

**INF1301 Programação Modular**  
**Período: 2016-2**  
**Prof. Flavio Bevilacqua**  
**3o. Trabalho**

Data de divulgação: 5 de outubro (quarta-feira)  
Data de entrega: 16 de novembro (quarta-feira)

## 1. Descrição do trabalho inicial

---

O objetivo deste trabalho é implementar um juiz de jogo de Xadrez entre dois jogadores humanos. A aplicação permitirá que peças sejam movimentadas de forma alternada como em uma partida de xadrez. Após cada movimento, a aplicação avisará se esta peça movimentada pode ser atacada por alguma peça oponente (ocupa uma das casas protegidas). No caso do Rei, caso ele esteja em uma casa protegida e não possa se movimentar para qualquer outra casa, o programa emite a mensagem "Cheque Mate". Caso ele esteja em uma casa protegida e possa se movimentar para outra casa, emite a mensagem "Cheque".

## 2. Descrição do terceiro trabalho

---

### Módulo Jogo

Este módulo estará acoplado ao módulo Tabuleiro criado no trabalho2 e será responsável pela interface com os jogadores. Devem ser desenvolvidos dois programas, um realizando testes e o outro de "produção". Este segundo programa não conterá o arcabouço de teste, uma vez que a aplicação não se utilizará de scripts.

No início do jogo, o "juiz" pede o nome dos participantes e apresenta o tabuleiro com as peças nas posições iniciais.

Cada rodada deverá apresentar o tabuleiro na tela e pedir para um dos jogadores realizar o movimento de uma de suas peças. O "juiz" deverá atualizar a lista de ameaçantes e ameaçados de cada peça e informar, após o movimento da peça, quais ela está ameaçando e quais a estão ameaçando.

Todas as rodadas seguirão assim, alternando os dois jogadores.

No caso do Rei, caso ele esteja em uma casa protegida e não possa se movimentar para qualquer outra casa, o programa emite a mensagem "Cheque Mate". Caso ele esteja em uma casa protegida e possa se movimentar para outra casa, emite a mensagem "Cheque".

A partida termina com o Cheque Mate ou a qualquer momento que um dos jogadores digite "FIM".

A versão com teste deve simular um trecho de um jogo. Para tal o script monta um tabuleiro em estado intermediário e realiza diversas (pelo menos 6) jogadas, verificando a validade delas e a corretude das listas "ameaçada" e "ameaçantes". Dica: tentem desenvolver o programa de teste de tal forma que baste substituir o arcabouço e o módulo de teste específico por um módulo (main) que efetue a entrada de dados necessária para o jogo em "produção".

## 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 os programas.
- os arquivos de *script* de teste desenvolvidos pelo grupo.
- o programa executável: TRAB3-1.EXE. Caso queiram entregar mais de um executável, estes deverão ter o nome no formato: TRAB3-2.EXE, TRAB3-3.EXE, e assim por diante. Cada módulo deverá ser testado separadamente e cada um terá o seu programa executável. Deverão existir 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. Planeje a sequência de desenvolvimento que permita o teste minimizando o desenvolvimento de módulos de apoio ao teste (stubs ou enforcements).
- um arquivo **LEIAME.TXT** contendo a explicação de como utilizar o(s) programa(s).
- um documento PDF com a estrutura unificada da aplicação (modelo, exemplo e assertivas), a arquitetura completa da aplicação (módulos, interfaces e funções de acesso) e o manual do usuário da aplicação.
- 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                | • codificar              |
| • especificar os módulos | • revisar código         |
| • especificar as funções | • planejar teste         |
| • revisar especificações | • revisar plano de teste |
| • projetar               | • testar                 |
| • revisar projetos       | • corrigir               |

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-Trab03-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.

## 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. Este documento também pode ser encontrado no anexo à descrição da disciplina distribuída na primeira aula.

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