

Relatório TP2:

1. Classe Tabela

A classe TabelaSimbolos foi desenvolvida com o propósito de gerenciar e armazenar informações sobre identificadores em um contexto de análise léxica e sintática. Este relatório visa descrever as funcionalidades principais da classe e fornecer uma análise crítica de sua implementação.

existIdent(self, nome): verifica se um identificador já está presente na tabela. Esta função é crucial para evitar a declaração duplicada de identificadores. No entanto, a implementação poderia ser simplificada utilizando diretamente o resultado da verificação (ou seja, `return nome in self.tabela`).

declarIdent(self, nomes, valor, linha): é responsável por adicionar identificadores à tabela, juntamente com seus valores associados. Este método realiza a verificação de existência antes de adicionar um identificador, para garantir a integridade da tabela de símbolos.

2. Analisador Sintático

2.1. Mudanças para criação da tabela de símbolos:

A função DECL_TIPO agora cria uma lista para armazenar ids declarados, que são adicionados na função LIST_ID. A função TIPO retornar o tipo dos ids na lista e logo após chamamos a função declarIdent para salvar ids e tipo no dicionario da tabela.

```
#DECL_TIPO → LIST_ID dpontos TIPO pvirg
def DECL_TIPO(self):
    nomes=[] #lista de para salvar ids
    self.LIST_ID(nomes)
    self.consome(tt.DPONTOS)
    valor =self.TIPO() #salva o tipo
    self.tabsimb.declaraIdent(nomes, valor, self.tokenAtual.linha) #declara os ids na tabela de simbolos
    self.consome(tt.PVIRG)

#LIST_ID → id E
def LIST_ID(self, nomes=[]):
    if self.tokenEsperadoEncontrado(tt.ID): #se for um id
        nomes.append(self.tokenAtual.lexema) #adiciona na lista de ids
    self.consome( tt.ID )
    self.E(nomes)
```