

1 Instruções Importantes

Nesta seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Escolha da Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem Java e pode ser utilizada qualquer ferramenta para gerar analisadores léxico e sintático.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do programa;
- arquivos de especificação léxica e sintática das ferramentas, se houverem;
- arquivo de *build* para geração automática dos analisadores e compilação do fonte (makefile, ant, shell script, etc);
- relatório do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código compila e executa em ambiente Unix / Linux. Programas que não compilam receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do(s) autor(es) do trabalho;
3. arquivo do relatório tenha a identificação do(s) autor(es) do trabalho;
4. arquivo compactado com os artefatos estão devidamente identificados com nome e matrícula.

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, relatório e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: corretude do programa, estrutura do código, redigibilidade e legibilidade. A corretude se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

O relatório deve conter informações relevantes para compilar, executar e auxiliar no entendimento da estratégia adotada para resolução do problema e do código fonte. Ressalta-se que o relatório não deve conter cópias do fonte – afinal o seu fonte é um dos artefatos enviado –, porém trechos de códigos podem ser incluídos caso facilite a explicação e elucidação das estratégias implementadas. O relatório deve apresentar as decisões de projetos tomadas: escolha de ferramentas, breve descrição das ferramentas utilizadas, estratégia de implementação do analisador sintático (LL, LR, PEG e separação entre léxico e sintático), estratégia

de interpretação, estruturas de dados usadas, representação da árvore de sintaxe abstrata, dentre outras informações.

O trabalho deverá ser apresentado ao professor da disciplina e, só será avaliado após a realização da entrevista, i.e., trabalhos que não forem apresentados não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema, os resultados e conclusões obtidas. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, relatório e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina após a entrega do trabalho.**

Atrasos serão penalizados por uma função exponencial de dias de atrasos, i.e., será reduzido do percentual da nota a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Dias de Atraso	Nota
1	$n \cdot 0.98$
2	$n \cdot 0.96$
3	$n \cdot 0.92$
4	$n \cdot 0.84$
5	$n \cdot 0.68$
6	$n \cdot 0.36$
7	0

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

2 Especificação Técnica do Trabalho

Ao longo do semestre será construído um compilador para a linguagem *lang*. Assim, após realizar a implementação do analisador léxico para a linguagem, o próximo passo é construir um analisador sintático. Consequentemente, uma das tarefas a serem realizadas nesse trabalho é a implementação de um analisador sintático para a linguagem *lang*.

Uma classe principal que chamará o analisador sintático é fornecida, denominada *lang.LangCompiler*. Essa classe usa uma interface, *parser.ParseAdaptor*, que encapsula o analisador sintático. Assim, você deverá implementar um subtipo de *parser.ParseAdaptor*, a qual irá receber o caminho do arquivo de entrada e chamar o seu analisador sintático. Ressalta-se que os códigos fornecidos não devem ser modificados, excetuando-se a classe *lang.LangCompiler*, a qual você deverá modificar para usar as classes desenvolvidas.

Após a implementação do analisador sintático, a próxima etapa na implementação da linguagem *lang* será a interpretação. Assim, após finalizar a implementação do analisador sintático, deve ser feito um interpretador para a linguagem, i.e., a partir da AST produzida pelo analisador sintático, você deverá interpretar a linguagem. Ressalta-se que existem programas que estão sintaticamente corretos, mas não tem significado bem definidos. Nessas circunstâncias, o seu interpretador deverá lançar uma exceção e finalizar a execução do programa.

Em suma, há duas tarefas a serem realizadas neste trabalho: 1) implementação de um analisador sintático; 2) implementação de um interpretador da linguagem.

3 Entrega do Trabalho e Apresentação

A data da entrega do trabalho será até o dia **25 de agosto de 2024**, via plataforma do GoogleClassroom. O trabalho deverá ser apresentado ao professor da disciplina até o dia 15 de setembro de 2024.

4 Pontos Extras

Os trabalhos que forem implementados usando algum analisador PEG serão bonificados com dois pontos adicionais.