

INF1301 Programação Modular

Período: 2017-1

Prof. Alessandro Garcia

Trabalho T2

Data de divulgação: 1 de abril (sábado)

Data de entrega: 8 de maio (segunda-feira)

1. Descrição do trabalho do período

O objetivo do trabalho do período é implementar o jogo Free Cell. Neste jogo, as cartas de um baralho começam divididas em oito sequências embaralhadas com os valores e naipes visíveis. A medida que o jogo se desenrola, cartas são passadas de uma sequência para outra seguindo determinadas regras. Existem também quatro espaços extras (Extra1, Extra2, Extra3, Extra4) para colocação de até uma carta. Além disso, existem mais quatro espaços de sequências ordenadas (Naipes1, Naipes2, Naipes3 e Naipes4) destinadas às sequências ordenadas, uma para cada naipe. Ao longo do período, o grupo deverá implementar um jogo de Free Cell 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 FreeCell, criar a arquitetura do programa, implementar e testar os módulos de embaralhamento, sequências visíveis, espaços extras e sequências ordenadas. 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.

Todo o trabalho deverá reutilizar a estrutura de lista duplamente encadeada genérica para armazenar as sequências e espaços extras. Esta estrutura estará encapsulada em um módulo de Lista Genérica que será acoplado a todos os módulos que o grupo achar necessário. Já existe uma implementação do módulo Lista Genérica, distribuído com o código do arcabouço, que deve ser reusado e, se necessário, melhorado e adaptado pelo grupo.

Passos a serem seguidos neste trabalho 2.

1. **Módulo Lista Genérica:** conforme mencionado acima, tal módulo deverá ser reutilizado e aperfeiçoado, se necessário. A massa de teste disponível para o módulo Lista Genérica é limitada e deve ser melhorada, conforme discutido em sala de aula.
2. **Especificação de Requisitos:** Estude as regras do Free Cell e crie um documento que contenha a Especificação de Requisitos para o programa que irá implementar o jogo. Esta especificação deve contemplar todos os requisitos a serem considerados em 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.
3. **Arquitetura do Programa:** Crie a arquitetura do programa, apresentando um diagrama contendo a definição de todos os módulos (e respectivas interfaces) necessários para a implementação do jogo Free Cell. 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.

4. **Sobre a estrutura de dados:** é importante que haja apenas UMA estrutura de dados para armazenar todos elementos do jogo. Esta estrutura de dados única será uma lista de listas. Cada nó desta lista de listas apontará para a cabeça de uma lista necessária, ou seja, um nó da lista principal apontará para a lista da sequencia1, outro para a lista da sequencia2, etc..., outro apontará para a lista de espaços extras, e assim por diante. Esta estrutura será especificada no trabalho 2, porém implementada no trabalho 4.
5. **Módulo de Embaralhamento:** 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.
6. **Elabore e teste separadamente o módulo de espaços extras.** Este módulo será responsável por uma lista de quatro nós representando cada espaço extra do jogo. Deverá ser criado um projeto separado apenas para este teste. Elabore o script para testar este módulo completamente.
7. **Elabore e teste separadamente o módulo de seqüência visível.** Este módulo será responsável por uma das oito listas de seqüência visível. Deverá ser criado um projeto separado apenas para este teste. Elabore o script para testar este módulo completamente. Não se esqueça de implementar neste módulo a regra que somente permite inserir uma nova carta apenas em cima da carta do topo (a que no tabuleiro gráfico fica totalmente visível) e esta carta inserida precisa ter cor invertida e estar em ordem decrescente (tal como a regra do jogo). Elabore cuidadosamente as condições de retorno de sucesso ou falha necessárias ao contexto.
8. **Elabore e teste separadamente o módulo de seqüência ordenada.** Este módulo será responsável por uma das oito listas de seqüência ordenada. Deverá ser criado um projeto separado apenas para este teste. Elabore o script para testar este módulo completamente. Não se esqueça de implementar neste módulo a regra que somente permite inserir uma nova carta apenas em cima da carta do topo (a que no tabuleiro gráfico fica totalmente visível na seqüência ordenada) e esta carta precisa ter o mesmo NAIPE e estar em ordem crescente (tal como a regra do jogo). Elabore cuidadosamente as condições de retorno de sucesso ou falha necessárias ao contexto.
9. 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.
- 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. (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.
- 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 XP**. Assegure-se que a versão do programa entregue é uma versão de produção, ou seja, sem dados e controles requeridos pelo debugger (versão release).

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!**