u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

CID EINF MA BQ BIOTEC

Disciplina: Programação II-2012/2013 Prova: Teste1 (10-04-2013)

Este teste tem a duração de **2 horas** e é **sem consulta**.

1. Implemente a classe *CartaJogar*. Esta classe deve providenciar um implementação para representar as cartas de um baralho de jogo convencional. Uma carta é de um dos quatro naipes possíveis:copas ♡; paus ♣; espadas ♠ e ouros ♢ e de um dos seguintes valores 2, 3, 4, 5 , 6, 7, 8, 9, 10, D, J, K, A.

   (a) Crie um construtor que dados dois inteiros, um entre 2 e 14 para o valor da carta e outro inteiro entre 1 e 4 para o naipe construa a respectiva carta. Use a ordem para os valores e para os naipes dada em 1

   (b) Apresente implementações para os métodos:

      i. selectores para saber o valor e o naipe da cartas;
      ii. `toString()`, que retorne uma representação do tipo 2_*Paus* para o dois de paus, por exemplo ou *D_Copas* para a dama de copas
      iii. `equals(Object x)`, que retorne true se duas cartas são iguais e false caso contrário
      iv. `compareTo(Carta y)`, que retorne o valor -1 se a instância é menor que y, 0 se são iguais e 1 se a instância é maior que y, e se as cartas são do mesmo naipe. No caso das cartas serem de naipes diferentes devolva 100. Use novamnete a ordem apresentada em 1 para a comparação das cartas.
      v. boolean `vermelha`(), que retorne true se é uma carta de ouros ou de copas
      vi. boolean `preta`(), que retorna true se a carta é de paus ou de espadas
      vii. `baralho()`, que retorna um array com as 52 cartas do baralho. Este método deve ser aplicado à classe. Por exemplo *CartaJogar.baralho()* deve retornar um array com todas as cartas dum baralho.

2. Implemente a classe *MaoJogo*. Uma mão de jogo é constituída por um determinado número de cartas de jogar. Defina para esta classe, além das variáveis de classe e variáveis de instância:

   (a) Um construtor que recebe um inteiro correspondente ao número de cartas da mão de jogo;

   (b) o método `void add(CartaJogar c)` que adiciona a carta passada no argumento à mão de jogo;

   (c) o método `void jogar(CartaJogar c)` que retira da mão de jogo a carta c;

   (d) o método `CartaJogar maisBaixa(int naipe)` que retorna a carta mais baixa do naipe do argumento que existe na mão de jogo. Se não existir nenhuma carta do mesmo naipe o método deverá retornar null.

3. Suponha que pretende implementar uma hierarquia de classes que represente as Cartas de Jogar. Apresente um esboço da hierarquia indicando as relações de parentesco para as classes da hierarquia. Indique para cada classe as respectivas variáveis e a assinatura dos métodos que considere relevantes nesta categorização. Deve ser possível usando esta hierarquia construir a carta "2Espadas" fazendo `new Espadas(2)`, ou criar o "AsCopas" fazendo `new Copas(14)`;