

Documentação de linguagem criada em JavaCC

Micael Barreto de Moraes

Igor Fabri Cervelin

1 - Entradas Válidas

A linguagem proposta permite as seguintes entradas desconsiderando a análise léxica das entradas do código.

ID do Token	Uso	Entrada
1	Entradas ignoradas	" ", "\t", "\n", "\r"
2	Abertura de comentários	//
3	Fim de argumento	;
4	Pós-incremento e Pós-decremento	++, --
5	Função	function
6	Condição	if
7	Condição encadeada	else, elseif
8	Laços de repetição	for, do/while
9	Estruturas condicionais	switch/case
10	Quebra de loopings	break
11	Números	[0-9],[0-9]
12	letras e dígitos	[a-z], [A-Z], [0-9]
13	Operadores lógicos	>, <, ==, !=
14	Operadores aritméticos	+, -, *, /
15	abertura e fechamento de chaves	{, }
16	abertura e fechamento de parênteses	(,)

17	Valores booleanos	true, false
18	Tipos	int, double, String, boolean
19	Atribuição de valores	=
20	Textos permitidos	[a-z], [A-Z], [0-9], _
21	Valores permitidos	[a-z], [A-Z], [0-9], _
22	Dois pontos	:

2 - Instruções

A linguagem permite as entradas das seguintes instruções no código com análise léxica não terminalis:

ID da Instrução	Uso	Explicação	Exemplo
1	Comentários	Os comentários devem ser iniciados por duas barras	//Comentário
2	Condições e encadeamentos condicionais	As condições são definidas pela entrada "if" e a devida condição, já o encadeamento ocorre pelas entradas "elseif" com condição e "else" sem condição	if(condição){ //instruções }elseif(condição_2){ //instruções }else{ //instruções }
3	Declaração de variáveis	As declarações com informação dos tipos seguido da variável, podendo haver declaração de variáveis de mesmo tipo na mesma linha com uso de vírgulas entre as mesmas, e também a atribuição dos valores para as variáveis	int a; String b, c; Double d = 3;
4	Operações aritméticas	As operações aritméticas se dão por atribuição de operações	a = a + a; b = b - b; c = c * c;

		à variável, podendo ser operações de soma, subtração, multiplicação e divisão	d = d / d;
5	Operações lógicas	São comparação entre valores sendo eles: maior que, menor que, igual a e diferente de	a > b b < 3 c == d d != a
6	Incremento e decremento	Definição do pós-incremento e pós-decremento de variáveis	i++ i--
7	Laço de repetição: for	Laço a partir de argumentos com incremento ou decremento da variável do argumento	for(i=0, i<10, i++){ //instruções }
8	estrutura condicional: switch/case	Definição do argumento e equivalência de valor	switch(i){ case 1: //instruções break; case 2: //instruções break; }
9	Funções	Define funções com instruções que o código pode ter, podendo ou não receber argumentos	function foo(){ //instruções } function bar(i){ //instruções }
10	Laço de repetição: do/while	Laço a partir da validação do argumento ao final da execução das instruções	do{ //instruções }while(i<10)

3 - Dificuldades

As dificuldades encontradas na confecção deste projeto são de ordem da ferramenta nas tentativas repetitivas e frustradas de gerar o documento automatizado que a ferramenta oferece por meio do jjdoc. Outra dificuldade foi com os finais de argumento que não possuíam o EOL, estes apresentavam erros ao compilar, que foi resolvido com o uso correto do fechamento de chaves no arquivo Main.jj.