

Lista 1 - Pilha

ESTRUTURA DE DADOS I – Pedro Nuno Moura

1) Usando a implementação de pilha apresentada em sala de aula, faça:

- a. Implemente uma função que transfere todos os elementos da pilha que chamou a função para a pilha vazia p, de tal forma que os elementos na pilha p fiquem na mesma ordem em que estavam na pilha que chamou a função. Assume-se que a pilha p tem tamanho maior ou igual ao da pilha que chamou a função. Implemente esta função de duas formas:
 - i. Utilizando uma pilha adicional. [void transferePilha(pilhaGenerica<T> p)]
 - ii. Não utilizando pilha adicional, sem usar vetor, e usando recursão. [void transferePilhaRecursivo(pilhaGenerica<T> p)]
- b. Implemente uma função revertePilha() que reverta a ordem dos elementos da pilha que chamou a função, utilizando duas pilhas adicionais.

2) Suponha que um programa execute uma sequência misturada de *pushes* e *pops*. Os *pushes* colocam os inteiros de 0 até 9 em ordem crescente na pilha; os *pops* imprimem os valores retornados. Qual (ais) da(s) sequências a seguir não poderia ocorrer?

- a. 4 3 2 1 0 9 8 7 6 5
- b. 4 6 8 7 5 3 2 9 0 1
- c. 2 5 6 7 4 8 9 3 1 0
- d. 4 3 2 1 0 5 6 7 8 9
- e. 1 2 3 4 5 6 9 8 7 0
- f. 0 4 6 5 3 8 1 7 2 9
- g. 1 4 7 9 8 6 5 3 0 2
- h. 2 1 4 3 6 5 8 7 9 0

3) Usando pilhas, crie uma classe Parênteses que receba um texto e avalie se os parênteses estão balanceados. Por exemplo, sua classe deve retornar true para “[()] { } { [() ()] () }” e false para “[()]”.

4) Elabore um método que retorne as letras invertidas das palavras que sofreram “push” armazenada numa estrutura de pilha genérica, preservando a ordem das palavras na frase. Por exemplo inserir “a maçã está podre” na pilha, deve ter como saída: “a ãçam átse erdop”. As operações básicas de uma pilha, push e pop, devem ser usadas.

5) Faça um programa que possua uma pilha e essa pilha tenha tamanho dinamicamente alocado de modo a evitar desperdício de memória. Essa pilha deve ser do tipo numérica e armazenar de modo crescente sempre. Faça as operações necessárias para garantir que a pilha sempre esteja nessa ordem.

6) Utilizando um vetor de 2 dimensões e com a classe pilhaGenerica<T>, implemente uma pilha do tipo genérica que armazena outras pilhas de String da seguinte forma:

```
public class pilhaGenerica<T>{  
  
    private Vector<T> pilhaPrincipal;  
    private Vector<String> pilhaSecundaria;  
    .  
    .  
    .  
  
    public void armazenaTexto(String texto){}  
  
    public String retornaTextoOriginal(){}  
  
    public String retornaTexto(){}  
  
}
```

O método armazenaTexto() recebe como argumento uma String contendo uma frase com tudo o que será armazenado. Exemplo: "Foi apenas um espectro de homem. Apenas passou pela vida, não viveu." Cada palavra será armazenada em uma posição da pilha principal. Cada posição da pilha principal será uma pilhaSecundária criada dinamicamente onde cada letra será armazenada em uma posição dessa pilha. Os espaços entre cada palavra marcam o começo e o fim de uma palavra. Crie a lógica para que o método retornaTextoOriginal() retorne o texto que foi originalmente inserido e outra para o método retornaTexto() que retorne o texto resultante dos consecutivos pops executados.