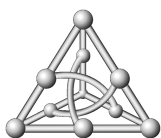


Prova 3

Algoritmos e Programação Orientada a Objetos I — 2015

Instruções para a realização da prova:

1. A prova contém **1 questão prática**, totalizando **10 pontos**, e tem duração de **30 minutos**.
2. A prova pode ser realizada **individualmente** ou em **dupla**;
3. **Crie uma pasta com o seu nome completo** dentro da pasta **Documentos**, e salve todos os arquivos **dentro dessa pasta**. Apenas o que for feito dentro dessa pasta será corrigido, pois é a única pasta que fica disponível posteriormente para o professor.
4. **Não altere a disposição do mobiliário da prova**; durante a prova, você usará um ambiente propício para a realização da mesma; é proibido reposicionar computador e monitor na prova;
5. **Coloque seu celular** em cima do gabinete, desligado. Caso o professor ouça o celular vibrando ou tocando durante a aula, o aluno receberá **nota zero**;
6. **Faça a prova em silêncio**; não converse durante a prova;
7. **Não tente plagiar a prova do(a) seu(sua) colega**, você pode prejudicar você e seu(sua) colega;
8. **Não é necessário verificar a entrada**; isto é, se seu programa solicita que o(a) usuário(a) informe um número inteiro e o(a) usuário(a) informa uma letra, um símbolo de pontuação, ou qualquer outra coisa diferente de um número, seu programa pode ter qualquer comportamento inesperado, tal como abortar abruptamente sua execução com respostas erradas, queimar o processador do computador, explodir o laboratório, etc;
9. **Não é necessário enviar mensagens amistosas para o(a) usuário(a)**; ou seja, se a questão solicita que o(a) usuário(a) informe um valor que será armazenado na variável *n*, não é necessário enviar uma mensagem antes da leitura tal como **Informe n**: ou qualquer outra semelhante; esta observação vale para qualquer informação que deve ser lida/informada pelo(a) usuário(a).
10. **Utilize apenas o que foi ensinado em sala de aula**. O uso de qualquer estrutura de programação ou estrutura de dados que não foi ensinada em sala de aula anulará a sua questão.
11. A interpretação dos exercícios também faz parte do processo de avaliação. Apenas se você achar que há um erro na descrição da questão, levante a sua mão e, em voz alta, faça a pergunta.



Mau-mau

1 Introdução

Um **baralho** é um conjunto de cartas usualmente utilizadas em jogos de mesa. Um baralho comum é composto por 52 cartas, e cada carta possui um **valor** e um **naipe**. Os possíveis valores e naipes de uma carta estão listados a seguir:

- *Valor*: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A;
- *Naipe*: ouros (O), espadas (E), copas (C) e paus (P).

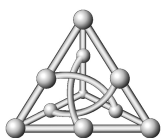
O seu objetivo neste prova é implementar, na linguagem Java, um jogo conhecido como **mau-mau**. O jogo tem de 2 a 8 participantes, que jogam de forma individual, e é muito semelhante ao jogo de cartas *Uno*.

2 Descrição do Jogo

O jogo deve ser jogado na seguinte sequência de passos:

1. Embaralhe o baralho;
2. Distribua 5 cartas do baralho para cada um dos jogadores;
3. Após a distribuição, as cartas que restaram a partir de agora são chamadas de *monte disponível*;
4. Pegue a carta do topo do *monte disponível* e coloque-a como primeira carta do *monte descartado*;
5. O jogador da vez deve descartar no topo do *monte descartado* uma carta do mesmo naipe ou mesmo número da carta que estava no topo do *monte descartado* e passar a vez para o próximo jogador;
6. Quando o jogador da vez não possuir uma carta adequada para descartar (não possui nenhuma carta com o mesmo naipe ou número da carta que está no topo do *monte descartado*), deve comprar uma carta do topo do *monte disponível* e, se essa carta puder ser descartada e o jogador quiser descartá-la, ele pode fazer. Caso contrário, a vez é passada para o próximo jogador;
7. O jogo finaliza quando algum jogador conseguir eliminar todas as cartas da mão.

Além das regras básicas apresentadas anteriormente, o *mau-mau* possui algumas regras adicionais, que são elas:

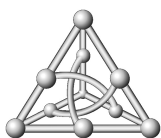


- O sentido inicial do jogo é o sentido horário;
- A compra de cartas pode ser feita mesmo se o jogador possuir alguma carta que possa ser descartada. Pode-se comprar no máximo 1 carta por vez;
- As cartas de efeito são cartas que, quando descartadas, provocam algum efeito a algum jogador ou a certas características do jogo. São elas:
 - *A*: faz com que o próximo jogador não jogue a rodada, pulando a sua vez;
 - *Q*: inverte o sentido do jogo. Se o sentido era horário, o sentido passa a ser anti-horário e vice-versa;
 - *J*: o jogador que a descartou tem o direito de escolher um naipe, sendo que o próximo jogador deve descartar uma carta do naipe escolhido ou outro valete. O valete pode ser descartado mesmo se o naipe e/ou o número da carta no topo do *monte descartado* não forem o mesmo;
 - 9: quando um nove é descartado, o jogador anterior compra 1 carta.

3 Implementação

O seu programa deve consistir, basicamente, das seguintes telas:

- *Tela de entrada*: o usuário deve escolher a quantidade de jogadores. Logo após, a *tela principal do jogador* é exibida;
- *Tela principal do jogador*: nessa tela, você deve exibir qual o jogador da vez, quais as suas cartas, qual a carta de que está no topo do *monte descartado* e qual a quantidade de cartas que os outros jogadores têm em mãos. Ainda nessa tela, você deve permitir com que o jogador da vez escolha uma das seguintes opções:
 1. *Descartar uma carta*: se o jogador atual escolher essa ação, seu programa deve pedir para que o jogador escolha qual das suas cartas será descartada. Caso ele escolha uma carta que não obedeça às regras do jogo, seu programa deve voltar à tela principal do jogador e aguardar que o jogador escolha novamente uma opção. Além disso, caso ele descarte sua última carta, informe que ele foi o vencedor e encerre o programa;
 2. *Comprar uma carta*: se o jogador escolher essa ação, seu programa deve retirar uma carta do topo do *monte disponível* e colocar na mão do jogador atual. Após essa ação, o jogador atual pode decidir descartar uma carta comprada no *monte descartado* ou passar a vez. Se o *monte disponível* ficar vazio, seu programa deve retirar todas as cartas do *monte descartado* (exceto a carta do topo), colocá-las no *monte disponível* e embaralhá-las;



4 Informações Adicionais

Seu programa deve utilizar os conceitos de orientação a objetos e encapsulamento, devendo conter, pelo menos, as seguintes classes:

- Carta, que modela uma carta do baralho;
- Mao, que modela as cartas que estão na mão de um jogador;
- MauMau, que possui o método *main*.

Abaixo segue um exemplo de método que pode ser utilizado para embaralhar um vetor de qualquer tipo, onde o tipo *T* deve ser substituído com o tipo de seu interesse.

```
public void embaralha(T vet[]) {  
    Random gerador = new Random();  
    int pos1, pos2, length = vet.length;  
    T aux;  
    for(int i = 0 ; i < length ; i++) {  
        pos1 = gerador.nextInt(length);  
        pos2 = gerador.nextInt(length);  
  
        aux = vet[pos1];  
        vet[pos1] = vet[pos2];  
        vet[pos2] = aux;  
    }  
}
```

Importe a classe *java.util.Random* na classe que inserir esse método.

BOA PROVA!