

Poker Hands

(Dificuldade: Alta)



19/05/2017

Um famoso cassino de repente enfrenta um grande declínio de sua receita. Eles decidem oferecer uma versão online de Poker. Pode ajudá-los escrevendo um algoritmo para rankear mãos de poker?

1 Tarefa

- Crie uma classe que represente uma mão de poker (PokerHand) e nela crie um método para compará-la com outra mão de poker.

```
Result result = pokerHand.compareTo(anotherPokerHand)
```

- O resultado deve ser um enum com os resultados: WIN e LOSS.
- A classe PokerHand deverá ter um construtor que aceite uma String contendo 5 cartas:

```
PokerHand pokerHand = new PokerHand("KS 2H 5C JD TD")
```

- As características da String de cartas são:
 - Um espaço é usado como separador de cada carta
 - Cada carta consiste em dois caracteres
 - O primeiro caracter é o valor da carta. Caracteres válidos são:
 - * 2
 - * 3
 - * 4
 - * 5
 - * 6
 - * 7
 - * 8
 - * 9
 - * T (10)
 - * J (Valete)

- * Q (Rainha)
- * K (Rei)
- * A (Ace)
- O segundo caracter é o naipe da carta. Caracteres válidos são:
 - * S (Espadas)
 - * H (Copas)
 - * D (Ouros)
 - * C (Paus)
- As regras de ordenação de cartas do Poker estão listadas na figura abaixo.
- **BÔNUS (Opcional):** Tornar possível ordenar uma lista de mãos de poker.



2 Testes

O funcionamento da solução deve ser garantida através de testes unitários. A seguir encontra-se um código em Java com as comparações e seus resultados. Essas são as comparações mínimas que devem ser feitas para garantir o funcionamento. Utilize esse código como base da implementação dos testes unitários.

Dica: Implemente os testes antecipadamente.

```
new PokerHand("TC TH 5C 5H KH").compareTo(new PokerHand("9C 9H 5C 5H AC")) == Result.WIN;
new PokerHand("TS TD KC JC 7C").compareTo(new PokerHand("JS JC AS KC TD")) == Result.LOSS;
new PokerHand("7H 7C QC JS TS").compareTo(new PokerHand("7D 7C JS TS 6D")) == Result.WIN;
new PokerHand("5S 5D 8C 7S 6H").compareTo(new PokerHand("7D 7S 5S 5D JS")) == Result.LOSS;
new PokerHand("AS AD KD 7C 3D").compareTo(new PokerHand("AD AH KD 7C 4S")) == Result.LOSS;
new PokerHand("TS JS QS KS AS").compareTo(new PokerHand("AC AH AS AS KS")) == Result.WIN;
new PokerHand("TS JS QS KS AS").compareTo(new PokerHand("TC JS QC KS AC")) == Result.WIN;
new PokerHand("TS JS QS KS AS").compareTo(new PokerHand("QH QS QC AS 8H")) == Result.WIN;
```

```

new PokerHand("AC AH AS AS KS").compareTo(new PokerHand("TC JS QC KS AC")) == Result.WIN;
new PokerHand("AC AH AS AS KS").compareTo(new PokerHand("QH QS QC AS 8H")) == Result.WIN;
new PokerHand("TC JS QC KS AC").compareTo(new PokerHand("QH QS QC AS 8H")) == Result.WIN;
new PokerHand("7H 8H 9H TH JH").compareTo(new PokerHand("JH JC JS JD TH")) == Result.WIN;
new PokerHand("7H 8H 9H TH JH").compareTo(new PokerHand("4H 5H 9H TH JH")) == Result.WIN;
new PokerHand("7H 8H 9H TH JH").compareTo(new PokerHand("7C 8S 9H TH JH")) == Result.WIN;
new PokerHand("7H 8H 9H TH JH").compareTo(new PokerHand("TS TH TD JH JD")) == Result.WIN;
new PokerHand("7H 8H 9H TH JH").compareTo(new PokerHand("JH JD TH TC 4C")) == Result.WIN;
new PokerHand("JH JC JS JD TH").compareTo(new PokerHand("4H 5H 9H TH JH")) == Result.WIN;
new PokerHand("JH JC JS JD TH").compareTo(new PokerHand("7C 8S 9H TH JH")) == Result.WIN;
new PokerHand("JH JC JS JD TH").compareTo(new PokerHand("TS TH TD JH JD")) == Result.WIN;
new PokerHand("JH JC JS JD TH").compareTo(new PokerHand("JH JD TH TC 4C")) == Result.WIN;
new PokerHand("4H 5H 9H TH JH").compareTo(new PokerHand("7C 8S 9H TH JH")) == Result.WIN;
new PokerHand("4H 5H 9H TH JH").compareTo(new PokerHand("TS TH TD JH JD")) == Result.LOSS;
new PokerHand("4H 5H 9H TH JH").compareTo(new PokerHand("JH JD TH TC 4C")) == Result.WIN;
new PokerHand("7C 8S 9H TH JH").compareTo(new PokerHand("TS TH TD JH JD")) == Result.LOSS;
new PokerHand("7C 8S 9H TH JH").compareTo(new PokerHand("JH JD TH TC 4C")) == Result.WIN;
new PokerHand("TS TH TD JH JD").compareTo(new PokerHand("JH JD TH TC 4C")) == Result.WIN;

```