Relatório Descritivo - Etapa 5

Fábio de Azevedo Gomes - Cartão 00287696
Lucas Lauck dos Passo 00285688

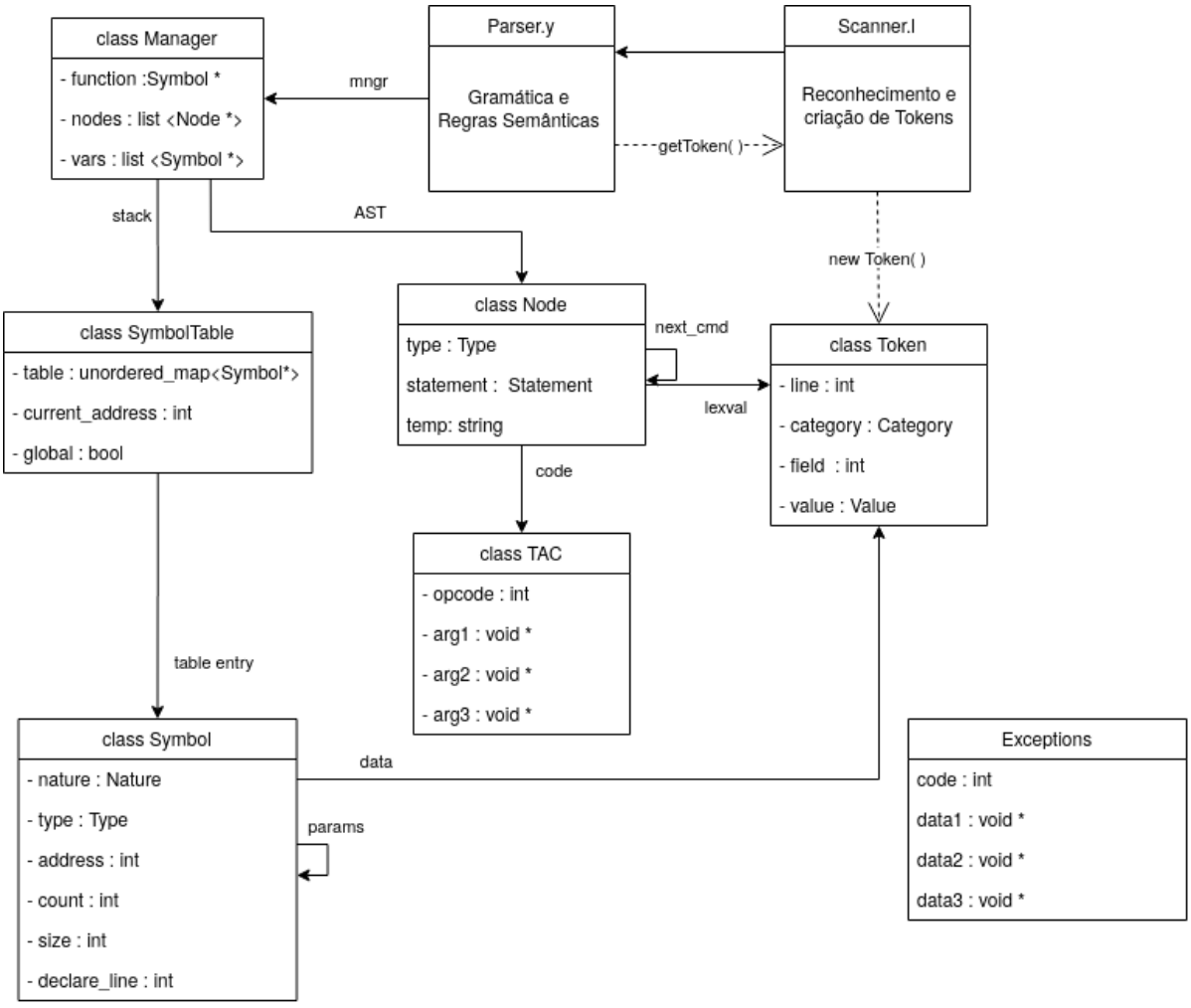
November 2020

1 Introdução

Este relatório foi escrito para esclarecer as mudanças no código com relação à entrega da etapa 4, dado que ele foi migrado para a linguagem C++ de forma a facilitar a modelagem das diferentes estruturas de dados requisitadas. Também foi criado um diagrama em "Pseudo-UML" para facilitar a compreensão da sua arquitetura, em conjunto com uma breve descrição de cada módulo.

Conforme requisitado na especificação, deixo registrado que a opção escolhida para a geração de código intermediário foi em **uma passagem**, isto é, junto ao processo das análises sintática e semântica e geração da `AST`.

2 Diagrama Pseudo-UML



3 Classes ou Módulos Implementados

3.1 Classe Manager

Contém as funções usadas pelo parser para manipulação das estruturas de dados do compilador de maneira mais "alto nível", como

- Entrar e sair de escopo
- Criar nodos para comandos na AST
- Verificar tipo e uso de nodos gerados
- Declarar símbolos na tabela de símbolos local ou global
- Procurar por símbolos na pilha de tabelas

3.2 Classe SymbolTable

Modela uma tabela de símbolos de um escopo estático. Provê funções de inserção e consulta a símbolos, e mantém os endereços de memória para seus símbolos. Utiliza um `unordered_set` que mapeia de nomes para objetos `Symbol`.

3.3 Classe Symbol

Modela um símbolo do programa, com sua linga de declaração, tipo, entre outros. Caso o símbolo seja de uma função, mantém uma lista ordenada de símbolos com os parâmetros esperados.

3.4 Classe Token

Modela um *Token* obtido pelo scanner (Antigo `valor_lexico`) e enviado para o parser. Provê métodos para acesso a esses dados.

3.5 Classe Node

Modela um nodo da `AST`, com seu `Token` correspondente, seus tipo, filhos, próximo nodo, código `TAC` gerado e seu registrador temporário. Provê funções para criação do nodo de forma genérica, e métodos específicos para geração de código `ILOC`.

3.6 Classe TAC

Modela uma instrução do código de três endereços, de acordo com a sintaxe `ILOC`. Mantém as informações sobre seu código de instrução e operandos. Também oferece funções para exportação do código no formato `string`.

3.7 Módulo Exceptions

Modela erros semânticos que estar presentes na entrada, e provê métodos para apresentá-los de maneira legível ao usuário.