

## MAC 110/115 – Introdução à Ciência da Computação – 1º/2008

### Primeiro Exercício-Programa: JOGO DE TRUCO

**Data de entrega: até 11 de abril de 2008 às 23h55, pelo sistema PACA.**

*Professores:* Marcelo Finger, Marcelo Queiroz e Roberto Marcondes

Neste exercício-programa, você irá criar um programa de computador que joga truco contra um usuário. Não se preocupe: o seu programa não deverá ser um exímio jogador de truco; quando muito ele terá habilidade comparável a uma pessoa que acabou de aprender as regras do jogo. Mesmo assim, será divertido!

#### Regras do truco

As regras oficiais do truco podem ser encontradas em diversos sítios da Internet, como:

- <http://www.rksoft.com.br/html/truco.htm>
- <http://www.guaru.com.br/trucorato/regras.asp>
- <http://www.megajogos.com.br/?page=regra&game=truco>

Para facilitar a implementação, utilizaremos uma versão simplificada do jogo de truco. O baralho será formado por cartas de apenas um naipe, numeradas de 1 a 10, com cartas de maior número possuindo maior valor. Existe uma quantidade ilimitada de cartas de cada número, de modo que os jogadores podem receber cartas contendo números iguais. Além disso haverá apenas dois jogadores, o usuário humano e o computador.

No começo de um jogo, cada jogador recebe três cartas aleatórias. O jogo possui três rodadas. Por convenção o usuário humano começa a partida. Em cada rodada, o usuário deve colocar uma carta na mesa, e será seguido pela jogada do computador. Aquele que colocar a maior carta ganha a rodada. No caso de empate, o vencedor é decidido aleatoriamente. Quem conseguir vencer duas rodadas ganha a partida.

## 1 Implementação

Neste EP você deverá fazer um programa que permita a um usuário jogar truco com o computador seguindo as regras acima. Para isso você deverá criar uma classe Java chamada `Truco.java` que simula uma partida de Truco. Esta classe deve *obrigatoriamente* possuir os seguintes métodos:

- `iniciaJogo()`. Distribui três cartas para cada jogador e inicia um novo jogo;
- `jogaCarta()`. O jogador coloca na mesa de jogo uma de suas cartas.

### 1.1 Distribuição das Cartas

No interior da classe, as cartas serão representadas por um inteiro entre 1 e 10. A classe `Truco.java` conterá 6 inteiros representando as 3 cartas de cada jogador.

Para distribuir as cartas aos jogadores, utilizaremos o gerador de número aleatórios do Java. Para tal, será necessário utilizar os seguintes comandos<sup>1</sup>:

```
// cria um objeto da classe Random, que possui métodos que
// permitem a geração de números aleatórios

java.util.Random gerador = new java.util.Random();
```

<sup>1</sup>Obs: A chamada do método `nextInt(i)` gera aleatoriamente um inteiro entre 0 e  $i-1$ , com distribuição uniforme.

```
// gera um número entre 1 e 10  
  
int carta = gerador.nextInt( 10 ) + 1;
```

## 1.2 Calculando o resultado de uma jogada

Quando o jogador (usuário) chama o método `jogaCarta` com um argumento inteiro correspondendo a uma carta que está na sua mão, significa que ele colocou a carta escolhida na mesa. Seu programa deve verificar que o usuário tem de fato esta carta em sua mão, e tratar um eventual erro de forma apropriada (por exemplo, imprimindo uma mensagem e dando outra chance ao jogador, ou imprimindo uma mensagem e encerrando a execução).

A estratégia (simplista) do computador corresponderá a jogar uma carta de sua mão que seja maior que a carta que o usuário colocou na mesa. Se o computador não possuir uma carta maior que a colocada na mesa, ele joga uma carta menor.

O método deve então calcular o resultado da jogada, informando ao usuário o placar atualizado.

No final da rodada, as cartas utilizadas tanto pelo usuário quanto pelo computador não podem mais ser utilizadas naquele jogo. DICA: As cartas já utilizadas podem ser representadas pelo número 0 (zero).

## 1.3 Final da partida

No interior da classe `Truco`, o placar será representado por dois atributos inteiros, representando o número de rodadas que cada jogador ganhou. O final do jogo ocorre quando um dos jogadores ganhar duas rodadas (mesmo que isso ocorra antes de três rodadas). Quando esta condição ocorrer, o programa deve declarar o fim da partida e informar o vencedor. Em caso de empate, o vencedor será determinado aleatoriamente. Para tal, pode-se utilizar a classe `Random` do Java.

## 2 Exemplo de execução

Um exemplo do que poderia ser apresentado no painel de interações (você pode apresentar as informações como quiser, desde que as mensagens sejam claras):

```
Welcome to DrJava.  
> Truco jogo = new Truco();  
> jogo.iniciaJogo();  
Você recebeu as cartas 7, 9, 3.  
Informe sua jogada:  
> jogo.jogaCarta(7);  
O computador jogou a carta 8.  
O computador venceu esta rodada.  
Placar: computador 1 x 0 usuário.  
Informe sua jogada:  
> jogo.jogaCarta(9);  
O computador jogou a carta 5.  
O usuário venceu esta rodada.  
Placar: computador 1 x 1 usuário  
Informe sua jogada:  
> jogo.jogaCarta(3);  
O computador jogou a carta 5.  
O computador venceu esta rodada.  
Placar: computador 2 x 1 usuario  
Fim da partida. Você perdeu...
```

## 3 Observações importantes

### 3.1 Sobre a elaboração:

Este EP poderá ser elaborado por equipes de dois alunos, desde que sejam respeitadas as seguintes regras:

- Os alunos devem trabalhar sempre juntos: a idéia é que deve existir uma cooperação;
- Caso em um grupo exista um aluno com maior facilidade, este deve explicar as decisões tomadas. E o seu par deve participar e se esforçar para entender o desenvolvimento do programa (chamamos isso de *programação pareada*, que é uma excelente prática que vocês devem se esforçar para adotar);
- Mesmo a digitação do EP deve ser feita em grupo: enquanto um digita, o outro acompanha o trabalho e participa das decisões.

Recomendamos fortemente que o exercício seja desenvolvido da forma descrita nesta observação. No entanto, se desejado, o exercício também poderá ser feito individualmente.

### 3.2 Sobre a avaliação:

- No caso de exercícios feitos em dupla, a mesma nota da correção será atribuída aos dois alunos do grupo;
- Não serão toleradas cópias! Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO (inclusive o original);
- Exercícios atrasados não serão aceitos;
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO;
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A qualidade do seu trabalho sob esse ponto de vista influenciará sua nota!
- As informações impressas pelo seu programa na tela devem aparecer da forma mais clara possível. Este aspecto também será levado em consideração no cálculo da sua nota;
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor será a disposição dele para dar-lhe uma nota generosa.

### 3.3 Sobre a entrega:

- O prazo de entrega é o dia 11/4/2008, às 23h55. Não será possível o envio de arquivos após este horário.
- O nome do arquivo da sua classe deve ser `Truco.java`.
- No início do arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```
/*
*****
/**  MAC 110 - Introdução à Computação          **/
/**  IME-USP - Primeiro Semestre de 2005      **/
/**  <turma> - <nome do professor>            **/
/**                                           **/
/**  Primeiro Exercício-Programa  --  Truco    **/
/**  Arquivo: Truco.java                  **/
/**                                           **/
/**  <nome do(a) aluno(a)>                   <número USP>  **/
/**  <nome do(a) aluno(a)>                   <número USP>  **/
/**                                           **/
/**  <data de entrega>                      **/
*****
*/
```

Não é obrigatório que o cabeçalho seja idêntico a esse, apenas que contenha (pelo menos) as mesmas informações.

- Junto com o programa (no final do arquivo), deve ser entregue uma simulação de 5 jogos, similar à apresentada na Seção “Exemplo de execução”.
- Não é necessário usar mais de uma classe para a realização do EP. No entanto, se você quiser usar mais de uma classe, todas deverão estar contidas no mesmo arquivo `Truco.java`. Se quiser, você poderá implementar outros métodos auxiliares além dos métodos obrigatórios `iniciaJogo()` e `jogaCarta()`.
- As simulações devem vir ao final do arquivo, na forma de comentários. Indique claramente o final de uma simulação e o início da próxima.
- Para a entrega, utilize o moodle em `paca.ime.usp.br`. Para enviar o programa, você precisa ter um cadastro de usuário e estar inscrito na página do curso. A inscrição é feita no link “Faça a minha inscrição neste curso”, que aparece na coluna da esquerda, na caixa “Administração”. O curso deve aparecer na caixa “Meus cursos”, logo abaixo.
- Você pode entregar várias versões de um mesmo EP até o prazo, mas somente a última será armazenada pelo sistema. Encerrado o prazo, o sistema não aceitará mais os EP’s. Os procedimentos de entrega para trabalhos individuais e em grupo diferem um pouco, consulte a ajuda do moodle para obter mais detalhes.
- Guarde uma cópia eletrônica do seu EP pelo menos até o fim do semestre!

## 4 Desafio!

Os desafios NÃO são obrigatórios, NÃO valem nota e NÃO devem ser entregues junto com o EP. Eles são apenas para aqueles que tiverem terminado o EP e quiserem desafiar e melhorar sua habilidade de programação.

1. No truco, existe o conceito da manilha, que é uma carta que possui o maior valor entre todas as demais. No começo de cada jogo, uma carta aleatória é retirada do baralho, de modo que a carta imediatamente superior à carta retirada passa ser a manilha. Adicione a manilha ao seu jogo de truco.
2. Em nosso EP, é sempre o usuário que inicia cada rodada. Elimine esta limitação, de modo que o jogador da primeira rodada seja escolhido alternadamente. O primeiro a jogar nas demais rodadas da partida é o vencedor da rodada anterior.
3. Nos jogos de truco estamos acostumados às cartas do baralho 4, 5, 6, 7, Q, J, K, A, 2 e 3. Mas em nossa versão, utilizamos números de 1 a 10. Modifique sua classe truco de modo que o usuário veja as cartas reais do truco ao invés de números entre 1 e 10.
4. Adicione os quatro naipes do baralho ao jogo de truco.

**Bom Trabalho!**