

INF1301 Programação Modular
Período: 2016-2
Profs. Flavio Bevilacqua
1o. Trabalho
Data de divulgação: 29 de agosto (segunda-feira)
Data de entrega: 19 de setembro (segunda-feira)

1. Descrição do primeiro trabalho

O primeiro trabalho tem por objetivo familiarizar o grupo com o funcionamento do arcabouço de teste.

Neste trabalho deverá ser modificado o módulo **LISTA** contido no diretório **lista** do arcabouço de apoio ao teste automatizado.

O módulo **LISTA** e o respectivo módulo de teste específico (**TESTLIS**) a que se refere o presente trabalho estão no diretório **Lista**. A documentação do arcabouço encontra-se no diretório **Documentos**.

Inicialmente prepare-se para poder realizar o trabalho:

1. Baixe e instale (“dezip”) o arcabouço de apoio ao teste automatizado **arcaboucoteste-2-02.zip**. Este arquivo contém toda a documentação necessária e encontra-se na página da aba **Software**.
2. Ajuste a janela de linha de comando **CMD** de modo que tenha cerca de 50 linhas de pelo menos 100 caracteres cada (ideal 120). Ajuste o buffer da janela **CMD** para que tenha cerca de 500 linhas da mesma largura que a janela física.
3. Ajuste o arcabouço para que opere corretamente com a plataforma que você está utilizando, em especial a versão do compilador. Para isto pode ser necessário reger 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.
4. Estude o arcabouço, compile e teste o programa **TesteLista** assim como fornecido no diretório **Lista**.
5. Estude o módulo de teste específico a ser alterado, bem como o módulo **LISTA** a ser testado.

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

- Modifique a estrutura de lista de forma que cada nó aponte para seu conteúdo armazenado. Atualmente cada nó possui um valor do tipo void (que será utilizado posteriormente como estrutura genérica). Com a mudança, este valor passará a ser um ponteiro para uma estrutura de dois campos: **{Iniciais (do tipo String), NomeCompleto (do tipo String)}**

Exemplo: {MCS, Maria Cristina Silva}

- Inclua uma função que receba estes dois parâmetros e retorne por referência a estrutura preenchida

- Altere todas as funções necessárias (e respectivas interfaces) para manipular esta estrutura. Por exemplo, ao incluir o nó na árvore, esta estrutura deverá ser passada por valor.
- Não esqueça de desalocar as estruturas desnecessárias
- Implemente uma função de inserção ordenada na lista. O algoritmo deverá utilizar sempre que possível as outras funções pré-desenvolvidas de forma que não haja duplicação de código. A chave de ordenação será **Iniciais**.
- Implemente uma função que apresente o conteúdo da lista em ordem, apresentando todas as informações de cada estrutura.
- Redija os seus próprios scripts de teste. Procure redigir scripts suficientemente rigorosos e completos.
- Verifique se tudo está correto (não se esqueça de documentar e alterar a documentação atual, conforme necessário). Se estiver, empacote (i.e. crie o arquivo **.zip** de entrega do trabalho) verifique se esse arquivo está correto e completo e envie-o para o instrutor.

2. 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 dois projetos, um para a lista tal como se encontra no arcabouço e outro com a lista modificada. Para cada um deve ser entregue:

- 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.
- programa executável (construto): **TRAB1-1.EXE**. com o projeto da lista tal como se encontra no arcabouço e **TRAB1-2.EXE** com o projeto da lista alterada. **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** 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 (versão release). **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.**

3. Critérios de correção básicos

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