

Relatório TP1:

1. Analisador Léxico

1.1. Classes

TipoToken: Define constantes para os diferentes tipos de tokens.

Token: Armazena informações sobre cada token, como tipo, lexema e linha.

Lexico: Responsável por realizar a análise léxica.

Métodos da Classe Token:

- **__init__(self, tipo, lexema, linha):** Inicializa um objeto Token com tipo, lexema e linha.

Métodos da Classe Lexico:

- **abreArquivo(self):** Abre o arquivo fonte para leitura.
- **fechaArquivo(self):** Fecha o arquivo fonte.
- **is_alpha(self, car):** Verifica se o caractere é uma letra.
- **is_digit(self, car):** Verifica se o caractere é um dígito.
- **is_alnum(self, car):** Verifica se o caractere é alfanumérico.
- **getChar(self):** Obtém o próximo caractere do arquivo.
- **ungetChar(self, c):** Devolve o último caractere lido para o arquivo.
- **getToken(self):** Obtém o próximo token do arquivo, implementando as regras léxicas definidas.

1.2. Código Principal

O código do analisador léxico implementa as regras de reconhecimento de tokens para a linguagem especificada. A linguagem contém palavras reservadas, identificadores, números inteiros e reais, operadores e delimitadores.

O analisador léxico é responsável por ler o arquivo de entrada, identificar os tokens presentes no código fonte e gerar instâncias da classe Token para cada ocorrência.

A classe TipoToken define constantes para os diferentes tipos de tokens, enquanto a classe Token armazena informações sobre cada token, como tipo, lexema e linha.

O analisador léxico também contém a classe Lexico, que implementa métodos para abrir e fechar arquivos, ler caracteres, obter tokens e lidar com palavras reservadas.

2. Analisador Sintático

2.1. Classe

Sintatico: Realiza a análise sintática do código fonte.

Métodos da Classe Sintatico:

- **__init__(self):** Inicializa um objeto Sintatico.
- **interprete(self, nomeArquivo):** Inicia a interpretação do arquivo.
- **atuallqual(self, token):** Verifica se o token atual é igual ao token esperado.
- **consume(self, token):** Consome o token atual se for igual ao token esperado, caso contrário, gera uma mensagem de erro.
- Métodos correspondentes a cada não terminal da gramática, como PROG, DECLS, LIST_DECLS, etc.

2.2. Código Principal

O código do analisador sintático implementa as regras gramaticais da linguagem, utilizando recursão descendente para cada não terminal. As regras são definidas de acordo com a gramática especificada.

O analisador sintático realiza a análise da estrutura do código fonte, verificando se ele está de acordo com as regras da linguagem. Em caso de desvio nas regras, são geradas mensagens de erro indicando a linha e o problema encontrado.

A classe Sintatico contém métodos correspondentes a cada não terminal da gramática, e o ponto de entrada é o método PROG, que representa a regra inicial. Cada método corresponde a uma produção da gramática.

O código faz uso da classe Lexico para obter os tokens necessários durante a análise sintática.