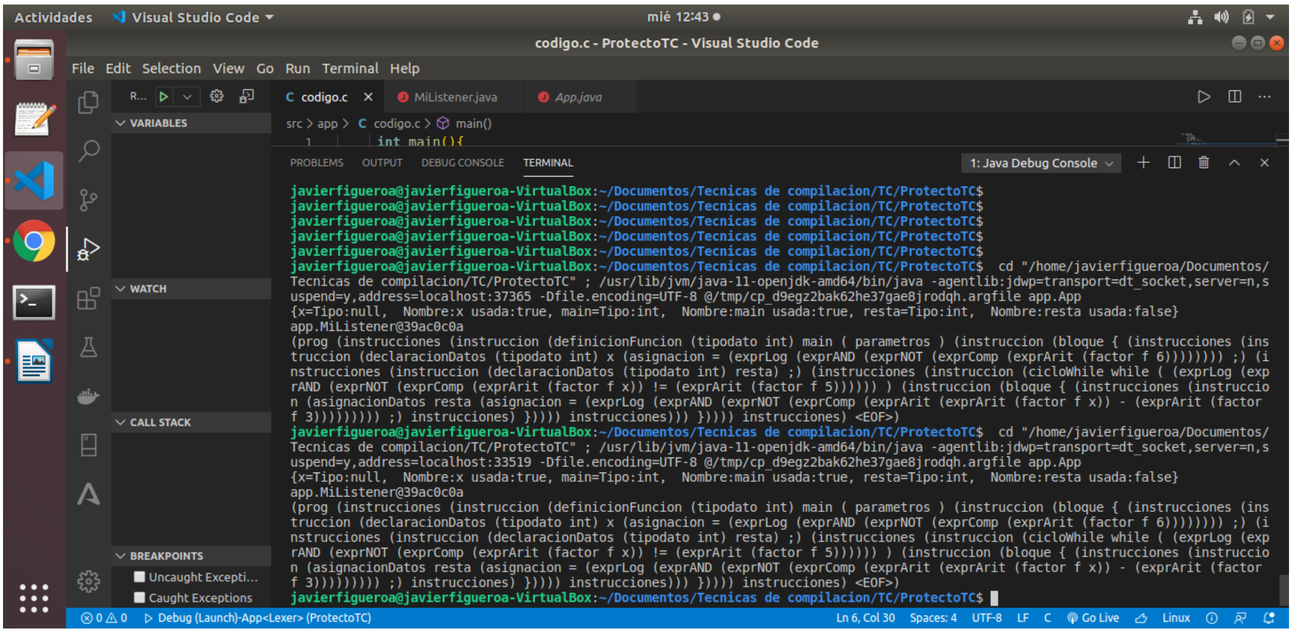
Código utilizado: Imágenes






Actividades
Visual Studio Code
mié 12:40
codigo.c - ProtectoTC - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Java Debug Console
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Uncaught Excepti...
Caught Exceptions
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$ cd "/home/javierfigueroa/Documentos/Tecnicas de compilacion/TC/ProtectoTC" ; /usr/lib/jvm/java-11-openjdk-amd64/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:38121 -Dfile.encoding=UTF-8 @/tmp/cp_d9egz2bak62he37gae8jrodqh.argfile app.App
{x=Tipo:null, Nombre:x usada:true, sum=Tipo:int, Nombre:sum usada:false, main=Tipo:int, Nombre:main usada:true}
app.MiListener@35432107
(prog (instrucciones (instruccion (definicionFuncion (tipodato int) main (parametros) (instruccion (bloque { (instrucciones (instruccion (declaracionDatos (tipodato int) x (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (factor f 2)))))))) ;) (instrucciones (instruccion (declaracionDatos (tipodato int) sum) ;) (instrucciones (instruccion (cicloIf if ((exprLog (exprAND (exprNOT (exprComp (exprArit (factor f x)) <= (exprArit (factor f 3))))))) (instruccion (bloque { (instrucciones (instruccion (asignacionDatos sum (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (exprArit (factor f x)) + (exprArit (factor f 2)))))))) ;) instrucciones) }))))) instrucciones))) }))))) instrucciones) <EOF>)
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
Debug (Launch)-App<Lexer> (ProtectoTC)
Ln 7, Col 2
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Actividades
Visual Studio Code
mié 12:42
codigo.c - ProtectoTC - Visual Studio Code
File Edit Selection View Go Run Terminal Help
codigo.c
MiListener.java
App.java
src > app > codigo.c > main()
1 int main(){
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Java Debug Console
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Uncaught Excepti...
Caught Exceptions
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$ cd "/home/javierfigueroa/Documentos/Tecnicas de compilacion/TC/ProtectoTC" ; /usr/lib/jvm/java-11-openjdk-amd64/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:37785 -Dfile.encoding=UTF-8 @/tmp/cp_d9egz2bak62he37gae8jrodqh.argfile app.App
{x=Tipo:null, Nombre:x usada:true, main=Tipo:int, Nombre:main usada:true, multi=Tipo:int, Nombre:multi usada:false}
app.MiListener@6aa8e115
(prog (instrucciones (instruccion (definicionFuncion (tipodato int) main (parametros) (instruccion (bloque { (instrucciones (instruccion (declaracionDatos (tipodato int) multi) ;) (instrucciones (instruccion (cicloFor for ((declaracionDatos (tipodato int) x (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (factor f 0)))))))) ; (exprLog (exprAND (exprNOT (exprComp (exprArit (factor f x)) < (exprArit (factor f 10)))))) ; (expresion x ++)) (instruccion (bloque { (instrucciones (instruccion (asignacionDatos multi (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (exprArit (factor f 2)) * (exprArit (factor f x)))))))) ;) instrucciones) }))))) instrucciones)) }))))) instrucciones) <EOF>)
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
Debug (Launch)-App<Lexer> (ProtectoTC)
Ln 1, Col 7
Spaces: 4
UTF-8
Select Encoding
Linux


Actividades
Visual Studio Code
mié 12:43
codigo.c - ProtectoTC - Visual Studio Code
File Edit Selection View Go Run Terminal Help
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Uncaught Excepti...
Caught Exceptions
codigo.c
MiListener.java
App.java
src > app > codigo.c > main()
int main(){
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Java Debug Console
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$ cd "/home/javierfigueroa/Documentos/Tecnicas de compilacion/TC/ProtectoTC" ; /usr/lib/jvm/java-11-openjdk-amd64/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:37365 -Dfile.encoding=UTF-8 @/tmp/cp_d9egz2bak62he37gae8jrodqh.argfile app.App
{x=Tipo:null, Nombre:x usada:true, main=Tipo:int, Nombre:main usada:true, resta=Tipo:int, Nombre:resta usada:false}
app.MiListener@39ac0c0a
(prog (instrucciones (instruccion (definicionFuncion (tipodato int) main (parametros) (instruccion (bloque { (instrucciones (instruccion (declaracionDatos (tipodato int) x (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (factor f 6)))))))) ;) (instrucciones (instruccion (declaracionDatos (tipodato int) resta) ;) (instrucciones (instruccion (cicloWhile while ((exprLog (exprAND (exprNOT (exprComp (exprArit (factor f x)) != (exprArit (factor f 5)))))))) (instruccion (bloque { (instrucciones (instruccion (asignacionDatos resta (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (exprArit (factor f x)) - (exprArit (factor f 3)))))))))) ;) instrucciones) })))) instrucciones))) })))) instrucciones) <EOF>)
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$ cd "/home/javierfigueroa/Documentos/Tecnicas de compilacion/TC/ProtectoTC" ; /usr/lib/jvm/java-11-openjdk-amd64/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:33519 -Dfile.encoding=UTF-8 @/tmp/cp_d9egz2bak62he37gae8jrodqh.argfile app.App
{x=Tipo:null, Nombre:x usada:true, main=Tipo:int, Nombre:main usada:true, resta=Tipo:int, Nombre:resta usada:false}
app.MiListener@39ac0c0a
(prog (instrucciones (instruccion (definicionFuncion (tipodato int) main (parametros) (instruccion (bloque { (instrucciones (instruccion (declaracionDatos (tipodato int) x (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (factor f 6)))))))) ;) (instrucciones (instruccion (declaracionDatos (tipodato int) resta) ;) (instrucciones (instruccion (cicloWhile while ((exprLog (exprAND (exprNOT (exprComp (exprArit (factor f x)) != (exprArit (factor f 5)))))))) (instruccion (bloque { (instrucciones (instruccion (asignacionDatos resta (asignacion = (exprLog (exprAND (exprNOT (exprComp (exprArit (exprArit (factor f x)) - (exprArit (factor f 3)))))))))) ;) instrucciones) })))) instrucciones))) })))) instrucciones) <EOF>)
javierfigueroa@javierfigueroa-VirtualBox:~/Documentos/Tecnicas de compilacion/TC/ProtectoTC$
Debug (Launch)-App<Lexer> (ProtectoTC)
Ln 6, Col 30 Spaces: 4 UTF-8 LF C Go Live Linux