

Trabalho MC322

Etapa 02

Jogo de Damas

O seu desafio aqui é escrever um conjunto de classes que simule o funcionamento de um jogo de damas. Este jogo tomará como base o jogo Resta Um e deve seguir duas premissas do mesmo:

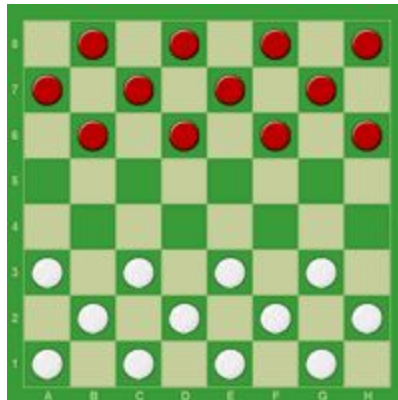
- deve haver pelo menos uma classe cujo objeto represente o tabuleiro;
- deve haver classes que representem peças do tabuleiro.

Como o jogo de damas tem dois tipos de peça - a peça comum e a dama - deve ser feita uma hierarquia de classes com uma classe que represente uma peça genericamente (com as características comuns de qualquer peça) e duas subclasses que representam a peça comum e a dama.

A requisição de movimento será feita obrigatoriamente à peça. A peça verificará se o movimento é compatível com o seu tipo (peça comum ou dama) e se o movimento é possível no tabuleiro. Se a peça verificar que é possível, a peça comunica ao tabuleiro o movimento desejado.

A verificação de captura pode ser feita pela peça ou pelo tabuleiro (você deve escolher), mas é o tabuleiro quem executa a remoção da peça.

- Tabuleiro:
 - o tabuleiro tem 64 casas (8 linhas e 8 colunas);
 - cada lado tem 12 peças, posicionadas conforme a ilustração:



- Lance:
 - o jogo está organizado em rodadas (lances) em que os oponentes se revezam em movimentos no tabuleiro;
- Movimento:

- peças comuns se movem uma casa na diagonal para frente somente se a posição estiver livre e for o seu lance;
- damas se movem qualquer número de casas na diagonal para frente ou para trás, contanto que a diagonal esteja desimpedida e seja o seu lance.
- Transformação em dama: quando uma peça comum atinge o lado oposto do tabuleiro ela se transforma em uma dama.
- Captura da oponente:
 - quando uma peça é capturada ela sai do tabuleiro;
 - cada vez que uma peça comum tem uma peça da cor oponente na sua casa vizinha diagonal (para frente ou para trás), seguida de uma casa vazia na mesma diagonal (destino), ela deve capturar a oponente se for seu lance - a captura envolve o movimento para o destino;
 - cada vez que uma dama tem uma peça da cor oponente (peça alvo) na sua diagonal (para frente ou para trás), havendo pelo menos um espaço livre na mesma diagonal depois da peça alvo (destino), ela deve capturar a oponente se for o seu lance - o caminho entre a origem e destino do movimento da dama só pode ter a peça alvo e a dama se move para o destino na captura;
 - mais de uma captura pode ser feito no mesmo lance.

Os movimentos a serem jogados no tabuleiro serão recebidos a partir da chamada de um método de um objeto, cuja classe já está codificada (veja detalhamento abaixo). O estado inicial do tabuleiro e seu novo estado depois de cada movimento deve ser mostrado no console na forma de caracteres, como ilustrado a seguir:

```

8 - P - P - P - P
7 P - P - P - P -
6 - P - P - P - P
5 - - - - - - -
4 - - - - - - -
3 B - B - B - B -
2 - B - B - B - B
1 B - B - B - B -
  a b c d e f g h

```

Os caracteres P representam as peças pretas, B são as brancas e - é uma casa vazia. A ilustração também mostra o estado inicial do tabuleiro.

Tabuleiro:

```

8 - P - P - P - P
7 P - P - P - P -
6 - P - P - P - P
5 - - - - - - -
4 - - - - - - -

```

```

3 B - B - B - B -
2 - B - B - B - B
1 B - B - B - B -
  a b c d e f g h

```

O tabuleiro deverá ser printado da forma mostrada acima. As posições do tabuleiro serão descritas utilizando as colunas de 'a' até 'h' e as linhas de 1 até 8, onde uma posição é dada combinando coluna com linha, por exemplo, **c4** consiste na coluna **c** e na linha **4**.

Entrada do Programa:

A entrada do programa será um arquivo `.csv` contendo todos os comandos a serem executados pelo jogo. Cada comando consistirá de uma posição inicial (posição da peça a ser movida) e uma posição final (posição para onde a peça selecionada será movida). No arquivo `.csv` os comandos serão separados por vírgulas, ou seja, cada comando, contendo posição inicial e final, será separado por vírgula do próximo. Está sendo disponibilizado uma classe (CSVReader) para ler esse arquivo `.csv` e retornar a entrada pronta num vetor de String, onde cada posição desse vetor consiste da posição inicial e da posição final separadas por ":".

Exemplo de uma posição do vetor contendo as entradas: **f4:d4**. Nesse exemplo, a posição inicial é a coluna **f** e a linha **4** e a posição final é a coluna **d** e a linha **4**.

A entrada usará o mesmo CSVReader da etapa anterior do trabalho.

Saída do Programa:

A saída do programa deve ser mostrada no console e segue o mesmo princípio da etapa anterior do trabalho. Deve ser mostrada a posição inicial (source) e final (target) da peça que vai ser movimentada na rodada, bem como o estado do tabuleiro após a movimentação. Antes da primeira movimentação mostre o estado inicial do tabuleiro.


Entrega

A entrega deve ser realizada via Github. Vocês devem submeter o código-fonte no seu Github e enviar, pela atividade do Classroom, o link do seu repositório no Github. A imagem abaixo mostra como enviar um link na atividade:

- 1) Entre na Atividade;
- 2) Clique em "Adicionar ou criar";
- 3) Link

Seus trabalhos

Pendente

 ir-atoms.ipynb
Arquivo binário

×

+ Adicionar ou criar



Google Drive



Link



Arquivo