

Trabalho Final da Disciplina
-Diretrizes-
Paradigmas de Linguagem de Programação - 2023.1
Professor: Josefran Bastos

1 Orientação Geral

Cada equipe irá receber uma linguagem de programação para realizar o que se pede no trabalho. De forma geral, o trabalho consistirá em três partes, descritas a seguir e detalhes mais adiante.

- Na primeira parte a equipe irá fazer um breve resumo histórico da linguagem juntamente com algumas curiosidades e inovações apresentadas pela linguagem.
- Na segunda parte a equipe irá implementar um analisador léxico/sintático para a linguagem. Dado um arquivo .txt, o analisador deverá funcionar de forma similar ao apresentado no Capítulo 4 do livro e retornar a árvore parser do código em questão.
- Na terceira parte, a equipe irá analisar características da linguagem relacionadas aos conteúdos apresentados do Capítulo 5 até a Seção 6.4 do Capítulo 6.

Nas Seções 2, 3 e 4 apresentamos alguns detalhes adicionais para facilitar o desenvolvimento do trabalho pelas equipes. Na Seção 5, apresentamos como, e por onde, o trabalho deverá ser entregue. Finalmente, na Seção 6, apresentamos algumas diretrizes finais do trabalho juntamente com parte dos critérios que serão utilizados para a correção do trabalho.

2 Primeira Parte - Análise Histórica

Acredito que o resumo apresentado na seção anterior já deixe bem claro o que é necessário apresentar nesta parte do trabalho. Esta parte se assemelha muito com o primeiro ciclo das palestras, portanto o material utilizado pelas equipes também poderá como fonte bibliográfica.

Vale ressaltar que não estou interessado em um curso sobre programação na linguagem em questão. Então não foquem tanto ensinando a programar, a menos que tenha algo interessante que vocês acham que devam apresentar.

3 Segunda Parte - Analisador Léxico/Sintático e Árvore Parser

Como a linguagem é muito vasta, desenvolver essa parte pode ser potencialmente extensa e cansativa, que não é o foco desse trabalho. Então o seu analisador deve se restringir a programas na linguagem que envolvam

- Ler informações do teclado;

- Imprimir informações no terminal;
- Realizar operações aritméticas básicas (somar, subtrair, dividir e multiplicar);
- Conseguir computar expressões aritméticas (ex: $2*3 + 1$).

O teste dessa parte será feito por meio de entradas de programas simples como “HelloWorld”, “Soma” e “CalcularExpressao”. O seu programa deverá identificar os lexemas, tokens, etc. Por fim, deverá apresentar a árvore parser para o programa apresentado. Note aqui que será necessário apresentar as gramáticas para as rotinas acima.

4 Terceira Parte - Debulhando a Linguagem

Nesta parte, a equipe deverá identificar os elementos da construção da linguagem. Como fomos até a Seção 6.4, então apenas os elementos até este ponto serão cobrados. Cabe a equipe revisitar os capítulos 5 e 6 e pesquisar cada elemento em relação à linguagem.

Por exemplo, em relação à Seção 5.2 que se refere a *nomes*. Apresentar se a linguagem é case-sensitive, algumas (mas não todas) palavras reservadas, se existe algum padrão específico para declaração de nomes, qual tamanho máximo, etc. Em geral, todas essas informações podem ser encontradas em documentações sobre a linguagem. Caso ache necessário, poderá apresentar exemplos na linguagem.

5 Sobre a Entrega - Prazo e Formato

5.1 Data de Entrega e Penalização

A data de entrega será até as 23:59 do dia 10/07/2023 e deverá ser enviada para o e-mail josefran@ufc.br seguindo o formato descrito na próxima seção. Os trabalhos entregues após a data limite será penalizado em 5 pontos até 24h horas de atraso e 10 pontos após 24h de atraso. Ou seja, um dia de atraso o trabalho vale metade da pontuação e a partir disso é zerado.

5.2 Formato de Entrega

O trabalho deve ser entregue em formato zip contendo os elementos abaixo listado.

1. Um arquivo PDF contendo as três partes acima explicitada. O arquivo ainda deve conter introdução e referências bibliográficas.
2. Um arquivo zip contendo os compiláveis e alguns casos teste. Caso tenha alguma especificação sobre a compilação o mesmo deverá ser adicionado em um arquivo chamado README.
3. Um arquivo README explicando de forma sucinta como executar o programa a fim de realizar meus próprios testes.
4. Um executável para caso eu tenha problemas de compilação.

6 Da Avaliação

O trabalho valerá nota entre 0 e 10. Além do conteúdo analisado, será avaliado a clareza e organização do trabalho. Trabalhos mal organizados, ou com pouco clareza não serão corrigidos. Dentre as características que podem afetar sua nota, temos

- Falta de organização;
- Código mal comentado;
- Código que não funciona;
- Trechos de texto copiados diretamente da internet ou de outros trabalhos (SERÁ USADO UMA FERRAMENTA ANTI-PLÁGIO). Para este caso, a nota do trabalho inteira será ZERADA;
- Deixar de fazer o que foi solicitado no trabalho;
- Texto sem fluidez;
- Inventar informações sem apresentar as devidas referências.

Algumas outras características poderão ser consideradas dependendo da particularidade de cada trabalho.