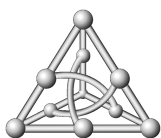


Prova Optativa 1

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 **1 hora e 40 minutos**.
2. A prova é **individual e sem consulta**;
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.



1. [Prática][10 pontos]

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 são listados a seguir, já em ordem crescente de importância (do menos valioso para o mais valioso):

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

Em um jogo de pôquer, a mão de um jogador é composta por 5 cartas, que podem formar jogos de diferentes valores. Os possíveis jogos em um jogo de pôquer são listados a seguir, já em ordem crescente de importância (do menos valioso para o mais valioso):

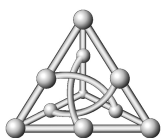
- Um par**: duas cartas do mesmo valor;
- Dois pares**: dois diferentes pares;
- Três de um valor**: três cartas de um mesmo valor;
- Sequência**: todas as cartas de valores consecutivos (respeitando a ordem crescente de importância);
- Flush**: todas as cartas do mesmo naipe;
- Full house**: um par e três de um valor;
- Quatro de um valor**: quatro cartas de um mesmo valor;
- Straight flush**: todas as cartas de valores consecutivos (respeitando a ordem crescente de importância) e de mesmo naipe;
- Royal flush**: *D*, *J*, *Q*, *K* e *A*, do mesmo naipe.

Caso os jogadores não possuam nenhum dos jogos listados acima, o jogador que possuir a carta de maior valor ganha a rodada.

Para simplificar a implementação, assuma que **nunca haverá empate**. Ou seja, dois jogadores nunca terão jogos de mesma importância.

A tabela a seguir mostra um exemplo de uma sequência de 4 rodadas jogadas por dois jogadores.

Rodada	Jogador 1	Jogador 2	Ganhador
1	2E 9O AO AC AP (três de <i>A</i>)	3O 6O 7O DO QO (flush de <i>ouros</i>)	Jogador 2
2	4P 6O QE QO QE (três de um valor)	3P 6O 7E JE JP (par de <i>J</i>)	Jogador 1
3	2O 2P 4E 4C 4P (<i>full house</i>)	3O 3E 3P 3C 9P (quatro de um valor)	Jogador 2



Escreva um programa em Java que recebe um valor n referente à quantidade de casos de testes. Logo após, para cada caso de teste, seu programa deve receber um conjunto de 10 cartas, que representam uma rodada de pôquer. As 5 primeiras cartas lidas são do Jogador 1 e as próximas 5 cartas são do Jogador 2. Assuma que todas as cartas são válidas (não há caracteres inválidos ou cartas repetidas), que as cartas nas mãos dos jogadores estão em ordem crescente de valor.

Para cada entrada, seu programa deve escrever na tela o nome do jogador que ganhou a rodada (1 ou 2).

Exemplo de entrada:

```
3
2E 9O AO AC AP 3O 6O 7O DO QO
4P 6O QE QO QE 3P 6O 7E JE JP
2O 2P 4E 4C 4P 3O 3E 3P 3C 9P
```

Exemplo de saída:

```
2
1
2
```

Observações adicionais

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

- Carta, que modela a carta de um baralho;
- Mao, que modela as cartas que estão na mão de um jogador. Essa classe deve possuir pelo menos os seguintes métodos: *umPar*, *doisPares*, *tresDeUmValor*, *sequencia*, *flush*, *fullHouse*, *quatroDeUmValor*, *straightFlush* e *royalFlush*. Todos esses métodos retornam um valor booleano que indica se o jogo em questão acontece na mão do jogador;
- JogoDePoquer, que possui um método main.

Sugestões

- Cada carta pode ser lida utilizando o método next da classe Scanner, que a armazenará como uma String;
- Separe o valor e o naipe de uma carta utilizando o método charAt da classe String.