

Exercício-Programa Zero - BCC
Data de entrega: 18/04/2006
Local de entrega: <http://paca.ime.usp.br/>

JoKenPo

- 1 Introdução
 - 1.1 O que é JoKenPo?
 - 1.2 Regras do jogo
 - 2 O Exercício
 - 2.1 O que é pra fazer
 - 2.2 Implementação
 - 2.3 Exemplo de execução
 - 3 Observações
 - 3.1 Elaboração
 - 3.2 Entrega
 - 3.3 Avaliação
-

1 Introdução

1.1 O que é JoKenPo?

Se você sabe o que é, pode pular esta parte :-).

JoKenPo É uma brincadeira semelhante a “2 ou 1” porém com 3 possibilidades (muito mais complexo!). As opções são: Papel (mão aberta), pedra (mão fechada) ou tesoura (dois dedos). Quem sabe depois de fazer este EP você não pensa em implementar um “Alex Kidd in Miracle World” (jogo da Sega: <http://www.sega.com/>)?

1.2 Regras do jogo

- Cada jogador escolhe sua opção secretamente, alguém conta até 3 e no “já” todos mostram suas opções.
- Papel ganha de Pedra e perde pra Tesoura
- Pedra ganha de Tesoura e perde pra Papel
- Tesoura ganha de Papel e perde pra Pedra

2 O Exercício

2.1 O que é pra fazer

Usando o DrJava, deve-se criar **uma** classe que implemente JoKenPo e uma simulação de **n** partidas.

2.2 Implementação

- Um jogo composto de partidas. Ganha o primeiro que conseguir ganhar a maioria de **n** partidas
- Cada partida vale um ponto. Derrota não perde pontos.
- Em caso de empate em uma partida, nem o jogador nem o computador devem ganhar ponto.
- Papel = 1, Pedra = 2 e Tesoura = 3.

O nome da classe java deve ser JoKenPo (arquivo **JoKenPo.java**) e deve conter obrigatoriamente o seguinte:

✓ Atributos

- Opção do jogador
- Opção do computador
- Pontos do jogador
- Pontos do computador
- Número de partidas

✓ Métodos

- `void novoJogo(n)`: Cria um novo jogo com **n** partidas (somente aceita **n** ímpar);
- `void escolhaDoJogador(int escolha)`: Recebe a opção “escolha” do jogador.
 - (1 = Papel, 2 = Pedra e 3 = Tesoura);
- `void escolhaDoComputador()`: Computador escolhe sua opção aleatoriamente;
- `void quemGanha()`: É o “já” do jogo;
- `void imprimirPlacar()`: Imprime o resultado da última rodada, os pontos totais do jogador e do computador;
- `void proximaPartida()`: Prepara o jogo para a próxima partida.
- `boolean verificarTerminoDeJogo()`: Verifica se alguém conseguiu a maioria de **n** pontos (isto é, $\lceil n/2 \rceil$). Retorna ‘true’ se verdadeiro, ‘false’ caso contrário. Exemplo: Se $n = 5$, ganha o primeiro que conseguir 3 pontos. É o tal melhor de 5.

Métodos ou atributos adicionais são permitidos (e, talvez, necessários).

✓ Detalhes

- O método `quemGanha()` deverá conferir se os dois jogadores já jogaram (i.e. chamaram os métodos `escolhaDo...()`), caso contrário mostra mensagem de aviso;
- O método `proximaPartida()` deverá conferir se o método `quemGanha()` já foi chamado, e também conferir se o jogo já acabou e mostrar mensagem de término se for o caso (use o método `verificarTerminoDeJogo()` para isso).
- Lembre-se: o número de partidas recebido por `novoJogo(int n)` deverá ser ímpar.

Dicas:

Para implementar o método `escolhaDoComputador()` utilize o gerador de números aleatórios do Java:

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

int numeroAleatorio = gerador.nextInt(3) + 1; // 1 + número aleatório
entre 0 e 2 => [1-3]
```

2.3 Exemplo de simulação

Um exemplo de simulação feito no painel de interações do DrJava:

```
JoKenPo jogo = new JoKenPo();
// Partida 1
jogo.novoJogo(3);
jogo.escolhaDoJogador(1);
jogo.escolhaDoComputador();
jogo.quemGanha();
jogo.imprimirPlacar();
jogo.proximaPartida();
```

```
// Partida 2
jogo.escolhaDoJogador(3);
jogo.escolhaDoComputador();
jogo.quemGanha(); // Deve conferir se todos jogaram
jogo.imprimirPlacar();
jogo.proximaPartida(); // Deve conferir se alguém já ganhou
```

...

3 Observações

3.1 Elaboração

Este EP pode ser elaborado por equipes de um ou 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 fica acompanhando o trabalho;
- ✓ Recomendamos fortemente que o exercício seja desenvolvido da forma descrita na observação acima, mas também pode ser feito individualmente.

3.2 Entrega

- ✓ O prazo de entrega é o dia 18/04/2006;
- ✓ O nome do arquivo da sua classe deve ser JoKenPo.java;
- ✓ No início do arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```

/*****
/** MAC 110 - Introdução à Computação      **/
/** IME-USP - Primeiro Semestre de 2006   **/
/** <turma> - <nome do professor>         **/
/** <data de entrega>                    **/
/**                                       **/
/** Primeiro Exercício-Programa - JoKenPo **/
/** Arquivo: JoKenPo.java                 **/
/**                                       **/
/** <nome do(a) aluno(a)> <número USP>     **/
/** <nome do(a) aluno(a)> <número USP>     **/
*****/

```

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

- ✓ Para a entrega, utilize o Paca (<http://paca.ime.usp.br/>). Para usá-lo, você precisa ter um cadastro (provavelmente todos os alunos já estão cadastrados). Inicialmente, a sua senha é igual ao seu número USP, você deve acessar o sistema e alterá-la o quanto antes. Entregue apenas o arquivo JoKenPo.java. 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 Paca para obter mais detalhes.
- ✓ Guarde uma cópia do seu EP pelo menos até o fim do semestre!

3.3 Avaliação

1. No caso de exercícios feitos em dupla, a mesma nota da correção será atribuída aos dois alunos do grupo;
2. Não serão toleradas cópias! Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO (inclusive o original);
3. Exercícios atrasados não serão aceitos;
4. Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO;
5. 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!
6. 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;
7. 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.

4 Interface Gráfica :-)

O jogo poderá ser visualizado através de uma interface gráfica. Para usá-la, siga os seguintes procedimentos:

- Baixe o arquivo 'JoKenPoGUI.jar' no moodle;
- Execute o DrJava, depois clique em 'Edit' > 'Preferences' > e escolha a categoria 'Resource Locations';
- Na seção 'Extra Classpath', clique 'Add' e selecione o arquivo 'JoKenPoGUI.jar' que foi baixado;
- Clique em 'Ok'.

Agora será necessário adicionar algumas linhas nos métodos de sua classe JoKenPo:

- Adicione um atributo janela na classe JoKenPo

```
Janela minhaJanela;
```

- No método novoJogo(int n), crie um objeto da classe Janela da seguinte maneira:

```
minhaJanela = new Janela();
```

A seguir você poderá chamar os seguintes métodos do objeto janela:

- void novoJogo(int n); // Mostra a janela
- void mostraEscolhas(int escolhaDoJogador, int escolhaDoComputador);
- void ajustaPlacar(int pontosDoJogador, int pontosDoComputador);
- void proximaRodada();

Exemplo: no método novoJogo da classe JoKenPo, você pode adicionar a linha `minhaJanela.novoJogo(n)`, e uma janela irá aparecer.

Agora é só escolher os locais apropriados dentro do seu código para chamar os métodos da interface gráfica.

Boa sorte!