

EXA869 – MI Processadores de Linguagens de Programação

Problema 1: *Estude e trabalhe duro, não “para inglês ver”!*

Tema: Analisador Léxico

Cronograma

21/8	Apresentação do Problema 1
23/8	Problema 1
28/8	Problema 1
30/8	Problema 1
04/9	Problema 1
06/9	Problema 1
08/9	Entrega do código-fonte (via e-mail)
11/9	Entrega do relatório do Problema 1 & Apresentação do Problema 2

Problema

Na chamada Era Vitoriana, “o Sol nunca se punha no Império Britânico”; e os ingleses eram tão dominantes geopoliticamente que foram alcunhados “Os senhores do mundo”. Até o futebol os caras inventaram! :)

Também sempre se notabilizaram pela espionagem: “Meu nome é...”.

Ernest Doyle Collins, de codinome Mula Velha, é o pior espião da Coroa de todos os tempos, e só não foi expulso ainda do MI-6, porque seu pai é um magnata que enche o Governo deles de doações no mínimo eticamente suspeitas. A Mula trabalha com burocracia, sonha em se reabilitar no serviço-secreto internacional, e uma denúncia cifrada que chegou em sua conta de e-mail é a última chance (por engano do remetente, que é *quase* tão incompetente quanto Collins, e fez a postagem errado; mas nos limitemos à história de um imbecil por problema neste MI).

Collins, A Mula Velha, não faz a menor ideia de como decodificar a mensagem tão cara a ele, mas um anexo não codificado contém a seguinte instrução do remetente: “Send the message to the students of MI Processors of Programming Languages of Uefs. A Lexical Analyzer must be implemented”. E, depois do trecho encriptado, a seguinte tabela está descrita:

Key-words	variables, methods, constants, class, return, empty, main, if, then, else, while, for, read, write, integer, float, boolean, string, true, false, extends
Identifiers	Letter (Letter Digit _)*
Numbers	(-)? Digit+ (. Digit+)?
Letter	a..z A..Z
Digit	0..9
Symbol	ASCII from 32 to 126 (except 34 and 39)
String	"(Letter Digit Symbol)*"
Character	'(Letter Digit Symbol)'
Operators	+ - * / == != > >= < <= && = ++ -- .
Delimiters	; , () { } []
Comments	<i>/* this is a block comment */ // this is an unique line comment</i>

Produto

Para ajudar A Mula Velha a descriptar a mensagem, o grupo tutorial deve implementar um analisador léxico baseado nas informações descritas naquela tabela.

O código-fonte (em Python, Java, C ou C++) do analisador léxico, que será aceito feito em dupla de matriculados, deve ser entregue por e-mail (**tdmaia@uefs.br & thiagodmmaia@gmail.com**) até as 23h59min do dia 08/9/24. Deve ser criada uma interface gráfica para o analisador. Essa interface deve apresentar três *frames*: um para a inserção do código-fonte, outro para a impressão da lista de *tokens* proveniente da análise léxica, e um terceiro, para a exibição dos erros léxicos, caso existam. Se não houver erro, uma mensagem de sucesso deve ser exibida.

Além disso, cada matriculado ou dupla deve entregar um relatório técnico, **em formato SBC**, contendo os seguintes itens: (1) Materiais e Métodos (descrição das técnicas utilizadas para a análise léxica, descrição das classes, atributos e métodos mais significativos e explicação do funcionamento do código); (2) Experimentos e Análise dos Resultados (apresentação dos testes elaborados para o código desenvolvido e discussão dos resultados obtidos – positivos e negativos). O relatório deve ser entregue nos mesmos e-mails mencionados, no dia 11/9/24.

Haverá prejuízo de dois pontos por **um** dia de atraso na entrega do código-fonte. Após esse prazo, o trabalho não será mais aceito. O relatório será aceito somente com o aceite do código-fonte.

Recursos para Aprendizagem

- AHO, A. V.; SETHI, S. & ULMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. Rio de Janeiro: LTC, 1995.
- LOUDEN, K. C. **Compiladores – Princípios e Práticas**. São Paulo: Thomson, 2004.
- HOPCROFT, J. E. *et al.* **Introdução à Teoria dos Autômatos, Linguagens e Computação**. 1ª edição, Editora Campus, 2002.
- MENEZES, P. F. B. **Linguagens Formais e Autômatos**. 5ª edição, Editora Sagra-Luzzatto, 2005.