

## Prática de Laboratório 6

### Objetivos:

- Tipos Abstratos de Dados - TADs.
- Implementação da TAD Pilha e Fila.

### Implementações

#### Nodo

```
public class Nodo {  
  
    private String element;  
    private Nodo next;  
  
    public Nodo(String s, Nodo n) {  
        element = s;  
        next = n;  
    }  
  
    public String getElement() {  
        return element;  
    }  
  
    public Nodo getNext() {  
        return next;  
    }  
  
    public void setElement(String newElem) {  
        element = newElem;  
    }  
  
    public void setNext(Nodo newNext) {  
        next = newNext;  
    }  
}
```

## Pilha

```
public class Pilha {  
    protected Nodo topo; // topo da pilha, onde os elementos serão inseridos e retirado  
    protected int tamanho; // Número de nodos na Pilha  
  
    // Construtor  
    public Pilha() {  
        this.topo = null;  
        this.tamanho = 0;  
    }  
  
    // Insere elementos na pilha  
    void Empilha(String s) {  
        Nodo n = new Nodo(s, this.topo);  
        this.topo = n;  
        this.tamanho++;  
    }  
  
    // Remove elementos da pilha  
    public String Desempilha() {  
        Nodo aux = this.topo;  
        this.topo = this.topo.getNext();  
        this.tamanho--;  
        return aux.getElement();  
    }  
  
    // Verifica se a pilha está vazia  
    boolean vazia() {  
        return topo == null;  
    }  
  
    //Retorna o tamanho da pilha  
    public int tamanho() {  
        return this.tamanho;  
    }  
  
    //Retorna a informação no topo  
    public String topo() {  
        return this.topo.getElement();  
    }  
  
    //Imprime os elementos da pilha  
    public void imprimePilha() {  
        System.out.println("Pilha: ");  
        Nodo aux = this.topo;  
        while (aux != null) {  
            System.out.println(aux.getElement());  
            aux = aux.getNext();  
        }  
    }  
}
```

## Fila

```
public class Fila {
    protected Nodo inicio, fim; // topo da pilha, onde os elementos serão inseridos e retirados
    protected int tamanho; // Número de nodos na Fila

    // Construtor
    public Fila() {
        this.inicio = this.fim = null;
        this.tamanho = 0;
    }

    // Insere elementos na fila
    void Enfileira(String s) {
        Nodo n = new Nodo(s, null);
        if (vazia())
            this.fim = this.inicio = n;
        else {
            this.fim.setNext(n);
            this.fim = n;
            this.tamanho++;
        }
    }

    // Retira elementos da fila
    public String Desenfileira() {
        Nodo aux = this.inicio;
        this.inicio = this.inicio.getNext();
        this.tamanho--;
        return aux.getElement();
    }

    // Verifica se a fila está vazia
    boolean vazia() {
        return this.inicio == null;
    }

    //Retorna a quantidade de elementos da fila
    public int tamanho() {
        return this.tamanho;
    }

    //Retorna o primeiro da fila
    public String frente() {
        return this.inicio.getElement();
    }

    //Imprime a fila
    public void imprimeFila() {
        System.out.println("Fila: ");
        Nodo aux = this.inicio;
        while (aux != null) {
            System.out.println(aux.getElement());
            aux = aux.getNext();
        }
    }
}
```

### Atividades

1. Implemente as TADs Pilha e Fila.
2. Considere as TADs Pilha e Fila e suas respectivas funções Empilha e Desempilha, Enfileira e Desenfileira. Crie uma função que recebe uma fila e que retorne essa fila invertida utilizando somente as funções acima.
3. Utilizando somente operações de empilhar e desempilhar, escreva um programa que remove um item com chave *c* fornecida pelo usuário da pilha. Ao final da execução da função, a pilha deve ser igual à original, exceto pela ausência do item removido.
4. Escreva um programa solicite ao usuário uma sequência de caracteres sem limite de máximo de tamanho e realize as seguintes operações usando uma pilha:
  - a. Imprimir o texto na ordem inversa;
  - b. Verificar se o texto é um palíndromo, ou seja, se a *string* é escrita da mesma maneira de frente para trás e de trás para frente. Ignore espaços e pontos.

Bom trabalho! 🖐