

Trabalhando com Linguagens de Programação Imperativas “Não-Convencionais”

Prof. Rodrigo Hübner

Abstract

Este trabalho prático tem por objetivo a prática de implementação de um programa em linguagens de programação imperativas não tão convencionais. Para este fim, é proposto o uso das linguagens de programação Lua, Ruby, Julia, Go, Swift, Groovy, R, C# e Ada.

1. Objetivo

Este trabalho tem como objetivo conhecer e aprender novas práticas de diversas linguagens de programação [1, 2, 3, 4, 5, 6, 7, 8, 9] não comumente utilizadas, com a implementação de alguns problemas ou contribuições pesquisadas pelos alunos.

As próximas seções descrevem (2) o trabalho proposto, (3) a avaliação, (4) questões importantes e, por fim, (5) a conclusão.

2. O Trabalho

Algoritmo Huffman de Compactação

Escreva um codificador/decodificar que use a técnica de codificação do algoritmo de Huffman.

A especificação do algoritmo de compactação de Huffman pode ser facilmente encontrada internet. Uma dessas especificações estão na parte de “materiais diversos” no moodle.

Notas:

- Deverá ser implementado apenas um programa;
- A escolha entre o codificador e o decodificador deverá ocorrer através de parâmetros;

- A entrada de dados deverá ser por arquivo;
- O arquivo deverá ter tamanho arbitrário.

Desenvolvimento de varredura de informações web

Desenvolvimento de um serviço que realiza uma varredura de informações específicas de uma página web. O serviço deve utilizar alguma tecnologia que processa corretamente arquivos web (HTML), tais como **XPATH**, **XML-DOM**, etc. A página web será disponibilizada pelo professor juntamente com as informações que deverão ser extraídas.

Notas:

- Não serão aceitos varreduras realizadas com abertura/edição comum de arquivos (p.e. primitivas, **open**, **write**, das bibliotecas padrões da linguagem em questão). Um serviço adequado deverá ser utilizado.
- A informação extraída deverá ser mostrada formatada no terminal.

3. Avaliação do Trabalho

O trabalho será avaliado de acordo com os seguintes critérios:

Codificação

Será avaliado a modularização e a elegância do código fonte, além de estar de acordo com os critérios que são discutidos em sala de aula.

Execução

Será avaliado se os programas fazem realmente o que foi solicitado (ou apresentado ao professor).

4. Questões Importantes

- **Cópias:** Qualquer tipo de cópia (trabalhos de colegas, internet, etc) **anulará** o trabalho, seja porções de código ou simplesmente o trabalho completo;
- **Entrega:** Deverá ser entregue o código fonte do trabalho em um arquivo compactado, com o nome do aluno no moodle;

- **Data:** O trabalho deverá ser entregue no dia 06/04/2016 até o horário da aula em que será apresentado;
- **Valor:** 2,0 pontos;

5. Conclusão

Este trabalho tem por objetivo implementar programas em linguagens de programação imperativas não convencionais. No final do trabalho, é esperado que cada aluno tenha aprimorado suas habilidades de programação e conheça a sintaxe geral de diversas linguagens de programação imperativas.

Referências

Lua, Ruby, Julia, Go, Swift, Groovy, R, C# e Ada.

- [1] <http://www.lua.org/>
- [2] <https://www.ruby-lang.org/pt/>
- [3] <http://julialang.org/>
- [4] <https://golang.org/>
- [5] <https://swift.org/>
- [6] <http://www.groovy-lang.org/>
- [7] <https://www.r-project.org/>
- [8] <http://www.mono-project.com/docs/about-mono/languages/csharp/>
- [9] <http://www.adacore.com/>