

	Curso	Ciência da Computação (Híbrido)
	Atividade Acadêmica	Tradutores
	Ano/Semestre	2024/2
	Professora	Maria Adelina Raupp Sganzerla
	Data	31/08/2024
	Atividade Vivencial 1 - Grau B 1.5 pontos	

Observações:

- O trabalho pode ser realizado em trios, duplas ou individual (trabalhar em equipe fortalece as discussões, estudos e pontos de vista).
- Será executado em quatro etapas (durante as atividades vivenciais), uma etapa será entregue, no final todo o trabalho deverá ser apresentado em sua íntegra (Grau B).
- Utilize as ferramentas escolhidas para a geração dos analisadores, a grande maioria possui uma ferramenta para o analisador léxico e outra para o sintático, observe a documentação.
- Cada ferramenta possui uma sintaxe e uma forma de executar os analisadores léxicos e sintáticos, portanto é importante ler e estudar a documentação, para que possam entender o que está sendo implementado.
- As entradas e saídas deverão seguir o padrão da ferramenta, ou seja, por arquivo ou com console.
- A Gramática de cada etapa, deverá ser entregue em um arquivo texto editável, junto com as especificações (quantas casas o float/real assume, quantos caracteres o char e outras informações necessárias para o entendimento).
- Compacte todos os arquivos gerados em uma única pasta para enviar, pois cada ferramenta é diferenciada.
- As etapas devem ser enviadas em separado, o conjunto completo será entregue no Grau B. Precisando de alguma correção extra, deve ser efetuada por e-mail: msganzerla@unisinios.br ou pelos nossos canais de comunicação.
- As datas de entregas e pesos, estão organizados/descritos nas etapas abaixo.
- A não entrega de uma etapa não invalida o trabalho, porém a mesma deverá ser apresentada na etapa final e terá pontuação diferenciada.
- Organize-se e Bom Trabalho!!!

Etapa 1 (Atividade Vivencial 1 - 31/08/2024)

Apresente a Gramática e os analisadores léxicos e sintáticos para o reconhecimento de declaração de variáveis no estilo (inspiradas) da Linguagem C (leia com atenção as especificações abaixo, não deve ser implementado nem mais, nem menos).

- Devem reconhecer os tipos: char – int – float.
- O char pode ser um caractere ou uma cadeia (se a ferramenta possui restrição de quantidade, deve ser informada), por exemplo `char a; char a[10];`.
- O int compreende todos os valores inteiros (declarar se a ferramenta possui alguma restrição de valores reconhecidos).
- O float compreende todos os valores reais, ou seja, com casas decimais (ponto flutuante) (declarar se a ferramenta possui alguma restrição de valores reconhecidos).
- As variáveis são apenas declaradas, não sendo possível inicializá-las.
- Identificadores: de acordo com as regras da Linguagem C (iniciam por letras ou '_', depois do segundo caractere pode ser número, letra ou '_' e o único caractere especial reconhecido é o '_'), lembre-se que a palavra vazia (ε) não pode ser considerada como identificador.
- Podem ser declarados individualmente ou por uma lista (sendo lista separados por vírgulas), por exemplo: `int a, b, c;`.
- A finalização de cada declaração será por ponto-e-vírgula (;).

Data de entrega: 04/09/2024 (quarta-feira), pelo Moodle, na seção da Atividade Vivencial (Módulo 3), Tradutores (Etapa Vivencial 1).

Peso: 1.5

O que entregar nesta etapa: Arquivo com as especificações e gramática (documento) + codificação da ferramenta (todos os arquivos para execução)

Etapa 2 (Atividade Vivencial 2 - 14/09/2024 - manhã)

Etapa 3 (Grau B - 28/09/2024)

Etapa 1

Eduarda Vargas dos Santos, Guilherme Mathias Dörr e Leonardo Agustoni Feilke

Definição de gramática:

Alfabeto completo: {int, float, char, [,], :, ., ", A-Z, a-z, _}

Tipos: {int, float, char}

Identificadores: {0-9, A-Z, a-z}

Caracteres especiais: { [,], :, ., ", _ }

Instruções para executar o código estão dentro do Arquivo README.md

Criação da gramática no ANTLR4:

```
grammar Expr;

CHAR_TYPE: 'char';
INT_TYPE: 'int';
FLOAT_TYPE: 'float';
IDENTIFIER: [a-zA-Z_][a-zA-Z_0-9]*;
SEMICOLON: ';';
COMMA: ',';
WS: [ \t\r\n]+ -> skip;
IDENTIFIER_LIST: IDENTIFIER (COMMA IDENTIFIER)*; // a, b, c;
ARRAY_DECLARATION: '[' [0-9]+ ']'; // [1123];

declaration: (
    char_declaration
    | int_declaration
    | float_declaration
) SEMICOLON;

char_declaration:
    CHAR_TYPE IDENTIFIER (ARRAY_DECLARATION)? (
        COMMA IDENTIFIER (ARRAY_DECLARATION)?
    )*; // char a, b[10], c;

int_declaration: INT_TYPE IDENTIFIER_LIST; // int a2, b3, c4;
float_declaration: FLOAT_TYPE IDENTIFIER_LIST; // float a1, b2;
```