

# TRABALHO DA DISCIPLINA PROGRAMAÇÃO ORIENTADA A OBJETOS 2

## 1 Considerações Gerais

### 1.1 Data de Entrega

O trabalho deve ser entregue até as 23:59 h do dia **04/06/2014**.

### 1.2 Grupo

O trabalho deve ser feito por 2 alunos.

### 1.3 Atraso

Para cada dia de atraso, a nota do trabalho será penalizada da seguinte forma: 2<sup>n</sup>, ou seja, caso o aluno atrase 1 dia, o aluno perderá 2 pontos, 2 dias de atraso 4 pontos e etc.

### 1.4 Forma de entrega

- 1.4.1 O código fonte do trabalho deve ser salvo em um repositório Git (e.g., GitHub);
- 1.4.2 O trabalho deve ser submetido no moodle;
- 1.4.3 O nome do arquivo deve ser no seguinte formador: “POO2\_Trab1\_Nome1\_Nome2.tar.gz”, onde nome1 é o primeiro nome e o último sobrenome do primeiro aluno e nome2 é o primeiro nome e o último sobrenome do segundo aluno.
- 1.4.4 O arquivo POO2 deve conter: um documento descrevendo os padrões de projeto utilizados no trabalho e um documento contendo a URL do projeto no GitHub. Nesse último documento, deve conter apenas a URL.
- 1.4.5 O documento descrevendo os padrões de projeto deve ser no formato PDF.

### 1.5 Atenção:

- 1.5.1 Trabalhos com erros de compilação receberão nota **ZERO**.
- 1.5.2 Trabalhos incompletos receberão nota **ZERO**.
- 1.5.3 Trabalhos com erros de lógica serão penalizados.
- 1.5.4 Trabalhos considerados iguais receberão nota **ZERO**. Será considerado plagio os trabalhos que tiverem mais de 0.1% de similaridade.
- 1.5.5 Caso um ou mais trabalhos sejam considerados iguais, o plagiador deve enviar um e-mail para o professor informando que copiou o trabalho. Desta forma, o aluno que teve o seu trabalho plagiado receberá a nota devida do trabalho. O plagiador ficará com a nota zero no trabalho.

## 2 Especificação do trabalho

### 2.1 Objetivo do Trabalho

O objetivo do projeto é desenvolver um jogo de xadrez. O jogo de xadrez deve permitir que dois jogadores joguem ao mesmo tempo, no mesmo computador.

### 2.2 Descrição do Problema

O objetivo do xadrez é dar xeque-mate ao Rei adversário, ou seja, colocando-o sob ameaça de captura (xeque), se quem ele tenha como escapar desse xeque. Para isso, cada jogador dispõe de 16 peças, sendo 1 Rei, 1 Rainha, 2 Bispos, 2 Cavalos, 2 Torres e 8 peões. As peças estão organizadas conforme a imagem abaixo.



Figura 1.: Exemplo de tabuleiro de xadrez.

É importante comentar que o jogador branco sempre inicia a partida. Em questões de pontuação, o peão vale 1 ponto, o bispo vale 3 pontos, o cavalo vale 3 pontos, a torre vale 5 pontos e a dama vale 9 pontos. Caso o rei seja capturado, é fim do jogo.

Além dos movimentos de cada peça, os seguintes movimentos devem ser implementados: Roque Grande, Roque Menor, *En Passant*, Xeque e Xeque-Mate. Esse movimentos estão descritos em detalhes em:  
[http://webchess.wvr.com.br/tutorial\\_xadrez/index.html](http://webchess.wvr.com.br/tutorial_xadrez/index.html) e  
<http://www.tabuleiroidexadrez.com.br/regras-do-xadrez.htm>.

### 2.3 Requisitos do Projeto

- **Requisitos Funcionais:**

- Uma vitória vale 3 pontos;
- Um empate vale 1 ponto;
- Uma derrota vale 0 pontos;
- Todo jogador é cadastrado com as seguintes informações: nome, email e apelido. O jogador pode editar os seus dados. Um jogador não pode editar os dados de outros jogadores;
- É necessário guardar o número de vitórias, derrotas e empates de cada jogador.
- O Ranking de cada jogador é feito tendo como base o número de vitórias, derrotas e empates;
- Deve permitir ao jogador iniciar uma nova partida contra um outro jogador;
- Deve permitir ao jogador iniciar uma nova partida contra o computador;
- O jogador pode solicitar empate. Caso isso ocorra, o jogador adversário deve concordar. Somente assim, a partida é finalizada. Nesse caso, cada jogador recebe um ponto;
- O jogador pode informar derrota em qualquer momento;
- O jogador pode solicitar que a partida seja interrompida em qualquer momento. Assim, a partida será retomada em qualquer momento posterior;
- Um jogador pode enviar mensagens para outro jogador durante a partida. As conversas devem ser salvas.

- **Requisitos não funcionais:**

- O projeto deve conter uma interface gráfica feita em java Swing;
- O projeto deve apresentar interface “amigável”. A interface deve conter menu, submenus e outros elementos do swing que permita uma boa navegação na interface;
- Todas as entidades devem ser salvas em um banco de dados;
- As conversas devem ser demonstradas em forma de chat;

### **3 Requisitos da Implementação**

- Reutilizem o trabalho do semestre passado feito em POO1.

- O código deve ser implementado utilizando Generics e Interface;
- Deve ser usado Reflection do Java;
- O código deve ter o menor número de classes possível;
- É proibido o uso de variáveis globais.
- O trabalho deve usar o MAVEN;
- O trabalho deve apresentar testes unitários de todas as funções de negócio;
- O trabalho deve utilizar Clean Code (funções pequenas, variáveis com nomes significativos, ...);
- O código deve ser orientado a objetos. Caso isso não ocorra, o trabalho NÃO será aceito.
- O trabalho deve implementar o padrão de projeto MVC e as subdivisões dessas camadas;
- O trabalho deve implementar os padrões criativos (e.g., método fabrica, fabrica abstrata, protótipo e singleton);
- O trabalho deve implementar os padrões estruturais (e.g., compositor);
- O trabalho deve apresentar os padrões de nomenclatura JAVA para: Classes, pacotes, variáveis e código.
- Programação orientada a interface e não a implementação.
- Banco de dados HyperSonic.
- Deve ser utilizado o padrão DAO para a camada de banco de dados;

#### **4 Avaliação**

Avaliação do trabalho será feito em três partes. A primeira parte é referente a implementação do problema proposto. A segunda parte é referente ao trabalho escrito solicitado. A terceira parte é referente a entrevista com cada membro do grupo. A nota final individual e será calculada da seguinte forma  $\text{Nota final} = ((N1 + N2)/2) * N3$ . N1 é referente ao código, N2 ao relatório e N3 referente a nota individual do membro do grupo na entrevista, respectivamente.

O trabalho escrito deve conter: (i) diagrama de padrões de projeto, (ii) explicação do uso de cada padrão, (iii) descrição da utilização do padrão MVC e (iv) uma discussão de como o uso de padrões de projeto auxiliou no desenvolvimento. A discussão deve ter como base dados numéricos retirados do sistema SONAR (<http://www.sonarqube.org/>). A tabela 1 apresenta a pontuação do relatório e a tabela 2 da implementação.

<b>Critério</b>	<b>Descrição</b>	<b>Valor</b>
Apresentação	O trabalho deve estar bem escrito e apresentar um sumário	1.0
Diagrama de classe	Diagrama dos padrões.	2.0
Explicação do uso de cada padrão de projeto.	Explicação do uso de cada padrão de projeto	2.0
MVC	Descrição da utilização do padrão MVC	2.0
Discussão sobre o uso de padrões	Deve descrever como os padrões de projeto ajudaram, ou não, na melhora do desenvolvimento do sistemas. Faça uma comparação com o código feito anteriormente. Utilize o software Sonar para a análise dos dados.	3.0

**Tabela 1 - Definição dos pontos do relatório.**

<b>Critério</b>	<b>Descrição</b>	<b>Valor</b>
Apresentação	O código deve estar limpo e seguindo os princípios de Clean code.	2.0
MVC	Utilização correta da distribuição das camadas	2.0
Banco de dados	Uso de banco de dados	1.0
Código em O.O e Padrões de projeto.	Código utilizando O.O corretamente e testes	5.0

**Tabela 2 – Definição dos pontos da implementação.**