Icono

Descripción generada automáticamenteUn dibujo de una persona

Descripción generada automáticamente con confianza bajaINSTITUTO POLITÉCNICO NACIONALESCUELA SUPERIOR DE CÓMPUTO

**PRACTICA 2**

“YACC EN JAVA”

ALUMNO:

LOPEZ PEREZ ALBERTO ANDREI

PROFESOR:

TECLA PARRA ROBERTO

GRUPO:

5CM4

MATERIA:

COMPILADORES

# Introducción

En esta práctica se buscó implementar los conocimientos de YACC desde la perspectiva de otro lenguaje de programación como lo es Java. En él para poder hacer uso de todo lo aprendido de YACC se buscó la realización de lo siguiente:

Use la carpeta grafibasi para dibujar círculos, líneas y rectángulos

Donde en la carpeta grafibasi que elegimos ya contiene varias de las clases necesarias para construir lo indicado donde solo es cuestión de modificar ciertos elementos para permitirnos el dibujado de esas tres formas en cualquier lugar de un canvas al mismo tiempo. Además nos pide realizar los siguientes ejemplos:

Dibujar un coche use 2 círculos y 2 rectángulos

Dibujar una casa con techo de 2 aguas y una puerta use 2 rectángulos y 2 líneas inclinadas 45 y 225 grados

Dibujar una persona use un circulo (cabeza), y 4 líneas ( 1 línea horizontal para los brazos y una vertical para el tronco y 2 líneas inclinadas 45 y 225 grados para las piernas)

Por lo que, para la realización de lo anterior, la compresión de como funciona la gramática es de suma importancia

# Desarrollo

Para empezar todo lo anterior, debemos mencionar que cada instrucción está dada por el nombre de lo que se quiere dibujar y después los parámetros necesarios separados por espacios, donde para hacer uso de ellos la gramática nos dice que:

Línea: line x1 y1 x2 y2 ; *(donde se ponen las coordenadas de inicio a fin)*

Círculo: circulo r x y ; *(donde se pone el radio y la posición en sus coordenadas)*

Rectángulo: rectángulo x1 y1 x2 y2 ; *(donde se ponen las coordenadas de inicio a fin)*

Por lo que entendiendo lo anterior salen las producciones:



Producciones de inst en forma.y

Como se puede observar, también se tiene una producción para poder cambiar el color los cuales van del 0 al 2 siendo el rojo, verde y azul.

También con las acciones en las producciones podemos darnos cuenta que aquí tenemos una implementación de Máquina Virtual de Pila, muy similar a la vista en **HO4.y,** donde también modificamos el archivo en grafibasi para poder permitir el ingreso de estas coordenadas haciendo más pop en la pila y de este modo, dibujar las figuras en diferentes zonas.



Métodos en Máquina.java para poder generar las figuras haciendo pop en la pila.

Finalmente, es importante mostrar el archivo dibujos.txt generado para poder probar el programa.

//COCHE

color 2 ;

rectangulo 100 100 250 75 ;

rectangulo 175 50 100 50 ;

color 0 ;

circulo 70 125 150 ;

circulo 70 250 150 ;

//CASA

color 1 ;

rectangulo 50 325 175 150 ;

color 2 ;

rectangulo 110 395 60 80 ;

color 0 ;

line 50 325 140 280 ;

line 140 280 225 325 ;

//PERSONA

color 0 ;

circulo 60 400 300 ;

color 2 ;

line 430 360 430 460 ;

line 430 380 500 320 ;

line 430 380 360 320 ;

line 430 460 380 500 ;

line 430 460 480 500 ;

## Compilación y Ejecución

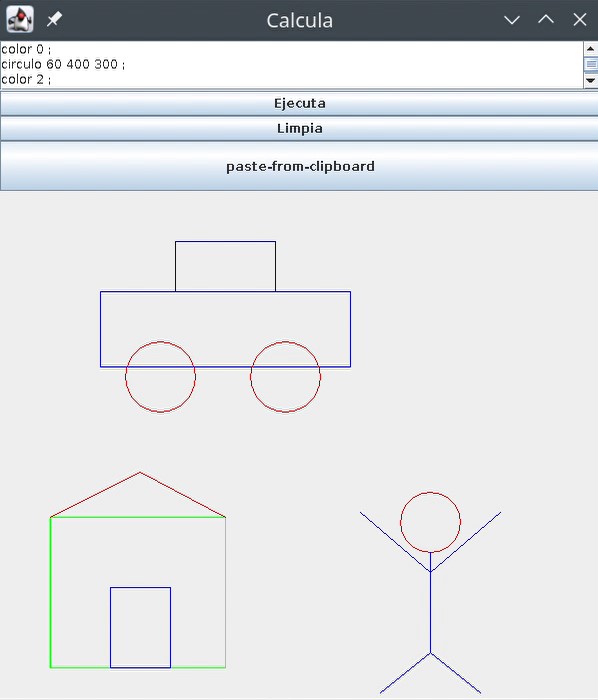
Para poder compilar y ejecutar:

Compilación de YACC: "yacc -J forma.y"

Compilación de todos los archivos java: javac \*.java

Ejecución: "java Parser"

## Programa en Ejecución



Pantalla de Grafibasi con los dibujos de ejemplo

# Conclusiones

Esta práctica, gracias a que el profesor nos otorgó ya mucho código para implementarlo, fue realmente sencillo realizar los cambios menores para poder hacer lo que se pide en a descripción de la práctica. Sin embargo, ciento que la dificultad de la práctica se presenta antes de realizarla ya que encontrar un YACC que otorgue el código en lenguaje Java fue lo difícil, tanto así que por cuestiones de JDK y el mismo Yacc me fue más útil usar una distribución de Linux para poder realizar la práctica. Además, debido a como funciona la gramática, realizar el archivo para poder generar las figuras indicadas (coche, casa y persona), fue algo de lo que hizo que ocupara un poco más de tiempo para que resultara todo perfecto. Si bien me gusto más utilizar Java que C en estos casos ( y sobre todo con lo de la máquina de pila) ya que con el paradigma orientado a objetos se logra comprender mejor las diversas secciones de la misma máquina y modificar los archivos del profe (y comprenderlos) fue mucho más sencillo ya que es más auto descriptivo con el paradigma.