UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS CRATEÚS
DISCIPLINA: COMPILADORES
PROFESSORA: LISIEUX MARIE MARINHO DOS SANTOS ANDRADE

Projeto de Compiladores -Análise Léxica
Regras Gerais

O desenvolvimento do Analisador Léxico deve ocorrer com o uso da linguagem de programação Java, ficando o analisador responsável por criar a tabela de símbolos, onde cada entrada é formada pelos seguintes elementos:

- Token;
- Lexema;
- Linha correspondente a posição do token;
- Valor.

COMPONENTES E COMPORTAMENTOS

I. É vedado o uso de geradores automáticos nesta etapa;

II. Os seguintes elementos não devem ser considerados pelo analisador:
- Espaços em branco;
- Caracteres formatadores (tabulação, nova linha, novo parágrafo). Lembrar que alguns dos caracteres especiais servem para fazer a contagem da linha;
- O analisador léxico também não deve considerar comentários.

III. Os seguintes tipos de token devem ser considerados:
- Construção de escopo – Blocos de comandos (principal, condicional e de repetição);
- Palavras-chaves – Comandos próprios da linguagem;
- Identificadores: constantes e variáveis;
- Operadores: lógicos e aritméticos;
- Expressões: condicionais, laços de repetição, aritméticas e lógicas;
- Delimitadores: ; (ponto e vírgula) . (ponto final) : (dois pontos) ( ) (abre e fecha parênteses # (hashtag) @ (arrouba);
- Comando de Atribuição;
- Tipos de dados: primitivos da linguagem (inteiros, reais, lógicos e vazio), strings e vetores.

IV. Os seguintes erros devem ser detectados pelo analisador léxico:
- Comentário aberto e não fechado;
- Símbolos não pertencentes a linguagem.

DAS ETAPAS

A construção do Analisador Léxico dará por meio de dois momentos

I. Desenvolver relatório (documento no padrão de artigo da SBC) que apresente a construção dos tokens da linguagem abordada por meio de expressões regulares e de diagrama(s) de transição(ões);

II. Implementar o analisador léxico com uso da linguagem Java, permitindo ao usuário ambiente para a inclusão do código fonte (importação ou campo editor), assim como visualização do processo de análise (retorno da detecção de erro).

SEÇÃO DE EXEMPLOS

Programa Fonte:

```
program teste; {programa exemplo}
begin
42 + (675 * 31) – 20925
end;
```

Tabela de Símbolos:

Token		Lexema	Linha
Tipo	Valor		
Palavra reservada	PROGRAM	program	1
Identificador	ID	teste	1
Pontuacao	VIR	;	1
Palavra reservada	BEGIN	begin	2
Numero	42		3
Operador	SOMA		3
Pontuacao	PARESQ		3
Numero	675		3
Operador	MULT		3
Numero	31		3
Pontuacao	PARDIR		3
Operador	SUB		3
Numero	20925		3
Palavra reservada	END	end	4
Pontuacao	VIR	;	4

Interface do Compilador: