



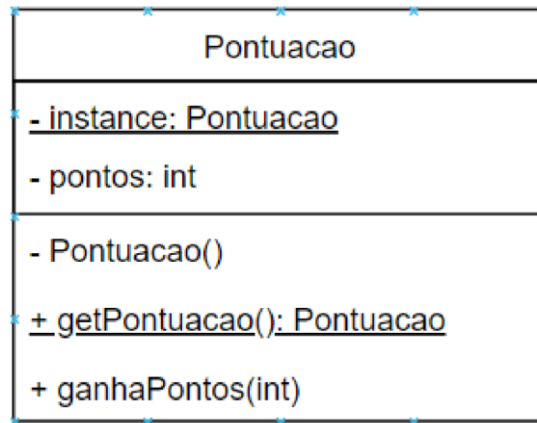
Sexta lista de exercícios de Programação Orientada a Objetos

Questões teóricas

1. Diferencie classe abstrata e interface.
2. Para que servem os tipos genéricos no Java?
3. O que são padrões de projeto? Qual sua importância em POO?
4. Explique o padrão Singleton com suas palavras e destaque o que é preciso fazer na classe para ela se tornar um Singleton.
5. Explique o padrão de projeto Factory destacando a principal vantagem de se usar esse padrão e os cuidados que o programador precisa ter.

Questões práticas

6. Crie a classe Carta, a qual terá um atributo do tipo **enum**) correspondente ao seu número (ÁS, DOIS, TRES, ... , DAMA, REI), e também um atributo Naipe. **O Naipe será uma Inner Class da Carta**, e terá um único atributo, também do tipo **enum** (OUROS, ESPADAS, PAUS, COPAS).
7. Crie a Interface Printavel, com um método público String imprimir(), o qual deverá ser implementado na Classe Carta. Nela, **retornar um texto** com o número e o naipe da carta, no formato "<número> de <naipe>" (ex.: "6 de ouros");
8. Crie a classe FactoryCarta pra gerar uma carta aleatoriamente. Na driver class, crie um ArrayList de cartas, que será a "mão" do jogador e inclua no array 5 cartas geradas aleatoriamente pela factory.
9. Ainda na driver class, imprima as cartas da mão do jogador (pelo método da interface que você implementou!).
10. Dada a classe Pontuacao abaixo, implementar seu Singleton. A pontuação inicial deve ser 0. Implemente também o método ganhaPontos, que somará o valor de entrada à pontuação atual.



11. Teste sua corretude na driver class: verifique se entre chamadas distintas de `getPontuacao()` os pontos de fato se acumulam.
12. Tente instanciar uma nova `Pontuacao` diretamente na driver class (`Pontuacao p = new Pontuacao();`). O que acontece?