INF1301 Programação Modular
Período: 2018-1
Prof. Alessandro Garcia
2o. Trabalho
Data de divulgação: 2 de abril (segunda-feira)
Data de entrega: 7 de maio (segunda-feira)

1. Descrição do trabalho do período

O objetivo do trabalho do período é especificar, projetar e implementar o jogo Truco. Neste jogo, dois, quatro ou seis jogadores são divididos em duas equipes, utilizando-se um baralho de **40 cartas**. Em cada **"mão"**, cada jogador recebe três cartas. Cada mão pode ter até 3 rodadas e pode valer 1, 3, 6 ou 12 pontos. A partida termina quando uma das equipes somar 12 pontos. Ao longo do período, o grupo deverá desenvolver o jogo Truco completo e operacional, levando em conta todas as regras existentes.

2. Descrição do trabalho T2

Neste trabalho, o objetivo é especificar os requisitos com base nas regras do jogo Truco, criar a arquitetura do programa, implementar e testar os módulos básicos do jogo. O módulo principal que administra todos os outros e permite jogar uma partida completa será desenvolvido no trabalho T4. Porém, é importante que no trabalho 2, os módulos desenvolvidos sejam testados individualmente de forma rigorosa.

Para armazenar as cartas do jogo de truco, o trabalho deverá **obrigatoriamente** reutilizar a estrutura de lista duplamente encadeada genérica disponibilizada pelo arcabouço. Esta estrutura estará encapsulada em um módulo de Lista Genérica que será acoplado a todos os módulos que o grupo achar necessário.

Passos a serem seguidos neste trabalho 2.

1. Estude as regras do jogo de truco e crie um documento que contenha a **Especificação de Requisitos** de **TODO O TRABALHO DO PERÍODO**. Mesmo que no trabalho 2 não seja objetivo completar a implementação do jogo como um todo, a especificação deverá nortear todo o projeto.

2. Crie a arquitetura do programa, apresentando um diagrama contendo a definição de todos os módulos (e respectivas interfaces) necessários para a implementação do jogo de truco. Novamente, isto se refere a **TODO O TRABALHO DO PERÍODO**. Este diagrama apresentará cada módulo necessário, suas interfaces, seus interrelacionamentos, e as funções disponibilizadas em cada interface. Adicione este diagrama ao documento, elaborado no item 1, criando assim uma boa especificação de programa que documente o trabalho que o grupo terá ao longo do período.

3. Elabore o modelo estrutural do trabalho. É importante que haja apenas UMA estrutura para cada jogo. Esta será uma lista de listas. Cada nó da lista principal apontará para a cabeça de uma lista com as cartas do jogador 1, outro para a lista com as cartas do jogador 2, etc..., outro apontará para a lista de cartas descartadas na mesa, e assim por diante. Esta estrutura será especificada no trabalho 2, porém implementada no trabalho3.

4. Elabore e teste separadamente o módulo de embaralhamento. Este módulo deverá receber uma sequência representando baralho ordenado e retornar esta sequência embaralhada. Deverá ser criado um projeto separado apenas para este teste. Elabore o script para testar este módulo completamente.

5. Para os testes, deve-se seguir uma linguagem de script definida pelo grupo. Além disso, incorpore, de forma rigorosa, implementação de assertivas nas funções dos módulos.

Observação importante: Não se preocupem ainda em implementar a lista de listas (estrutura principal). Esta só será implementada no trabalho 4. No entanto, esta será ESPECIFICADA no documento do trabalho 2 conforme já definimos neste enunciado.

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:

- Documento com Especificação de Requisitos, Arquitetura do Programa, Modelo Físico conforme notação UML apresentada na aula de Modelagem (contendo o Modelo, um Exemplo da estrutura com base no modelo e as respectivas assertivas estruturais).

- Um arquivo **LEIAME.TXT** contendo a explicação de como utilizar o(s) programa(s).

- Os arquivos-fonte dos diversos módulos que compõem o programa e dos seus respectivos módulos de controle de testes executáveis. Obs.: O projeto do Trabalho 2 deverá obedecer uma estrutura de diretórios bem definida. (siga como padrão a estrutura de diretórios adotada pelo projeto /instrum disponível no arcabouço)

- Os arquivos de script de teste desenvolvidos pelo grupo.

- Os arquivos de composição (.comp) que permitem a criação dos arquivos .make e .build.

- Os arquivos batch (.bat) que auxiliam o processo de geração dos arquivos .make e .build. (veja os exemplos geramake.bat e geratudo.bat no arcabouço).

- Os arquivos batch (.bat) que auxiliam o processo de geração dos arquivos .exe. (veja os exemplos compila.bat e compilatudo.bat no arcabouço).

- Os arquivos *batch* (.bat) que coordenam a execução dos testes. (veja o exemplo testatudo.bat no arcabouço).

- Programa executável (construto) em cada projeto: **TRAB2-1.EXE**, **TRAB2-2.EXE**, **TRAB2-3.EXE**, e assim por diante. Obs.: Caso o cliente de email não deixe anexar arquivos com

extensão .exe, renomeie os arquivos para uma outra extensão qualquer, como por exemplo: **TRAB2-1.EXE.txt**. Deixe claro no arquivo **LEIAME.TXT** qual o padrão de renomeação adotado pelo grupo.

- 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.

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!