

Exercício Pilhas.

1) – Implemente uma Classe PilhaInteger que utilize uma lista do tipo Integer, com os métodos empilhar(Integer), Integer desempilhar(), boolean pilhaVazia(), Integer exibirTopo().

Class PilhaDouble	
Atributo	
(-) List<Integer> pilha	
Métodos	
(+) empilhar(Integer):void	
(+) desempilhar(): Integer	
(+) pilhaVazia(): boolean	
(+) exibirTopo(): Integer	

2) Crie um método que recebe uma String e retorne True se a String é Palíndromo. Ou seja, o nome é lido da mesma forma nos dois sentidos. Para isso utilize uma pilha para inverter o nome, exemplo: “maria” => “airam”. Depois compare o nome invertido com o original, se forem iguais é palíndromo.

- Utilize os métodos do objeto String :
 - .charAt(index); para acessar os caracteres da String.
 - .length(); para verificar a quantidade de caracteres da String.

3) Crie um método chamando validaBalanceamento que recebe uma String e verifica se uma sequência de caracteres contém parênteses balanceados. Retornando True ou False.

Por exemplo:

“{ [()] }” → Balanceado

“{ [()] }” → Desbalanceado

Utilize a seguinte lógica: Quando encontrar um símbolo de abertura {, [, (, adiciona o símbolo de fechamento equivalente a pilha. Por exemplo: encontra “{” → Empilha “}”. Quando encontrar um símbolo de fechamento },],), compara com o topo da pilha.

- Se for verdadeiro remove o símbolo da pilha.
- Se for falso retorna false.
- Se o método chegar ao final da String e a pilha estiver vazia, retorna True.

3) Considere uma pilha que armazene números inteiros, escreva uma função para ordenar os valores da pilha em ordem crescente, utilize pilhas auxiliares para realizar a ordenação. Por exemplo:

