

Nome:

Turma:

Professor: Giuliano Paes Carnielli

Data:

Avaliação Prática de Prog. Orientada a Objetos: Projeto Jogo de Xadrez

Instruções:

1. Esta atividade deve ser desenvolvida em grupos de 5 alunos.
2. Todo grupo deve ter um nome único.
3. A entrega deve ser feita, por email, como um pacote ZIP contendo um projeto do Eclipse.
4. O título do email deve ter o formato: [POO] - Projeto Pratico <1/2> - <nome do grupo>
5. Um trabalho estará sujeito a anulação imediata caso: não compile, não apresente minimamente as funções esperadas.
6. Os nomes de todos os membros do grupo devem aparecer no cabeçalho de arquivos de código e no corpo do email usado para entregar a avaliação.
7. O código deve ser organizado e bem documentado, com o risco de decréscimo na nota.
8. O plágio ocasionará a anulação de todas as avaliações envolvidas.
9. **BOM TRABALHO.**

1. Jogo de Xadrez:

Incluiremos novas funcionalidades ao Jogo de Xadrez simplificado, desenvolvido na primeira etapa deste projeto.

Os grupos devem observar as especificações fornecidas neste documento. Qualquer variação deve ser discutida com o professor, aprovada e documentada.

2. Pré-Requisitos:

O programa deve possuir, minimamente, as seguintes classes. Os detalhes são deixados para que os grupos decidam, mas tudo o que for sugerido abaixo, deve efetivamente fazer parte do projeto.

Esta etapa baseia-se no projeto construído no primeiro bimestre. Portanto, para que as novas funcionalidades possam ser adicionadas conforme as especificações deste documento, alguns pré-requisitos devem ser atendidos:

- a. Cada grupo deve ter uma versão funcional do programa "Jogo de Xadrez"
- b. Observações feitas pelo professor, durante a avaliação da 1a Etapa, devem ter sido implementadas
- c. O programa deve ser estruturado conforme as especificações da 1a parte do projeto
- d. A Classe Jogo deve possuir um construtor que cria todas as peças do jogo, armazenando-as em uma Lista de Peças. Depois cria o Tabuleiro.
- e. A Classe Peca deve armazenar as coordenadas atuais de uma peça no tabuleiro
- f. A Classe Tabuleiro deve possuir um construtor que estabelece a configuração inicial do tabuleiro de acordo com uma lista de peças passada como parâmetro

3. Preparação:

Para viabilizar a implementação das novas funcionalidades, as seguintes modificações são necessárias.

Nome:

Turma:

Professor: Giuliano Paes Carnielli

Data:

Classe Jogada (Nova):

Esta classe deve ser usada para registrar cada jogada em uma partida. Deve ter os seguintes atributos:

- Peca pecaMovida: referência à peça que foi movida na jogada
- Peca pecaCapturada: referência a uma peça capturada na jogada (se não houver captura, este atributo deve ser *null*)
- Coordenada Origem: coordenada de origem da jogada (x, y)
- Coordenada Destino: coordenada de destino da jogada (x, y)

Classe Jogo: um novo atributo deve ser acrescentado para armazenar jogadas.

- *Collection* jogadas: deve-se escolher uma coleção adequada para armazenar as jogadas realizadas como em uma "Pilha". A decisão sobre o tipo de *Collection* é parte do exercício

Classe Peça: um novo atributo deve ser acrescentado, para individualizar cada peça.

- `int codPeça`: código individual de uma peça. Cada peça no jogo deve ter seu próprio código. Considerando que temos 32 peças no jogo, uma sugestão seria numerar cada peça de 1 a 32

4. Novas Funcionalidades:

A. Função "Desfazer" (*Undo*):

Um jogador deve ter a possibilidade de se "arrepender" de uma jogada realizada. Se isso acontecer antes do adversário realizar uma nova jogada, o jogador arrependido poderá digitar "*undo*" e sua última jogada será desfeita e uma nova poderá ser realizada.

Esta função deve usar a "*Collection* jogadas" para recuperar a última jogada realizada. Os dados contidos no objeto da última jogada devem ser suficientes para permitir a reversão da jogada.

Considerações:

- Toda jogada bem sucedida deve criar um objeto 'Jogada', que armazena suas informações.
- Necessário apenas um nível de "*undo*". Se o comando "*undo*" for passado duas vezes seguidas, uma mensagem de erro informando "Operação não disponível" deve ser exibida.
- O comando "*undo*" será digitado sempre na vez do adversário. Por isso, esse comando deve retornar a vez para o jogador anterior.

Ex:

Branças: "2d 4d" <-- Jogada das brancas

Pretas: "undo" <-- Na vez das pretas, as brancas se arrependem

Branças: "2e 4e" <-- Nova jogada das brancas

- A operação de "*undo*" deve remover o objeto 'Jogada' que foi desfeito

Nome:

Turma:

Professor: Giuliano Paes Carnielli

Data:

B. Função "Salvar/Carregar":

Deve ser possível salvar uma partida em andamento e carregá-la mais tarde.

- i. Os dados de uma partida devem ser salvos em um arquivo texto. O jogo deve dar opção ao jogador de nomear o arquivo para salvar o jogo, mas deve armazenar todos os arquivos em um diretório próprio.
- ii. O grupo deve definir o formato dos dados salvos no arquivo. Entretanto, os dados salvos devem ser os seguintes:
 - o Situação de todas as peças (lista de peças)
 - o Situação das jogas realizadas (pilha de jogadas)
- iii. O novo atributo "codPeca" deve ser usado para refazer corretamente as ligações entre Jogada e as peças (ex. cada lado possui 8 peões, mas cada Jogada precisa referenciar de forma correta o exato peão movido na jogada em questão)
- iv. O construtor de Jogo, no momento, cria todas as peças e as posiciona de forma estática. Isso deve ser alterado de forma que as peças sejam criadas de acordo com os dados presentes em um arquivo de jogo
- v. Para tornar o funcionamento uniforme, devemos criar um arquivo de jogo "default.xdz", que não pode ser sobrescrito, contendo a situação inicial de um novo jogo
- vi. O construtor de Jogo deve ser alterado para que pergunte ao jogador se ele deseja "Iniciar nova partida" ou "Carregar uma partida".
 - o A primeira opção deve usar "default.xdz" para iniciar um jogo novo.
 - o A segunda opção deve exibir uma lista de arquivos de jogo, e esperar que o usuário entre o nome de um deles, para então carregar os dados do arquivo escolhido.

C. Função "Tratamento de Erros":

O código do programa 'Jogo de Xadrez' deve ser refatorado para que todo o tratamento de erros seja feito com base em Exceções.

Ex: um jogador entra uma jogada que aponta para uma casa vazia do tabuleiro. Uma exceção deve ser lançada e tratada de forma estruturada.