

ESTRUCTURAS DE CÓDIGO DEL COMPILADOR

Las estructuras de código son generadas a través de los componentes léxicos, donde son concatenaciones de los componentes léxicos que siguen un patrón dicha estructura, por ende las estructuras de este compilador que se pueden generar son las que están en la siguiente tabla:

Estructuras de Declaraciones de puerto (DP)	
(Puerto) (Tipo puerto) (Identificador) (Fin de línea)	port_1 proximity proximitySensor!
Estructuras de Declaraciones con valor (DCV)	
(Declaración) (Tipo dato) (Identificador) (Operador asignación) (VALORES) (Fin de línea)	var string holaMundo = 'HolaMundo'!
Estructuras de Declaraciones sin valor (DSV)	
(Declaración) (Tipo de dato) (Identificador) (Fin de línea)	const string number3 !
Estructura de Asignaciones (A)	
(Identificador) (Operador asignación) (VALORES) (Fin de línea)	number2 = 3.3 !
Estructura de funciones de los motores con valor (FMCV)	
(Motor) (Signo agrupación "(") (Identificador) (Signo puntuación ",", ") (VALORES) (Signo agrupación ")") (Fin De línea)	move(motor1,34)!
Estructura de funciones de los motores sin valor (FMSV)	
(Motor) (Signo agrupación "(") (Identificador)(Signo agrupación ")") (Fin De línea)	move(motor1)!
Estructura de Método de impresora a consola (MIC)	
(Método "print") (Signo agrupación "(") (Método "console") (Signo puntuación ",", ") (VALORES Identificador) (Signo agrupación ")") (Fin de línea)	print(console,'El motor1 se esta moviendo')!

RoboKit

Estructura de Método de impresora a LCD (MIL)	
(Método "print") (Signo agrupación "(") (Identificador) (Signo puntuación ",") (VALORES Identificador) (Signo agrupación ")") (Fin de línea)	print(Hola,'El motor1 se esta moviendo')!
Estructura de Operaciones (O)	
(Identificador) (Operador asignación) (Identificador) (Operador aritmético) (Identificador) (Fin de línea)	result = a + b!
Estructura de las funciones sin parámetros (FSP)	
(Estructura control "function") (Identificador) (Signo agrupación "(") (Signo agrupación ")") (Signo agrupación "{") (FMCV FMSV)* (MIC MIL)* (Signo agrupación "}"))	function moverMotor(){ move(motor1)! print(console,'El motor1 se esta moviendo')! }
Estructura de las funciones con parámetros (FCP)	
(Estructura control "function") (Tipo dato) (Identificador) (Signo agrupación "(") (Tipo dato) (Identificador) ((Signos puntuación ",") (Tipo dato) (Identificador))* (Signo agrupación ")") (Signo agrupación "{") (O)* (Método "return") (Identificador) (Fin de línea) (Signo agrupación "}")	function int sumaNumerosInt(int a, int b){ result = a + b! return result! }
Estructura de llamadas de funciones con valor (LFCV)	
(Identificador) (Signo agrupación "(") (VALORES) ((Signo puntuación ",") (VALORES))* (Signo agrupación ")") (Fin de línea)	moverMotor(5)!
Estructura de llamadas de funciones sin valor (LFSV)	
(Identificador) (Signo agrupación "(") (Signo agrupación ")") (Fin de línea)	moverMotor()!
Estructura de control Begin (B)	
(Estructura control "begin") (Signo agrupación "{") ((LFCV) (LFSV) (FMCV) FMSV)* (Signo agrupación "}")	begin{ moverMotor()! detenerMotor()! escribirLCD(5)! }

Estructura de Método de los sensores (MS)	
(Identificador) (Operador asignación) (Identificador) (Signo puntuación ".") (Método sensor) (Signo agrupación "(") (Signo agrupación ")") (Fin de línea)	Distancia = puerto1.distance()!
Estructura de Método Delay (MD)	
(Método sensor "delay") (Signo agrupación "(") ((Identificador) (VALORES)) (Signo agrupación ")") (Fin de línea)	delay(1000)!
Estructura de control If Relacional (IR)	
(Estructura control "if") (Signo agrupación "(") ((Identificador) (VALORES)) ((Operador relacional) (Operador asignación)) ((Identificador) (VALORES)) (Signo agrupación ")") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}")	if(number1>number2){ moverMotor()! }
Estructura de control If Lógico con dos valores (IL2V)	
(Estructura control "if") (Signo agrupación "(") ((Identificador) (VALORES)) (Operador lógico) ((Identificador) (VALORES)) (Signo agrupación ")") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}")	if(number1 and number2){ moverMotor()! }
Estructura de control If Lógico con un valor (IL1V)	
(Estructura control "if") (Signo agrupación "(") ((Identificador) (VALORES)) (Signo agrupación ")") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}")	if(number1){ moverMotor()! }
Estructura de control Else Relacional (ER)	
(Estructura control "if") (Signo agrupación "(") ((Identificador) (VALORES)) ((Operador relacional) (Operador	if(number1>number2){ moverMotor()! }else{

<p>asignación)) ((Identificador) (VALORES)) (Signo agrupación "{") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}")) (Estructura control "else") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}"))</p>	<pre> detenerMotor()! } </pre>
Estructura de control Else Lógico con dos valores (EL2V)	
<p>(Estructura control "if") (Signo agrupación "{") ((Identificador) (VALORES)) (Operador lógico) ((Identificador) (VALORES)) (Signo agrupación "{") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}")) (Estructura control "else") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}"))</p>	<pre> if(number1 and number2){ moverMotor()! }else{ detenerMotor()! } </pre>
Estructura de control Else Lógico con un valor (EL1V)	
<p>(Estructura control "if") (Signo agrupación "{") ((Identificador) (VALORES)) (Signo agrupación "{") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV))* (Signo agrupación "}") (Estructura control "else") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV) (MD))* (Signo agrupación "}"))</p>	<pre> if(number1){ moverMotor()! }else{ detenerMotor()! } </pre>
Estructura de control Loop (L)	
<p>(Estructura control "loop") (Signo agrupación "{") ((MS) (MIC) (MIL) (A) (DCV) (DSV) (FMCV) (FMSV) (LFCV) (LFSV IR IL2V IL1V ER EL2V EL1V) (MD))* (Signo agrupación "}"))</p>	<pre> loop{ if(number1>number2){ moverMotor()! }else{ detenerMotor()! } Distancia = puerto1.distance()! number1 = read(temperatureSensor!) number2 = 25.0 ! } </pre>