Integración de código de Parser y Lexer para generar una aplicación con interfaz gráfica de usuario en Netbeans. Lenguajes y autómatas II, Ene-Jun 24 JGRamos

En Netbeans, crear un proyecto nuevo que incluya la clase **Principal (con el método main())**.
Agregar una ventana al proyecto (JFrame Form y llámela GUI)
Desde la clase **Principal,** en el método **main** invocar a dicha ventana.

```
GUI gui = new GUI();
  gui.setLocationRelativeTo(null);
  gui.setTitle("Analizador semántico");
  gui.setVisible(true);
```

Copiar los archivos **compilerParser**.java y **compilerLexer**.java y pegarlos al paquete donde tiene la clase Principal y la clase GUI. Al hacer la copia de los archivos del lexer y del parser al paquete de su proyecto verifique que queden bien integrados, es posible que tenga que escribir la sentencia

package nombre_de_su_paquete;

en la primera línea de tales archivos para garantizar su correcta incorporación.

no aparece esa parte

SELECCIONAR su proyecto, enseguida botón derecho del ratón, del menú desplegado elegir, Properties -> Libraries -> Compile -> Classpath ->ícono **+** -> Add JAR/Folder -> enseguida buscar y seleccionar el archivo **antlr-3.5.2-complete.jar** en la ruta que lo tenga guardado y hacer click en botón **OK**. Esto garantizará que se pueda reconocer las clases de ANTLR.

En la ventana GUI agregar dos textArea, cambie el nombre para que se llamen: taEntrada y taSalida

Agregue un botón cuya acción será tomar el texto de taEntrada y pasárselo al parser. Para ello seleccione el botón y haga click en botón derecho del mouse y elija Events->Mouse->MouseClicked para agregar el código del manejador de eventos del click del mouse para ese botón.

Enseguida se muestra el código asociado al click de tal botón.

En la mayoría del código las excepciones son introducidas por Netbeans
El código que nos toca introducir a mano aparece en negritas. En este caso la gramática se llemó compiler.g, de ahí el nombre de las Clases **compilerLexer** y **compiler**Parser

```
//////////////////////////////////////////////////////////////////////////////////////////
  ANTLRInputStream input=null;

      String s= "";      s= taEntrada.getText();
      StringBufferInputStream str = new StringBufferInputStream(s);

      try {
          input = new ANTLRInputStream(str);
      } catch (IOException ex) {
          Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
      }

      compilerLexer lexer = new compilerLexer(input);
      CommonTokenStream tokens = new CommonTokenStream(lexer);

      compilerParser parser = new compilerParser(tokens);
                        parser.setSalida(taSalida);

      try {
          parser.inicial();
```

Integración de código de Parser y Lexer para generar una aplicación con interfaz gráfica de usuario en Netbeans. Lenguajes y autómatas II, Ene-Jun 24 JGRamos

```
    } catch (RecognitionException ex) {
        Logger.getLogger(GUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

En la clase compilerParser, declarar un atributo que recibirá una referencia al textArea y que se empleará para escribir la salida del compilador. La declaración será:

private javax.swing.JTextArea salida;

y un método para recibir la referencia

```
public void setSalida(javax.swing.JTextArea _salida)
   {        salida=_salida;    }
```

Buscar todos los System.out.println(parámetros) y cambiarlos por un
salida.append(cadena de texto);

Esto hará que las salidas de texto se escriban en el textArea de salida.

en donde poner esta parte, ??

Para eso el archivo de la gramática debió generarse con el siguiente código para manejo de errores:
El código va después de grammar Compiler;

```
@members {
   public void displayRecognitionError(String[] tokenNames,
                                       RecognitionException e) {
      String hdr = getErrorHeader(e);
      String msg = getErrorMessage(e, tokenNames);
      System.out.println("Que te fijes!!!: " + hdr + " " + msg);
   }
}
```

El código aparecerá en el Parser y será necesario cambiarlo para mandar los errores al TextArea denominado salida