INF1301 Programação Modular
Período: 2015-2
Profs. Flavio Bevilacqua
1o. Trabalho
Data de divulgação: 12 de agosto (quarta-feira)
Data de entrega: 31 de agosto (segunda-feira)

1. Descrição do trabalho do período

O objetivo deste primeiro trabalho é familiarizar o aluno com os conceitos básicos da modularização de programas e da utilização do arcabouço de teste automatizado.

2. Descrição do primeiro trabalho

Neste trabalho deverá ser modificado o módulo exemplo **ARVORE** contido no diretório **Simples** do arcabouço de apoio ao teste automatizado. O arcabouço e alguns exemplos de seu uso estão no arquivo **arcaboucoteste-2-02.zip** disponibilizado para a turma via e-mail

Ao instalar (*dezipar*) este arquivo serão criados diversos diretórios. O módulo **ARVORE** e o respectivo módulo de teste específico (**TSTARV**) a que se refere o presente trabalho está no diretório **Simples**. A documentação do arcabouço encontra-se no diretório **Documentos**.

Inicialmente prepare-se para poder realizar o trabalho:

1. Baixe e instale ("*dezipe*") o arcabouço de apoio ao teste automatizado **arcaboucoteste-2-02.zip**.
2. Ajuste o arcabouço para que opere corretamente com a plataforma que você está utilizando, em especial a versão do compilador. Para isto é necessário regerar a biblioteca **ArcaboucoTeste.lib** contida no diretório **Arcabouc\Objetos**. Siga as instruções contidas no documento **ArcaboucoTeste-2-00-LeiaMe.pdf** que se encontra no diretório **Documentos** do arcabouço de apoio ao teste automatizado.
3. Estude o arcabouço, compile e teste o programa **ExemploSimples** assim como fornecido no diretório **Simples**. O objetivo disto é aprender a operar o arcabouço bem como ajustar as ferramentas de desenvolvimento a serem usadas, por exemplo o Visual Studio.
4. Estude o módulo de teste específico a ser alterado, bem como os módulos árvore e lista a serem testados.

O objetivo do 1º. trabalho é modificar e testar o módulo **ARVORE** executando as seguintes tarefas:

- Modificar o módulo para que o mesmo passe a operar uma matriz n X n. Deverá ser possível criar várias matrizes a partir deste módulo. Em cada matriz, todos os nós devem estar interligados com os oito adjacentes.
- Cada nó da matriz devem armazenar um ponteiro para uma lista. A estrutura de lista deverá estar encapsulada em outro módulo denominado **LISTA**. Já existe uma pasta **lista** no arcabouço com um exemplo deste módulo. Adapte esta lista para armazenar caracteres e acople ao módulo de matriz.
- Adapte as funções da matriz para que possam manipular as listas e os caracteres armazenados nas listas. Seu script de teste deverá primeiro criar uma lista inteira e depois armazená-la no nó da matriz.
- Para facilitar, não será necessário operar os nós das listas após as mesmas terem sido armazenadas na matriz.

- Adapte o módulo **TESTMAT (o equivalente ao TESTARV para a matriz)** de teste do módulo **MATRIZ** de modo que você possa testar as alterações introduzidas. O módulo de teste deve ser capaz de operar simultaneamente com até 10 matrizes.
- Atualize toda a documentação contida nos módulos de definição.
- Crie scripts completos para testar o módulo de Lista e o módulo de Matriz de Listas alterados pelo grupo.

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 o programa.
- Os arquivos de *script* de teste desenvolvidos pelo grupo.
- Os arquivos *batch* (**.bat**) que coordenam a execução dos testes.
- O programa executável (construto): **TRAB1-1.EXE apresentando o teste da Lista e o TRAB1-2.EXE apresentando o teste da Matriz de Listas. Deverão existir dois projetos (cada um com sua pasta específica e tudo que é necessário para a geração do executável respectivo) dentro do ZIP a ser entregue. Se esta organização não for respeitada perde-se dois pontos no trabalho.**
- 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-Trab01-idGrupo**. 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**. **CABE RESSALTAR QUE SE O PROGRAMA FOR EXECUTADO PELO PROFESSOR E NECESSITAR DE BIBLIOTECAS ADICIONAIS DESNECESSÁRIAS, O TRABALHO PERDERÁ 8 PONTOS MESMO TENDO RODADO NA MÁQUINA DOS COMPONENTES DO GRUPO. SUGESTÃO: ENVIE UM EXECUTÁVEL ANTES DO DIA DA ENTREGA PARA SER TESTADO NA MÁQUINA DO PROFESSOR.**
- Haverá perda de 1 ponto por dia de atraso considerando dias úteis de segunda a sábado. A tolerância é até 7h do dia seguinte.

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!