

# INF1301 Programação Modular

Período: 2019-1

Prof. Alessandro Garcia

4o. Trabalho

Data de divulgação: 3 de junho (segunda-feira)

Data de entrega: 1 de julho (segunda-feira)

## 1. Descrição do trabalho do período

O objetivo do trabalho do período é especificar, projetar, implementar e testar todos os módulos do jogo de Ludo (também conhecido como "Furbica"). Este é um jogo de tabuleiro para dois, três ou quatro jogadores. Cada jogador: (i) possui quatro peões de uma cor específica, e (ii) possui uma casa de origem e casa final no tabuleiro. O objetivo de cada jogador é ser o primeiro que, partindo de sua casa de origem, dá a volta no tabuleiro e chega com quatro peões à casa final antes dos adversários. Cada peão pode ser movido cada vez que o seu jogador lança o dado.

No trabalho T2, o objetivo foi especificar os requisitos com base nas regras do jogo de Ludo, criar a arquitetura do programa, implementar e testar os módulos básicos do jogo. Os módulos desenvolvidos foram testados individualmente. **O módulo principal que administra todos os outros e permite jogar uma partida completa será desenvolvido no trabalho T4. Outros módulos não entregues no T2 também serão concluídos no T4.** Além disso, no T4, também será incorporada instrumentação em um módulo central do programa de forma a tornar o programa do jogo de Ludo mais confiável.

Como resultado final do T4, o grupo entregará o jogo de Ludo completo e operacional, levando em conta todas as regras existentes, de acordo com a especificação de requisitos entregue no T2. Assume-se que o programa seguirá a arquitetura também entregue no T2. Além disso, um dos módulos conterá instrumentação necessária para análise da cobertura dos testes, conforme descrito em detalhes na próxima seção.

## 2. Descrição detalhada do trabalho T4

Primeiramente, neste quarto trabalho deve ser concluída a implementação de todos os módulos do jogo de Ludo, de tal forma que seja possível a realização de partidas entre jogadores. Deve ser utilizada a versão completa do arcabouço de teste. Esta, em adição à versão simplificada, possui o módulo **CONTA** para controle da cobertura de testes. Os *scripts* de teste a serem desenvolvidos neste trabalho devem satisfazer a cobertura por arestas para algumas funções (conforme descrito abaixo). Estude o exemplo contido na pasta "*Instrum*" do arcabouço.

Lista detalhada das tarefas do 4o. trabalho:

- conclua a implementação do programa do jogo de Ludo através da implementação dos módulos que não foram concluídos ou codificados no trabalho T2. A implementação deve estar aderente a especificação de requisitos e ao modelo da arquitetura, ambos já entregues no T2.
- entregue novamente as especificações dos requisitos e o modelo da arquitetura, reportando e justificando quaisquer modificações que foram necessárias em tais documentos após a entrega do T2.
- o módulo LISTACIRCULAR deve conter assertivas executáveis de entrada e saída em todas suas funções.

- **IMPORTANTE:** para possibilitar o teste de partidas completas do programa do jogo de Ludo, forneça um documento que guie um testador a testar o programa via Interface (linha de comando), de forma que possa ser certificado que as partidas do jogo estão funcionando por completo. Esse documento pode ser redigido em um arquivo Word ou .txt. Importante lembrar que este documento para teste completo do jogo é adicional aos scripts de teste individual de cada módulo. O documento deve ser detalhado o suficiente com os comandos a serem usados na interface para que o testador consiga testar partidas por completo (ou seja, testar/simular diferentes partidas do jogo).
- baixe e estude a documentação e os exemplos de código do arcabouço de teste. Estude o módulo **CONTA**.
- produza *scripts* de teste capazes de examinar o funcionamento do **módulo LISTACIRCULAR** segundo o critério de completeza **cobertura de arestas**. Todas funções deste módulo devem conter contadores de passagem controlando cada aresta de decisão existente em seu código, mesmo aquelas arestas associadas com assertivas. O módulo **CONTA** deve ser utilizado para a análise automatizada deste critério. Ao final do teste nenhum dos contadores criados para o módulo **LISTACIRCULAR** deverá conter zero. Ou seja, o teste deve ser completo segundo o critério cobertura de arestas, sendo que este critério será controlado pela instrumentação inserida.
- conforme mencionado acima, deve ser produzido os *scripts* de teste de modo que o conjunto de casos teste percorra todas as arestas do código das funções do módulo **LISTACIRCULAR**. Mostre que isto de fato ocorreu através de contadores de passagem inseridos nas funções. Caso um determinado *script* “voe” (ou seja, a execução do programa seja interrompida), particione-o em vários de modo que cada condição que leve a um cancelamento seja testada por um *script* individual. Explique a razão para o *script* “voar”.

### 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. Todos os módulos, funções, structs, e variáveis globais devem ser devidamente especificados segundo o padrão.

O trabalho deve ser enviado por e-mail em um único arquivo **.ZIP** (codificação do *attachment*: MIME). O arquivo **ZIP** deve conter:

- os arquivos de documentação de requisitos e modelo arquitetural, bem como relato e justificativas das alterações (caso necessárias) feitas após o T2.
- os arquivos fonte dos diversos módulos que compõem o programa.
- os programas executáveis: **TRAB4-*i*.EXE**, em que *i* é um identificador ordinal do programa. Um dos programas executáveis deverá ser com a instrumentação ligada e o outro deverá ser com a instrumentação desligada.
- os *scripts* **MAKE** e, se for o caso, **BUILD** necessários para gerar os programas executáveis.
- todos os *scripts* de teste.
- o arquivo de declaração dos contadores.
- arquivo de totalização dos contadores. O *script* de teste utilizado para verificar estruturas corretas deve conter um comando que zere todos os contadores. Dessa forma a sequência de teste corretamente totalizará as contagens, mesmo que o teste seja efetuado repetidas vezes.

- os arquivos *batches* necessários. Em particular, um arquivo *batch* (**.bat**) que encadeia os testes a serem realizados e finaliza imprimindo o conteúdo do arquivo de totalização de contadores para os módulos onde está sendo feito o controle de arestas nos testes.
- um arquivo **LEIAME.TXT** contendo a explicação de como utilizar os programas e os *scripts* de teste.
- tantos arquivos **RELATORnome.TXT** quantos forem os membros do grupo. O elemento nome deve identificar o membro do grupo. Estes arquivos devem conter uma tabela de registro de trabalho organizada conforme descrito nos enunciados de trabalho anteriores.

**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. 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 completa!
- A mensagem de encaminhamento deve ter o assunto (*subject*) **INF1301-Trab04-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**.

Observações:

- a mensagem de encaminhamento deve conter o assunto (*subject*) **INF1301 Trabalho 4 aaa-bbb-ccc** no qual

- ♦ **aaa-bbb-ccc** identifica a composição do grupo (duas ou três letras por participante)
- ♦ texto deve conter somente a lista dos alunos que compõem o grupo (formato: número de matrícula, nome e endereço e-mail).
- não deve ser acrescentado qualquer texto extra à mensagem.
- caso o instrutor não consiga ler o arquivo **zip** na sua máquina o grupo perde 1 ponto e terá a chance de resubmeter o trabalho em um arquivo correto.
- 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 Windows. O programa deve operar em uma janela CMD (DOS) sob este sistema operacional.

#### **4. Critérios de correção básicos:**

Leia atentamente a folha de critérios de correção entregue no primeiro dia de aula. Esta folha encontra-se em formato pdf, na página Web da disciplina.

São verificados ainda:

- modelos e assertivas não existem **-3**, assertivas não existem ou excessivamente incorretas **-2**; modelos e/ou assertivas com algumas incorreções **-1** ponto.
- teste não é completo **-2**
- teste não utiliza contadores **-2**

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