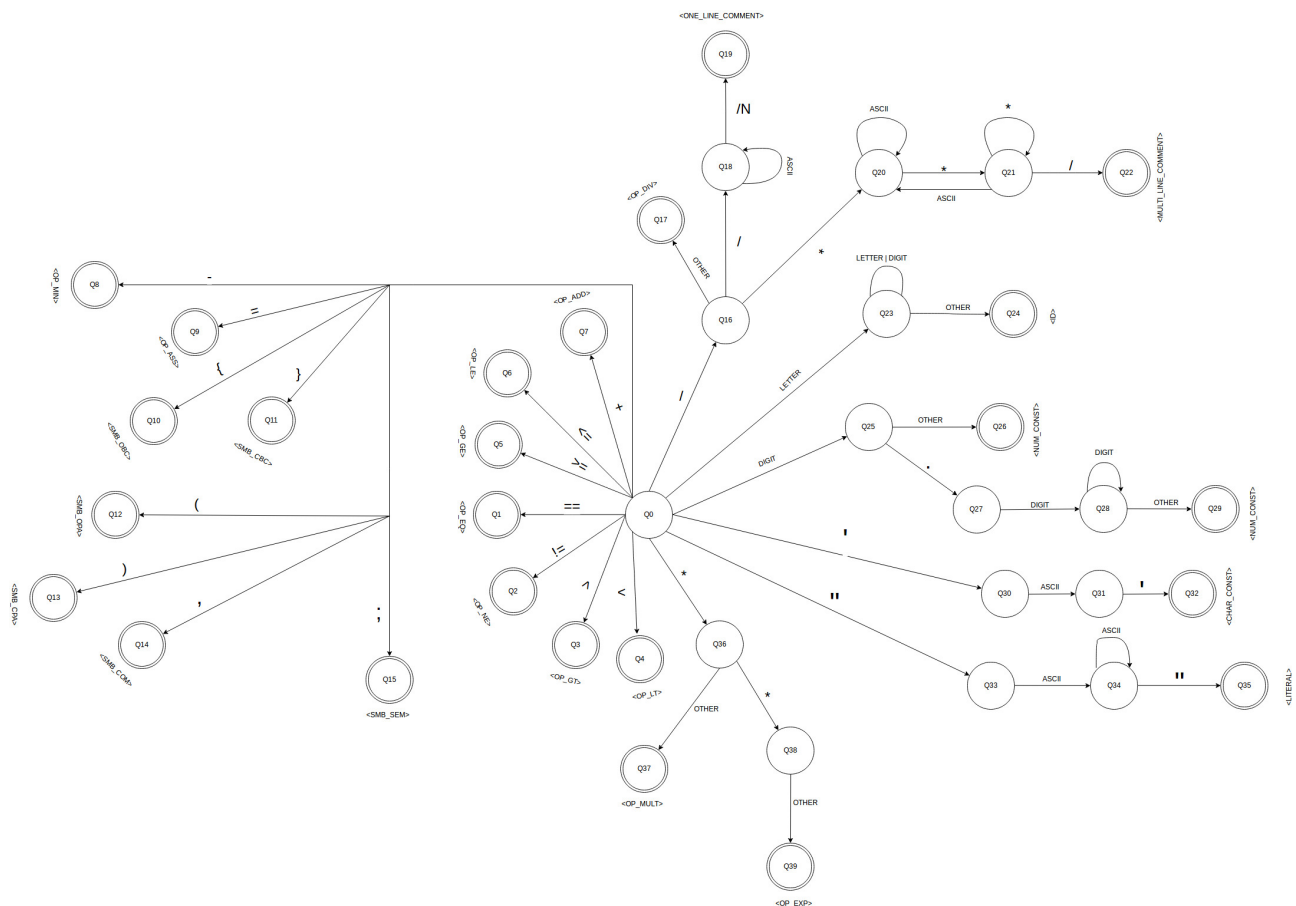


Autômato



Relatório do Analizador Léxico

Lexer.java //Definição do lexer

Lexer() - Cria acesso ao arquivo fonte

getNextToken() - Retorna o próximo token

getState() - Retorna o estado atual do lexer

setState() - Define um novo estado para o lexer

getCurrentChar() - Retorna o caractere atual lido

incrementKnewChar() - Incrementa variável de verificação de caractere desconhecido

Menu.java // Interface gráfica para menu de escolha

Description.java // Descrição do token

Starter.java // Inicializa a execução do programa

Starter() - Construtor de inicialização

FileHandle.java // Manipulação de arquivo

getInstance() - Retorna o acesso ao arquivo fonte

closeFile() - Fecha arquivo aberto

fallbackCursor() - Volta o cursor do arquivo a um caractere anterior

getEof() - Retorna o valor padrão do fim de arquivo

getCurrentLine() - Retorna a linha de leitura atual

getCurrentColumn() - Retorna a coluna de litura atual

setIncrementLine() - Incrementa a linha atual em mais 1

setIncrementColumn() - Incrementa a coluna atual em mais 1

setLookAhead() - Define a próxima posição do cursor de leitura do arquivo

Token.java // Definição de modelo de token

Token() - Construtor de token

getLine() - Retorna linha de token criado

getColumn() - Retorna linha de token criado

getLexeme() - Retorna o Lexema do token criado

getTag() - Retorna a tag do token criado

toString() - Retorna string com padrão do token criado

Keywords.java // Definição das keywords padrão

getAllKeyWords() - Retorna todas keywords criadas

SymbolsTable.java // Definição da tabela de símbolos

setSymbol() - Adiciona símbolo na tabela de símbolos

getToken() - Retorna token existente na tablea de símbolos ou null quando não existir

getDescription() - Retorna descrição do token solicitado

Tags.java // Definição de tags em forma de enumeração

Symbols.java // Definição de condicionais

analyse() - Condicional de verificação de estados que pertençam ao conjunto de símbolos da linguagem

Operators.java // Definição de condicionais

analyse() - Condicional de verificação de estados que pertençam ao conjunto de operadores da linguagem

Constants.java // Definição de condicionais

analyse() - Condicional de verificação de estados que pertençam ao conjunto de constantes da linguagem

LiteralAndIdentifiers.java // Definição de condicionais

analyse() - Condicional de verificação de estados que pertençam ao conjunto de padrão literal e identificadores da linguagem

ErrorMessage.java // Definição de mensagens de erro

invalidCaractere() - Mensagem de caractere inválido

incompleteToken - Mensagem de token incompleto

unclosedString() - Mensagem de literal não fechado

moreThanOneLineString() - Mensagem de literal com mais de uma linha

emptyString() - Mensagem de literal vazio, sem conteúdo

unclosedComment() - Mensagem de comentário de duas linha sem fechar

Execução do Programa

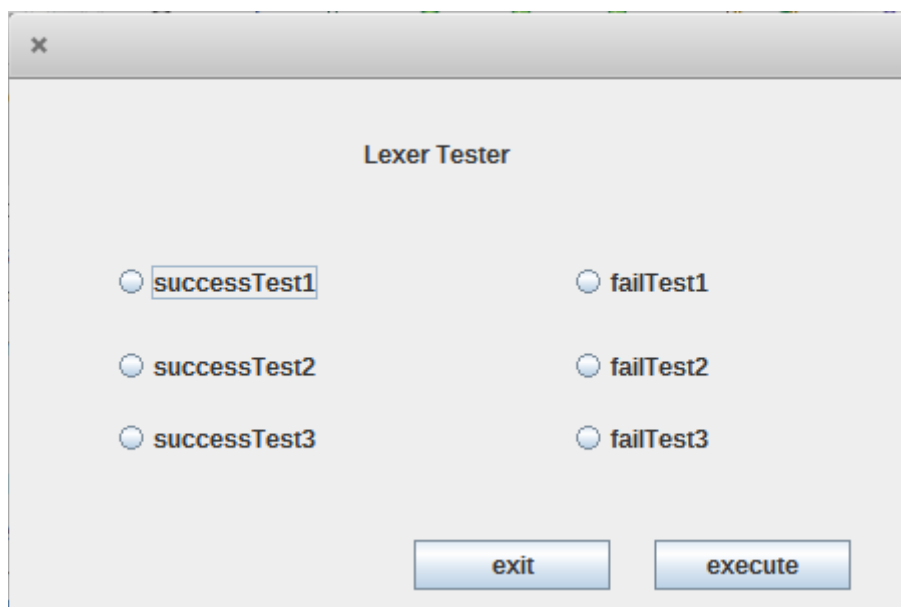
Baixe e instale o JDK :

<http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html>

Instale o ambiente de desenvolvimento eclipse:

<http://www.eclipse.org/downloads/>

Após importar o projeto e executar irá aparecer a seguinte interface:



No qual deve-se marcar os casos de sucesso ou falha e clicar no botão executar. Para mostrar no console o resultado do lexer e a tabela de símbolos.