PITÁGORAS
FACULDADE

Atividade Parcial 2

Disciplina: Algoritmos e Estruturas de Dados **Turma**: 2º e 3º Período

Data Disponibilização: 21/05/2014 **Professor**: José Alexandre Macedo

Data da Entrega: 05/06/2014

O trabalho pode ser realizado individualmente ou em grupo de 2 pessoas.

O trabalho deve ser enviado para email jamacedo@inf.ufes.br com os nomes dos integrantes do grupo e assunto "Parcial 2 – Algoritmos e Estruturas de Dados". Em anexo deve conter:

- Código fonte completo em arquivo pdf;
- Relatório com os detalhes de implementação, dizendo o que foi e o que não foi implementado. Dificuldades e facilidades.

Conta ponto:

- Código bem organizado e comentado;
- Código que funciona como especificado;
- Relatório que combina com o programa e o código fonte.

Desconta ponto:

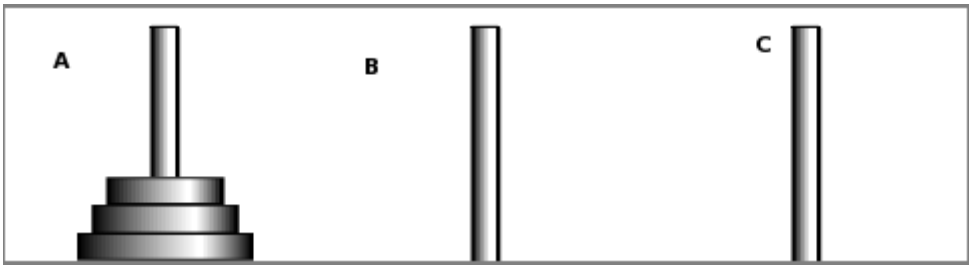
- Código que funciona de forma diferente do especificado ou do relatado, ou que não funciona;
- Relatório mal escrito ou incompleto.

Zera o trabalho:

- Cópias de trabalhos irão zerar a nota dos grupos onde houve a cópia.

Torre de Hanói

Torres de Hanói é um quebra-cabeça muito antigo e conhecido. Ele é constituído de um conjunto de N discos de tamanhos diferentes e três pinos verticais, nos quais os discos podem ser encaixados.



Cada pino pode conter uma pilha com qualquer número de discos, **desde que cada disco não seja colocado acima de outro disco de menor tamanho**. Na configuração inicial todos os discos estão no pino 1. O objetivo é mover todos os discos para o pino 3, sempre obedecendo à restrição de não colocar um disco sobre outro menor.

Objetivo do Trabalho

Criar um projeto Java que implemente o jogo da Torre de Hanói utilizando pilhas. O jogo deverá ser criado com 3 pinos, ou seja, 3 pilhas. Os arquivos necessários serão:

- Pilha.java (disponibilizado)
- No.java (disponibilizado)
- TorreHanoi.java
    - Deverá ser implementado com funções e variáveis descritas abaixo.
    - Contém 3 variáveis do tipo Pilha (uma para cada pino) e uma variável inteira totalDiscos.
    - Funções
        - **void realizaJogada(Pilha pilhaOrigem, Pilha pilhaDestino)** que movimenta um disco de uma pilha para outra, caso a jogada seja legal. Se a jogada for ilegal, o movimento não é realizado. Lembre-se, uma jogada será ilegal quando o disco do topo de destino for menor que o disco do topo de origem.
        - **boolean verificaJogo(Pilha pino1, Pilha pino2, Pilha pino3, int totalDiscos)** responsável por informar que o jogador ganhou o jogo quando o número de discos no pino 3 for igual ao número total N de discos, ou que o jogo não terminou se sobrar algum disco fora do pino 3.
        - **void exibeJogo (Pilha pino1, Pilha pino2, Pilha pino3)** capaz de exibir o conteúdo de cada pilha (altere a implementação da Pilha para isso).
- Programa.java
    - Deverá ser o arquivo executável (com **main**).
    - O número de discos deverá ser digitado pelo usuário neste arquivo.
    - Após conhecer o número de discos você deve **inicializar os pinos (empilhando discos)**, sendo que o primeiro deve conter os N discos e os demais ficam vazios.
    - Teste as funções do jogo e apresente um conjunto de movimentações onde no resultado final o jogador ganha, ou seja, a função **verificaJogo** retorna **true**.