

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
Disciplina INF01121 – Modelos de Linguagens de Programação – Semestre 2011/2

Prof. Leandro Krug Wives

Instruções para o Trabalho Final

O **objetivo** deste trabalho consiste em fornecer aos alunos a oportunidade de conhecer uma linguagem moderna e que agregue características multiparadigma, combinando orientação a objetos com características funcionais. Permitirá ainda com que os alunos demonstrem que aprenderam os princípios de programação relacionados com os diferentes paradigmas estudados, além de oportunizar demonstrarem que são capazes de analisar e avaliar linguagens de programação seguindo os critérios estudados em aula.

- 1) Este trabalho pode ser realizado **em grupos de três pessoas (preferencialmente)**.
- 2) Cada grupo deve **selecionar uma das seguintes opções**: Common Lisp, C++ 11, D (2.0), Dart, F#, Java 7, JavaScript, Objective Caml, Perl, PHP, Python, Ruby, Scala. Tais linguagens foram selecionadas por apresentarem características funcionais e orientadas a objeto. **Podem ser utilizadas outras linguagens, desde que possuam princípios de programação orientada a objetos combinados com programação funcional** (nesse caso, antes de escolher, pergunte ao professor).
- 3) Cada grupo deve **selecionar um problema a ser resolvido com a linguagem**. O objetivo é demonstrar como as funcionalidades orientadas a objeto e funcionais da linguagem foram usadas na solução do problema. Sugere-se os seguintes problemas:
 - a) Implementação de um jogo do tipo Tower Defence (http://en.wikipedia.org/wiki/Tower_defense/);
 - b) Implementação de uma variação do jogo Risk ([http://en.wikipedia.org/wiki/Risk_\(game\)/](http://en.wikipedia.org/wiki/Risk_(game)/)) ou a versão Brasileira War (<http://pt.wikipedia.org/wiki/War>). A fim de tornar o jogo menos complexo e menos demorado, sua versão pode envolver somente dois adversários (seja outro ser humano ou o computador). Preferencialmente, o jogo pode utilizar a API do Google Maps para posicionar os exércitos.
 - c) Simulador de determinação de escopo (dinâmico versus estático). Ou seja, desenvolver um simulador capaz de aceitar definições de subprogramas e variáveis locais, utilizando uma pseudolinguagem simples. Com base nisso, demonstrar como ficaria sua pilha de chamadas (*call-stack*) e o conteúdo das variáveis locais a cada passo de execução.
 - d) Escolha de um problema pessoal ou motivador. No caso, o grupo deve encaminhar sua ideia ao professor, descrita em detalhes, que avaliará sua viabilidade.
- 4) Cada grupo deve **preencher o Wiki sobre tópicos e grupos do trabalho final, disponível no Moodle, definindo os componentes do grupo e o assunto do trabalho**. Essa definição deve ser feita **até o dia 30 de Abril**.
- 5) Os grupos deverão **entregar um relatório de trabalho, pelo Moodle, até o dia definido no Cronograma da disciplina disponibilizado no Moodle**. Atrasos implicarão em perda de nota.
- 6) **O relatório de trabalho deverá conter os itens descritos abaixo**:
 - a) Capa;
 - b) Apresentação do problema solucionado, descrevendo quais são os requisitos e as funcionalidades implementadas. Apresentação dos frameworks, bibliotecas e ferramentas utilizados no desenvolvimento.
 - c) Apresentação da linguagem, descrevendo suas características, fundamentos, funcionalidades, aplicações (aplicabilidade em questões práticas), benefícios, problemas, limitações;
 - d) Detalhamento da implementação, focando na descrição de como os seguintes elementos foram usados para resolver o problema escolhido:
 - Classes, atributos e métodos;
 - Encapsulamentos (proteção, separação de interface da implementação)
 - Composição, herança
 - Polimorfismo por inclusão e paramétrico
 - Funções como elemento de 1ª ordem
 - Funções de ordem maior (map, reduce, foldr/foldl...)
 - Manipulação de listas através de funções puras e recursão
 - Currying (ou funções não nomeadas, ou lambda, com mais de um parâmetro)
 - Casamento de padrões
 - e) Análise crítica da linguagem, envolvendo uma tabela com os critérios e propriedades estudados em aula, com notas/valores justificados.
 - f) Conclusões, descrevendo ainda as dificuldades encontradas;
 - g) Bibliografia (devem seguir as normas da ABNT).

- h) Código-fonte (não é necessário incluir executáveis e outros arquivos). Não é necessário incluir o código-fonte no arquivo do relatório. Você pode gerar um arquivo PDF do relatório e combinar ele com os arquivos-fonte em um único arquivo .zip ou .rar e encaminhar este arquivo compactado via Moodle.
- 7) Os **resultados** dos trabalhos **serão apresentados ao professor**, em data definida no cronograma da disciplina. Sugere-se que os alunos usem uma máquina Virtual para a implementação de suas soluções, visto que todos os computadores possuem o Virtual PC da Microsoft ou o Virtual Box da Sun, facilitando o transporte e demonstração da aplicação resultante.

Avaliação

A avaliação do trabalho **incluirá os seguintes critérios**: desenvolvimento e detalhamento dos itens do relatório, aplicação dos conceitos de programação estudados, utilização correta dos recursos da linguagem escolhida, correção, legibilidade, confiabilidade e originalidade, uso de referências, formatação e estilo do texto (conforme ABNT). Outros aspectos de avaliação poderão ser incluídos a critério do professor.

Conforme descrito no Plano da Disciplina, disponível no Moodle Institucional ([HTTP://moodleinstitucional.ufrgs.br](http://moodleinstitucional.ufrgs.br)), este trabalho correspondente a 25% da nota final.

Em caso de dúvidas, consulte o professor. Bom trabalho!