



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)



COMPILADORES

NOMBRE DEL ALUMNO:

- SANTOS MÉNDEZ ULISES JESÚS

PRÁCTICA:

- USO DE JAVA PARA MODO GRÁFICO

NÚMERO DE PRÁCTICA: 2

OPCIÓN 2:

- DIBUJAR USANDO LA CARPETA GRAFIBASI

FECHA DE ENTREGA:

- 03/04/2023

GRUPO:

- 3CM14

Uso de Java para modo gráfico (Grafibasi)

Introducción:

En esta práctica se busca aplicar lo que se aprende durante el curso, en un lenguaje de programación distinto, en esta ocasión se ocupa JAVA, haciendo uso de YACC en Java así como el uso de las distintas bibliotecas que tiene este lenguaje orientado a objetos.

En la carpeta que el maestro nos compartió vienen distintos programas por etapas de grafibasi, donde ya hay distintas clases para hacer dibujos o para manejar una imagen, no obstante para poder ejecutar esta práctica se requiere hacer algunos cambios en la gramática y en el archivo de texto donde vienen algunas cadenas a ingresar para generar las figuras que se solicitan.

Se busca cumplir con el siguiente requerimiento en el documento:

“Use la carpeta grafibasi para dibujar círculos, líneas y rectángulos (usar polimorfismo)”

Se deben realizar las siguientes figuras:

- 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 círculo (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)

Desarrollo:

Principalmente se tuvo que realizar la instalación del JDK y el JRE en Ubuntu, ya que al ingresar el comando `java -version` en la terminal, nos arrojó una cadena que nos decía que no existía dicha orden, no obstante nos dio algunas formas de instalación,

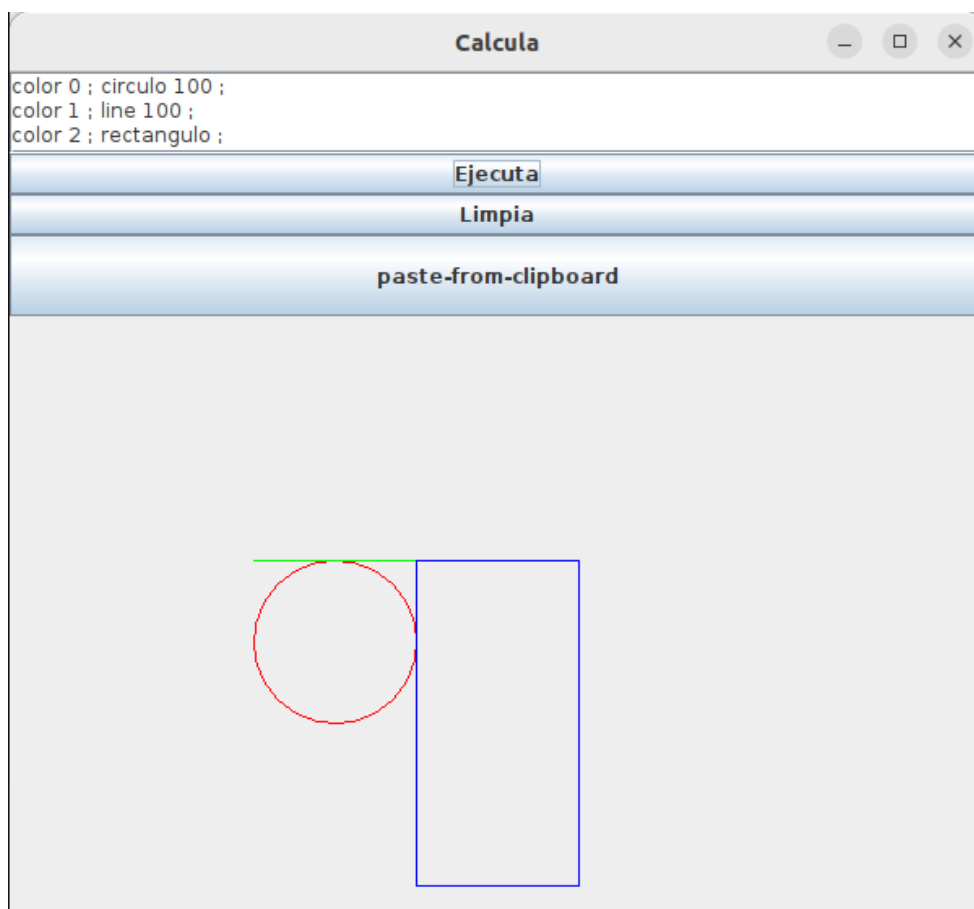
```
ulisespc04@ulisesm04:~$ java -version
java version "20" 2023-03-21
Java(TM) SE Runtime Environment (build 20+36-2344)
Java HotSpot(TM) 64-Bit Server VM (build 20+36-2344, mixed mode, sharing)
```

Después de la instalación, se tuvo que instalar YACC para JAVA que es `byaccj`, de esta forma se generará un archivo de salida con extensión `.java`, en todos los casos vistos en grafibasi generaba un `Parser.java` y un `ParserVal.java`.

```
ulisespc04@ulisesm04:~$ byaccj
usage:
byaccj [-dlrtvj] [-b file_prefix] [-Joption] filename
where -Joption is one or more of:
    -J
    -Jclass=className
```

Inicialmente durante el laboratorio se nos fue solicitado compilar a forma.y y ejecutar a Parser.java

El ejemplo que muestro a continuación fue el que se compiló y ejecutó:



En la gramática del archivo forma.y tenemos la estructura de la gramática que construiremos para hacer las producciones necesarias:

```
%%  
list :  
    | list ';'   
    | list inst ';' {  
        maq.code("print"); maq.code("STOP"); return 1 ;  
    }  
    ;  
inst: NUMBER { ((Algo)$$obj).inst=maq.code("constpush");
```

```

    maq.code(((Algo)$1.obj).simb);

}

| RECTANGULO NUMBER NUMBER NUMBER NUMBER {

    maq.code("constpush");
    maq.code(((Algo)$2.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$3.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$4.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$5.obj).simb);
    maq.code("rectangulo");

}

| LINE NUMBER NUMBER NUMBER NUMBER {

    maq.code("constpush");
    maq.code(((Algo)$2.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$3.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$4.obj).simb);
    maq.code("constpush");
    maq.code(((Algo)$5.obj).simb);
    maq.code("line");

}

| CIRCULO NUMBER NUMBER NUMBER {

    maq.code("constpush");
    maq.code(((Algo)$2.obj).simb);

```

```

maq.code("constpush");
maq.code(((Algo)$3.obj).simb);
maq.code("constpush");
maq.code(((Algo)$4.obj).simb);
maq.code("circulo");
    }
| COLOR NUMBER { maq.code("constpush");
    maq.code(((Algo)$2.obj).simb);
    maq.code("color");
    }
;
%%

```

Lo que se tiene en la gramática es que se tiene implementada una máquina virtual de pila como fue vista en clase, cuenta con las funciones constpush, también se tiene una producción para poder cambiar el color que va del 0 al 2 siendo rojo, verde y azul.

Este fragmento es muy similar al HOC4 visto en clase donde se convierten expresiones al postfijo haciendo uso de una máquina virtual de pila.

Finalmente se tiene el archivo creado de dibujos.txt donde se pusieron las cadenas a ingresar en la GUI:

```

color 2 ;
rectangulo 300 100 250 75 ;
rectangulo 370 50 100 50 ;
color 0 ;
circulo 70 320 150 ;
circulo 70 450 150 ;
color 1 ;
rectangulo 40 60 175 150 ;

```

```
color 2 ;
rectangulo 100 130 60 80 ;
color 0 ;
line 40 60 120 30 ;
line 215 60 120 30 ;
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 ;
```

En el fragmento anterior se colocaron las coordenadas correspondientes para las funciones que se ocupan como drawRect, drawOval y drawline en sus clases correspondientes donde vienen mejor definidas.

Se procede a compilar el programa haciendo uso de byjaccj

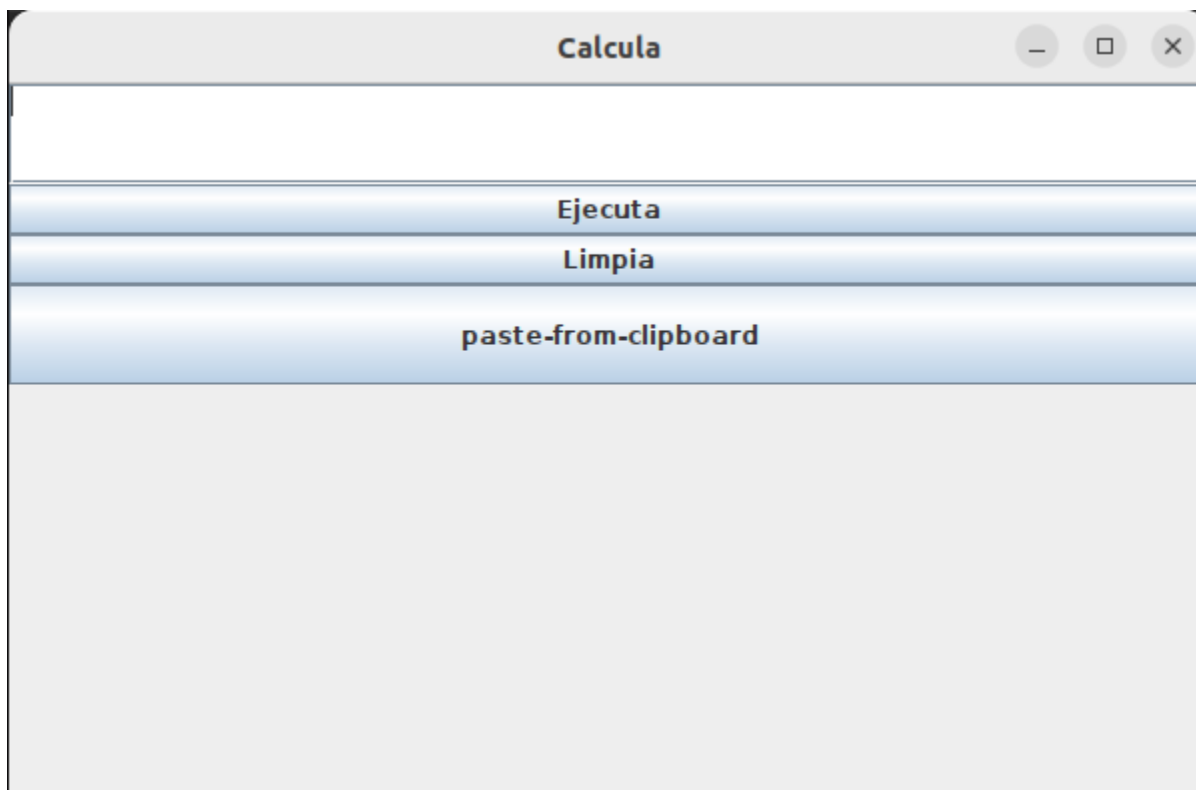
```
ulisespc04@ulisesm04:~/Descargas/practica2$ byaccj -J forma.y
ulisespc04@ulisesm04:~/Descargas/practica2$ javac *.java
Maquina.java:60: warning: non-varargs call of varargs method with inexact argument type for last parameter;
                     metodo=c.getDeclaredMethod(inst, null);
                                             ^
    cast to Class for a varargs call
    cast to Class[] for a non-varargs call and to suppress this warning
Maquina.java:61: warning: non-varargs call of varargs method with inexact argument type for last parameter;
                     metodo.invoke(this, null);
                                   ^
    cast to Object for a varargs call
    cast to Object[] for a non-varargs call and to suppress this warning
Maquina.java:80: warning: [removal] Double(double) in Double has been deprecated and marked for removal
                     pila.push(new Double(s.val));
                               ^
Parser.java:525: error: unreachable statement
break;
^
Note: Maquina.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 error
3 warnings
```

Se generó un error y esto impide que se puedan compilar las clases, esto se soluciona poniendo como comentario la línea del error, esto es debido al return 1 y el break; ya que el return 1 funciona como un final del sistema

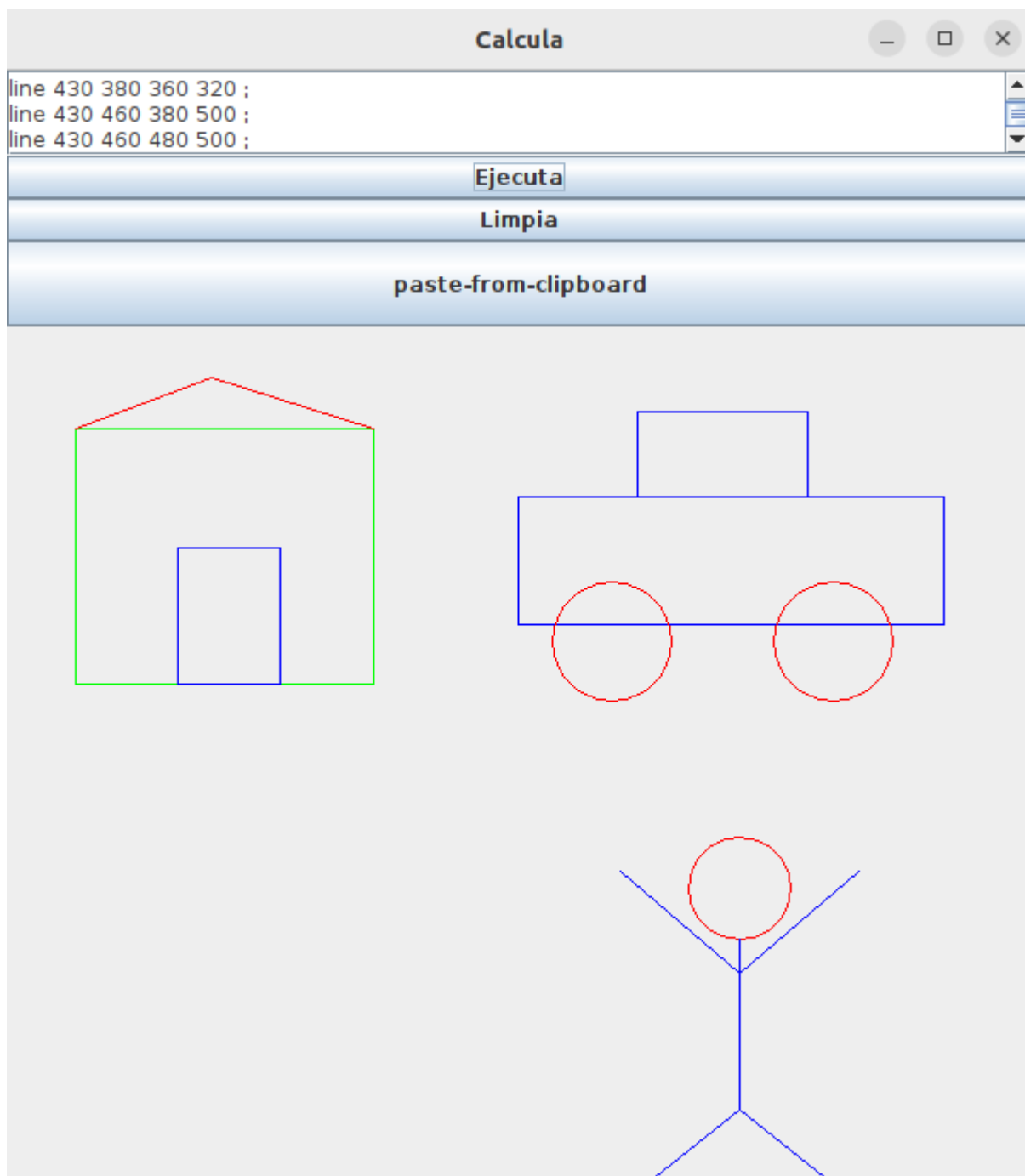
```
case 3:  
  // #line 23 "forma.y"  
  {  
      maq.code("print"); maq.code("STOP"); return 1 ;  
  }  
  // break;
```

Después que se comentó esa línea se compilan las clases haciendo uso de javac *.java

Finalmente se ejecuta el programa con java Parser abriendo una GUI



Copiamos las cadenas del archivo dibujos.txt y presionamos el botón paste-from-clipboard



Conclusiones:

En conclusión me fue algo laborioso el instalar el JDK de Java aunque después se me hizo más complicado entender el error que me daba el archivo Parser.java., al final con el tema visto en clase me fue más sencillo entender el comportamiento de la máquina virtual de pila y el cómo funcionan las funciones para hacer las figuras, ahora no solo puedo ocupar yacc para c sino también para java, entonces me resulta muy útil si quiero emplear una GUI o si solo quisiera ejecutar en una terminal.