INF1301 Programação Modular
Período: 2013.2
Prof. Flavio Bevilacqua

2o. Trabalho

Data de divulgação: 16 de setembro (segunda-feira)

Data de entrega: 9 de outubro (quarta-feira)

1. Descrição do trabalho do semestre

Neste semestre desenvolveremos um módulo reconhecedor de xeque-mate em um tabuleiro de Xadrez. O tabuleiro será implementado utilizando um grafo dirigido. Cada vértice corresponderá a uma casa do tabuleiro. Cada peça será instanciada em uma estrutura lista de listas que armazenará os diversos movimentos possíveis desta peça. Este Xadrez poderá utilizar peças com movimentos personalizados pelo usuário. Utilizando o guia de movimentos de cada peça implementado nas listas de listas, o programa desenvolvido aqui permitirá a disposição de peças no tabuleiro e a posterior verificação se o rei está em xeque mate ou ainda é possível andar para uma casa. Neste trabalho não ocorrerão partidas de xadrez. Basta dispor as peças e verificar se o rei está em xeque mate.

O modelo básico de um grafo dirigido implementado usando listas de arestas pode ser visto na figura 1. É **obrigatória** a implementação do grafo utilizando a estrutura abaixo e instanciando estruturas do módulo Lista Genérica.

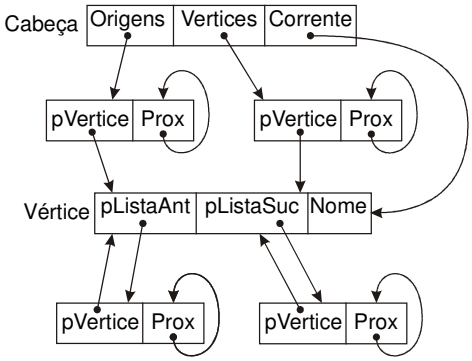

Figura 1. Modelo físico de um grafo dirigido

2. Descrição do segundo trabalho

Neste trabalho deverá ser criado o módulo **GRAFO** genérico. Os vértices deste grafo apontam para valores do tipo **void**. Os elementos das listas de antecessores e sucessores também apontam para valores do tipo **void**. Deve ser desenvolvido um módulo **LISTA** que implementa uma lista genérica. Este módulo deve ser utilizado neste trabalho.

Procure seguir o seguinte roteiro para realizar o trabalho:

1. Crie um diretório para elaborar o projeto do reconhecedor de xeque mate. Siga a estrutura de diretórios dos exemplos contidos no pacote do arcabouço de apoio ao teste.
2. Faça todos os ajustes de modo que o módulo **LISTA** do arcabouço possa ser utilizado em seu trabalho, compile e teste completamente este módulo.
3. Projete o módulo **GRAFO**. Reformule o modelo da figura 1 e redija ***todas*** as assertivas estruturais. O modelo e as assertivas deverão constar do arquivo **zip** de entrega do trabalho e deverão estar em conformidade com o grafo utilizado para teste do módulo.

4. Implemente e teste o módulo **VERTICE**. Para efeito de teste, os vértices devem referenciar *strings*.
5. Implemente e teste o módulo **GRAFO** compondo-o com os módulos **VERTICE** e **LISTA**. Para efeito de teste as listas de sucessores devem referenciar rótulos do tipo *string*, tal como está no módulo de teste fornecido junto com o módulo **LISTA**. As listas de antecessores não devem referenciar rótulos.

Obs: deverá ser possível percorrer o grafo, inserir e excluir vértices, inserir e excluir arestas, obter valor do vértice corrente, alterar valor do vértice corrente, e demais operações que o grupo achar necessárias.

3. Entrega do Trabalho

O trabalho deve ser feito em grupos de dois ou três alunos. Os programas devem ser redigidos em "**C**". Não será aceita nenhuma outra linguagem de programação. Todos os programas devem estar em conformidade com os padrões dos apêndices de 1 a 10 do livro-texto da disciplina. Em particular, os módulos e funções devem estar devidamente especificados.

Recomenda-se fortemente a leitura do exemplo e do texto explanatório do arcabouço. Além de mostrar como implementar um teste automatizado dirigido por *script*, ilustra também as características de um programa desenvolvido conforme os padrões do livro.

O trabalho deve ser enviado por e-mail em um único arquivo **.zip** (codificação do *attachment*: **MIME**). Veja os *Critérios de Correção de Trabalhos* contidos na página da disciplina para uma explicação de como entregar.

O arquivo **.zip** deverá conter:

- Os arquivos-fonte dos diversos módulos que compõem os programas.
- Os arquivos de *script* de teste desenvolvidos pelo grupo.
- Os arquivos *batch* (**.bat**) que coordenam a execução dos testes.
- Os diversos programas executáveis (construtos): **TRAB2-1.EXE**, **TRAB2-2.EXE**, **TRAB2-3.EXE**, e assim por diante. Repare que o teste da lista sozinho gera um executável, o teste do vértice gera outro executável, o teste do grafo gera outro, e assim por diante.
- O modelo físico do grafo, um exemplo e as correspondentes assertivas estruturais.
- Um arquivo **LEIAME.TXT** contendo a explicação de como utilizar o(s) programa(s).
- Tantos arquivos **RELATORIO-nome.TXT** quantos forem os membros do grupo. O tema **nome** deve identificar o membro do grupo ao qual se refere o relatório. Estes arquivos devem conter uma tabela de registro de trabalho organizada como a seguir:

Data Horas Trabalhadas, Tipo Tarefa, Descrição da Tarefa Realizada

Na descrição da tarefa redija uma explicação breve sobre o que o componente do grupo fez. Esta descrição deve estar de acordo com o Tipo Tarefa. Cada Tipo Tarefa identifica uma natureza de atividade que deverá ser discriminada explicitamente, mesmo que, durante uma mesma sessão de trabalho tenham sido realizadas diversas tarefas. Os tipos de tarefa são:

- ♦ estudar
- ♦ especificar os módulos
- ♦ especificar as funções
- ♦ revisar especificações
- ♦ projetar
- ♦ revisar projetos
- ♦ codificar módulo
- ♦ revisar código do módulo

- redigir script de teste
- revisar script de teste
- realizar os testes
- diagnosticar e corrigir os problemas encontrados

Observações:

- **Dica**: Preencha esta tabela de atividades ao longo do processo. NÃO DEIXE PARA ÚLTIMA HORA, POIS VOCÊ NÃO SE LEMBRARÁ DO QUE FEZ TAL DIA, TAL HORA. Com relatórios similares a esse você aprende a planejar o seu trabalho.
- **Importante**: O arquivo **ZIP**, DEVERÁ CONTER SOMENTE OS ARQUIVOS RELACIONADOS A ESTE TRABALHO. Caso o arquivo enviado contenha outros arquivos que os acima enumerados (por exemplo: toda pasta do arcabouço, arquivos **.bak**, arquivos de trabalho criados pelo ambiente de desenvolvimento usado, etc.) o grupo **perderá 2 pontos**. Gaste um pouco de tempo criando um *diretório de distribuição* e um **.bat** que copia do *diretório de desenvolvimento* para este diretório de distribuição somente os arquivos que interessam. Verifique se esta cópia está realmente completa!
- A mensagem de encaminhamento deve ter o assunto (*subject*) **INF1301-Trab02-idGrupo** conforme o caso. O tema **idGrupo** deve ser formado pelas iniciais dos nomes dos membros do grupo. O texto da mensagem deve conter somente a lista de alunos que compõem o grupo (formato: número de matrícula, nome e endereço do e-mail). **Perde-se 2 pontos caso não seja encaminhado desta forma**. Mais detalhes podem ser encontrados no documento *Critérios de Correção dos Trabalhos* disponível na página da disciplina.
- O programa será testado utilizando o programa compilado fornecido. Deve rodar sem requerer bibliotecas ou programas complementares. O sistema operacional utilizado durante os testes será o **Windows 7**. Assegure-se que a versão do programa entregue é uma versão de produção, ou seja, sem dados e controles requeridos pelo *debugger*. Executáveis que não executem pois dependem de DLLs ou quaisquer outros arquivos do Visual Studio para funcionar, provocarão perda de 8 pontos no trabalho.

4. Critérios de correção básicos

Leia atentamente o documento *Critérios de Correção dos Trabalhos* disponível na página da disciplina. Muitas das causas para a perda substancial de pontos decorrem meramente da falta de cuidado ao entregar o trabalho.

Não deixem para a última hora.
Este trabalho dá trabalho!