

MANUAL TÉCNICO

IDE:

Netbeans 8.2

Version de JAVA:

java versión “1.8.0-201”

JAR agregados:

- java – cup – 11b.jar
- java – cup – 11b-runtime.jar
- jflex – full – 1.7.0.jar

jFlex:

Jlex es una herramienta desarrollada en Java que toma como entrada un archivo “entrada”, con este crea un archivo fuente java entrada.lex.java correspondiente al analizador léxico. En otras palabras, Jlex es un generador de analizadores léxicos. Los analizadores léxicos toman como entrada una cadena de caracteres y la convierten en una secuencia de tokens.

JCup:

JAVA CUP es un parser-generator. Es un analizador sintáctico que construye un parser para gramáticas tipo LALR(1), con código de producción y asociación de fragmentos de código JAVA. Cuando una producción en particular es reconocida, se genera un archivo fuente Java, parser.java que contiene una clase parser, con un método Symbol parser().

Instalacion de Jflex y Jcup:

1. JDK (Descarga el JDK de la página Web de Sun para los sistemas Operativos Linux, Windows y Solaris). Si se tiene instalado alguna versión de JAVA, solo es necesario buscar el directorio en el que se encuentra el JDK de esta.
2. Archivo para la generación de las clases para Jlex llamado Main.java que se puede descargar de la página Web de Jlex.
3. Descarga del sitio Web de CUP el código fuente de los archivos necesarios.

GRAMÁTICA VB:

start with B

B ::= METODOS B

| COMENTARIO_A B
| COMENTARIO_B B
|
| error B

VARIABLES ::= Dim id C CONC_VAR_DEF

ASIGNACIONES ::= id igual VALOR CONC_VAR_AS
| condThis punto id igual VALOR CONC_VAR_AS

CONC_VAR_DEF ::= coma id C CONC_VAR_DEF
| puntoComa ENTER

CONC_VAR_AS ::= coma id C CONC_VAR_AS
| puntoComa ENTER

C ::= condAs condTipo
|

VAR ::= tipoInt
| tipoChar
| tipoFloat

OPERACION ::= parentesis_a OPERACION parentesis_b
| OPERACION mas OPERACION
| OPERACION menos OPERACION
| OPERACION por OPERACION
| OPERACION div OPERACION
| OPERACION modulo OPERACION
| TIPO_VAL

TIPO_VAL ::= id
| numero
| numero punto numero
| comillaSimple id comillaSimple
| comillaSimple numero comillaSimple
| menos id
| menos numero
| menos numero punto numero

METODOS ::= tipoPublic condSub id parentesis_a PARAMETROS parentesis_b
 CUERPO_METODO condFin condSub

| tipoPublic condFunction id parentesis_a PARAMETROS parentesis_b condAs condTipo
CUERPO_METODO condFin condSub

PARAMETROS ::= PARAMETROS2 id condAs condTipo

PARAMETROS2 ::= PARAMETROS2 condAs condTipo id coma

TIPO_METODO ::= tipoChar]

| tipoFloat

| tipoChar

| tipoVoid

CUERPO_METODO ::= CUERPO_METODO VARIABLES

| CUERPO_METODO ASIGNACIONES

| CUERPO_METODO IF

| CUERPO_METODO FOR

| CUERPO_METODO WHILE

| CUERPO_METODO DO_WHILE

| CUERPO_METODO MENSAJE

| CUERPO_METODO SELECT

| CUERPO_METODO LECTURA_DATOS

| CUERPO_METODO COMENTARIO_A

| CUERPO_METODO COMENTARIO_B

| CUERPO_METODO error

RETORNO ::= retorno OPERACION puntoComa ENTER

IF ::= condIf EXT2 condThen CUERPO_METODO ELSE condFin condif

ELSE ::= condElseIf EXT2 condThen CUERPO_METODO ELSE

| condElse condThen CUERPO_METODO RETORNO

| ENTER

FOR ::= codFor parentesis_a INT id igual VAL_FOR puntoComa EXT2 puntoComa id INC
parentesis_b llave_a CUERPO_METODO llave_b

INT ::= tipoInt

INC ::= igual OPERACION

OP ::= menor igual

| menor

| mayor igual

| mayor

- | igual
- | igual igual
- | admiracion igual

WHILE ::= condWhile EXT2 CUERPO_METODO condEndWhile

DO_WHILE ::= condDo CUERPO_METODO condLoop condWhile EXT2

SELECT ::= condSelect conCase parentesis_a id parentesis_b CASOS condEnd condSelect

CASOS ::= condCase VAL_SELECT CUERPO_METODO CASOS

- | condCase condElse CUERPO_METODO
- |

EXT2 ::= EXT2 and EXT2

- | EXT2 or EXT2

- | BOOLEAN3

- | admiracion parentesis_a BOOLEAN3 parentesis_b

- | admiracion BOOLEAN3

BOOLEAN3 ::= OPERACION OP COPERACION

LECTURA_DATOS ::= id igual TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

- | TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

TIPO_INPUT ::= intInput

- | FloatInput

- | CharInput

TEXTO ::= VAL TEXTO2

- | comilla TEXTO_PLANO comilla TEXT2

TEXTO2 ::= mas TEXTO

- | ampersand TEXTO
- |

COMENTARIO_A ::= comentario_a TEXTO_PLANO

COMENTARIO_B ::= comentario_b TEXTO_PLANO Comentario_c

GRAMATICA JAVA

start with AA

A ::= A tipoPublic clase id llaveA ENTER B llaveB ENTER

B ::= VARIABLES B
| METODOS B
| CONSTRUCTOR B
| COMENTARIO_A B
| COMENTARIO_B B
|
| error B

VARIABLES ::= VAR id C CONC_VAR_DEF

ASIGNACIONES ::= id igual VALOR CONC_VAR_AS
| condThis punto id igual VALOR CONC_VAR_AS

CONC_VAR_DEF ::= coma id C CONC_VAR_DEF
| puntoComa ENTER

CONC_VAR_AS ::= coma id C CONC_VAR_AS
| puntoComa ENTER

C ::= igual VALOR
|

VAR ::= tipoInt
| tipoChar
| tipoFloat

OPERACION ::= parentesis_a OPERACION parentesis_b
| OPERACION mas OPERACION
| OPERACION menos OPERACION
| OPERACION por OPERACION
| OPERACION div OPERACION
| OPERACION modulo OPERACION
| TIPO_VAL

TIPO_VAL ::= id
| numero
| numero punto numero
| comillaSimple id comillaSimple
| comillaSimple numero comillaSimple
| menos id
| menos numero
| menos numero punto numero

METODOS ::= tipoPublic TIPO_METODO id parentesis_a PARAMETROS parentesis_b llave_a
CUERPO_METODO RETORNO llave_b ENTER
| tipoPublic TIPO_METODO id parentesis_a parentesis_b llave_a CUERPO_METODO
RETORNO llave_b ENTER

CONSTRUCTOR ::= tipoPublic id parentesis_a PARAMETROS parentesis_b llave_a
CUERPO_METODO llave_b ENTER
| tipoPublic parentesis_a parentesis_b llave_a CUERPO_METODO llave_b ENTER

PARAMETROS ::= PARAMETROS2 VAR id

PARAMETROS2 ::= PARAMETROS2 VAR id coma
|

TIPO_METODO ::= tipoChar]
| tipoFloat
| tipoChar
| tipoVoid

CUERPO_METODO ::= CUERPO_METODO VARIABLES
| CUERPO_METODO ASIGNACIONES
| CUERPO_METODO IF
| CUERPO_METODO FOR
| CUERPO_METODO WHILE
| CUERPO_METODO DO_WHILE
| CUERPO_METODO MENSAJE
| CUERPO_METODO SWITCH
| CUERPO_METODO LECTURA_DATOS
| CUERPO_METODO COMENTARIO_A
| CUERPO_METODO COMENTARIO_B
| CUERPO_METODO error

RETORNO ::= retorno OPERACION puntoComa ENTER
|

IF ::= condIf parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO RETORNO llave_b

ELSE ::= condElse condIf parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO RETORNO
llave_b
| condElse llave_a CUERPO_METODO RETORNO llave_b
| ENTER

FOR ::= codFor parentesis_a INT id igual VAL_FOR puntoComa EXT2 puntoComa id INC
parentesis_b llave_a CUERPO_METODO llave_b

INT ::= tipoInt
|

INC ::= igual OPERACION

OP ::= menor igual
| menor
| mayor igual
| mayor
| igual
| igual igual
| admiracion igual

WHILE ::= condWhile parentesis_a EXT2 parentesis_b llave_a CUERPO_METODO llave_b

DO_WHILE ::= condDo llave_a CUERPO_METODO condWhile parentesis_a EXT2 parentesis_b
puntoComa

SWITCH ::= condSwitch parentesis_a id parentesis_b llave_a CASOS llave_b

CASOS ::= noCase VAL_SWITCH dosPuntos CUERPO_METODO BREAK CASOS
| default dosPuntos CUERPO_METODO BREAK
|

BREAK ::= quiebre puntoComa
| RETORNO

EXT2 ::= EXT2 and EXT2
| EXT2 or EXT2
| BOOLEAN3
| admiracion parentesis_a BOOLEAN3 parentesis_b
| admiracion BOOLEAN3

BOOLEAN3 ::= OPERACION OP COPERACION

LECTURA_DATOS ::= id igual TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa
| TIPO_INPUT parentesis_a TEXTO parentesis_b puntoComa

TIPO_INPUT ::= intInput
| FloatInput
| CharInput

TEXTO ::= VAL TEXTO2
| comilla TEXTO_PLANO comilla TEXT2

TEXTO2 ::= mas TEXTO
|

COMENTARIO_A ::= comentario_a TEXTO_PLANO

COMENTARIO_B ::= comentario_b TEXTO_PLANO Comentario_c

