Pilhas Dinâmicas – Lista 2 Estrutura de Dados
Prof. Leandro Colevati FATEC ZL

Suporte para GitHub:

Carregando arquivos pelo navegador: https://drive.google.com/open?id=1Klf0HCJcB_4O5B7efMXrs_YYDXGwH89U
Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH: http://www.youtube.com/watch?v=fFBSazTSGZw
Usando Github com Github Desktop em Projetos Eclipse: http://www.youtube.com/watch?v=EqHIjlYyS4U
Usando Github com SSH no Terminal Linux com chave gerada no Eclipse: http://www.youtube.com/watch?v=0s699q5Sja4
Usando Github com SSH no Terminal Windows com chave gerada no Eclipse: http://www.youtube.com/watch?v=DaydwPB2WSI

Vídeo Suporte:
Gerar Bibliotecas Java: https://youtu.be/9x3_c_0i6O0
Pilha Dinâmica: https://youtu.be/ahi8U7OnKco

Gerando link compartilháveis: https://drive.google.com/file/d/1cyoxa5W67MY5xDM6gCYpklel1GU3QCKa

Para todos os exercícios, quando solicitado teste de mesa, carregar a solução para um drive compartilhado e quando solicitado desenvolvimento, definir o que se pede e aplicar o código em Java e carregar a solução no Github.

1. Fazer um teste de mesa do algoritmo abaixo, deixando explícito os valores de saída em console e a estrutura da pilha

```
Pilha p = new Pilha();
int[] vetor = {100, 200, 1, 50, 39, 44, 25, 16, 99, 45, 33, 18, 102, 92};
int tamanhoVetor = vetor.length;
Para (i = 0 ; i < tamanhoVetor ; i++) {
      Se (pilhaVazia() == verdadeiro) {
            p.push(vetor[i] - 10);
      } Senao Se (vetor[i] mod 5 == 0) {
            p.push(vetor[i] / 5);
      } Senao Se (vetor[i] mod 3 == 0) {
            p.push(vetor[i] * 3);
      } Senao {
            int v1 = p.pop();
            escreva (v1 / 2);
      }
}
```

2. Criar um teste de mesa que demonstre a utilização de uma pilha de inteiros para realizar a operação de cálculo de fatorial de um dado número.

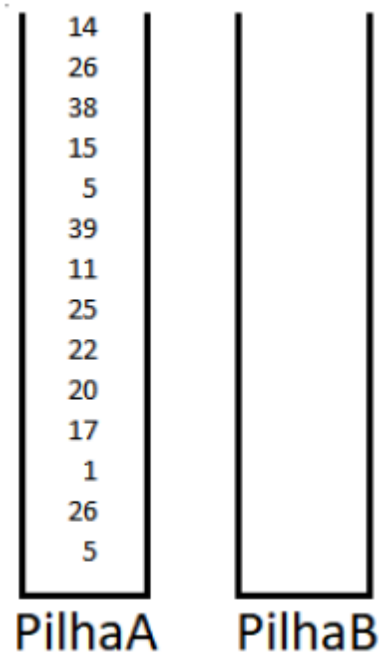
Para o teste, deve se usar um número de 0 a 10.

Utilizando as operações possíveis em uma pilha, apresentar o teste de mesa que receba um número de entrada e, utilizando uma pilha vazia e operações, chegar no valor do fatorial do número de entrada.

3. Baseado na lógica do Exercício 2, criar uma aplicação Java, baseada na biblioteca de pilha de int, que faça:
   1) Criar uma classe view (Principal.java) que, na main:
      a. Peça ao usuário um valor inteiro de 0 a 10 (Caso o usuário digite um valor fora desses limites, pedir novamente, até que o valor atenda à solicitação);
   2) Criar uma classe controller chamada FatController, que tenha um método fatorial(int valor): int e faça:
      a. Inicialize uma pilha
      b. Usando a pilha como suporte, calcule o fatorial do valor de entrada
      c. O método deve retornar esse valor
   3) O método main da classe Principal.java deve chamar o método fatorial(int valor): long e exibir em console o valor do fatorial.

O método fatorial(int valor): long deve estar baseado nas operações da pilha de inteiros (push(), pop(), size(), top(), isEmpty())

4. Considere as estruturas de pilha abaixo e, demostre a sequência de operações (que pode ser representado por uma estrutura) que resolva o seguinte problema: Na pilha PilhaA só são permitidos números pares e na pilha PilhaB só são permitidos números ímpares. Qual sequência de passos para resolver o problema? Demonstrar com teste de mesa.



5. Considere a estrutura de pilha abaixo e, apresente uma solução para, fazendo no máximo 2 pops consecutivos, apresentar o maior valor da pilha no console. Demonstrar com teste de mesa.