

## Compiladores (CCP-N7)

Prof. Responsável: Carlos Henrique Odenique Jardim

e-mail: [cursospatrao@gmail.com](mailto:cursospatrao@gmail.com)

### 1º Trabalho Prático

#### 1. Objetivos

O objetivo deste trabalho é colocar em prática os conceitos estudados na disciplina Compiladores, particularmente sobre a Análise Léxica.

#### 2. Entrega do trabalho

O trabalho deverá ser realizado em grupo de 2 alunos e deverá ser entregue até o dia 18/04. Não serão aceitos trabalhos após esta data. Os grupos deverão apresentar os trabalhos nas aulas dos dias 18/04 ou 20/04, conforme distribuição que faremos na aula.

Devem ser entregues todos os códigos fontes do programa, escritos na linguagem Java, sendo estes documentados internamente. Também deve ser entregue um arquivo JAR executável do programa. Se alguma biblioteca complementar for necessária, ela deve estar incluída no projeto. Os códigos fontes devem estar organizados na forma de um projeto do software Eclipse.

O trabalho deve ser entregue, compactado em um arquivo no formato ZIP ou RAR, para o e-mail "[cursospatrao@gmail.com](mailto:cursospatrao@gmail.com)". O arquivo compactado não deve conter arquivos executáveis, com extensão "exe". **No campo assunto do e-mail deve constar obrigatoriamente a descrição "[COMPILADORES:TRABALHOB1]"**. No corpo do e-mail deve constar o nome e o respectivo código de matrícula de cada integrante do grupo. Caso estes requisitos não sejam cumpridos, o trabalho não será corrigido.

### 3. Especificação do trabalho

Para elaboração do trabalho, deve ser construído um programa que implementa um Analisador Léxico para a linguagem LALG. O programa deve respeitar a seguinte especificação:

- Implementar uma função que constrói a tabela de palavras reservadas. Essa função será invocada uma única vez no início da execução do compilador.
- Implementar o analisador léxico como uma função, pois depois será utilizada pelo analisador sintático. Sempre que chamada, a função retornará dois valores: uma cadeia lida e seu *token* correspondente. Os erros devem ser detectados, apropriadamente tratados e relatados ao usuário.
- Criar um programa principal que chama o analisador léxico várias vezes até que todo o programa-fonte tenha sido consumido. Deve-se criar um arquivo de saída com uma cadeia e seu *token* por linha.

Alguns pontos devem ser obrigatoriamente respeitados pelo grupo:

- Interface gráfica com o usuário (GUI) que deve permitir ao usuário selecionar arquivos e visualizar o código fonte a ser compilado. A interface também deve mostrar ao usuário os itens léxicos identificados e no caso de erros, relatar o defeito com mensagens apropriadas e indicar as linhas e colunas do arquivo fonte no qual os defeitos estão presentes.
- Usabilidade: intuitiva, simples, amigável.
- Modularização do código.
- Sub-rotinas para cada tarefa: criação da tabela de palavras reservadas, busca na tabela, reconhecimento de *tokens*, etc.
- A busca na tabela de palavras reservadas deve ser eficiente.
- Bons costumes de programação.
- Documentação do código.
- Variáveis e sub-rotinas com nomes significativos.
- Endentação.
- Não utilizar expressões regulares, implementar os autômatos.
- Tratar comentário de múltiplas linhas entre '{' e '}'.

Deverão ser entregues:

- Código fonte;
- Relatório sucinto e objetivo contendo:
  - Decisões de projeto e de implementação, com justificativas;
  - Projeto do analisador léxico, ou seja, os respectivos autômatos finitos;
  - Visão geral dos módulos e organização do analisador léxico;

- Passo a passo para compilar/interpretar o analisador léxico e executá-lo.

Exemplo:

- Programa LALG de entrada:

```
program lalg;
{entrada}
var a: integer;
begin
readd(a, @, 1);
end.
```

- Saída do Analisador Léxico:

```
program - program
lalg - id
; - ;
var - var
a - id
: - :
integer - integer
; - ;
begin - begin
readd - id
( - (
a - id
, - ,
@ - erro
, - ,
1 - num
) - )
; - ;
end - end
. - .
```

O programa deve fornecer o formato de saída completo seguinte:

item léxico	Identificador do símbolo	(linha, coluna) que ocorre o símbolo
-------------	--------------------------	--------------------------------------

## 4. Critérios de avaliação

Os seguintes critérios serão aplicados para avaliação do trabalho:

- Adequação a especificação do trabalho.
- Qualidade da solução para os algoritmos.
- Qualidade da solução para a interação com o usuário.
- Documentação do trabalho.

Os trabalhos serão considerados inaptos para correção, sendo atribuída a nota 0 (zero), se apresentarem os ao menos um dos seguintes problemas:

- Cópia total ou parcial de trabalho implica em zero para todos os envolvidos.
- Cópia total ou parcial de trabalho publicado em qualquer mídia, seja ela eletrônica ou não, implica em zero para todos os envolvidos.
- O projeto não compila.
- Os arquivos entregues do trabalho não podem ser lidos, por exemplo, estão corrompidos.
- O documento complementar não foi entregue ou não explica o que foi pedido no trabalho.