

```

public class Pilha {
    private Node topo;
    private int tamanho;

    // Construtor para inicializar a pilha vazia
    public Pilha() {
        this.topo = null; // Inicialmente a pilha está vazia, então o topo é null
        this.tamanho = 0;
    }

    // Classe Node - representa um nó na pilha
    private static class Node {
        private String elemento;
        private Node proximo;

        // Construtor para criar um nó com um elemento especificado
        public Node(String elemento) {
            this.elemento = elemento;
            this.proximo = null;
        }
    }

    // Método para verificar se a pilha está vazia
    public boolean estaVazia() {
        return (topo == null);
    }

    // Método para empilhar um elemento
    public void empilhar(String elemento) {
        Node novoNode = new Node(elemento);
        novoNode.proximo = topo;
        topo = novoNode;
        tamanho++;
        System.out.println("Elemento \"" + elemento + "\" empilhado com sucesso.");
    }

    // Método para desempilhar um elemento
    public String desempilhar() {
        if (estaVazia()) {
            System.out.println("A pilha está vazia. Não é possível desempilhar elementos.");
            return null;
        }
        String elementoDesempilhado = topo.elemento;
        topo = topo.proximo;
    }
}

```

```

        tamanho--;
        System.out.println("Elemento \'' + elementoDesempilhado + '\'' desempilhado com
sucesso.");
        return elementoDesempilhado;
    }

    // Método para obter o elemento do topo da pilha sem removê-lo
    public String topo() {
        if (estaVazia()) {
            System.out.println("A pilha está vazia. Não há elemento no topo.");
            return null;
        }
        return topo.elemento;
    }

    // Método para exibir todos os elementos da pilha
    public void exibir() {
        if (estaVazia()) {
            System.out.println("A pilha está vazia.");
            return;
        }
        System.out.println("Elementos na pilha:");
        Node atual = topo;
        while (atual != null) {
            System.out.println(atual.elemento);
            atual = atual.proximo;
        }
    }

    public static void main(String[] args) {
        Pilha pilha = new Pilha();

        // Empilhando elementos
        pilha.empilhar("Elemento 1");
        pilha.empilhar("Elemento 2");
        pilha.empilhar("Elemento 3");

        // Exibindo o topo da pilha
        System.out.println("Topo da pilha: " + pilha.topo());

        // Desempilhando um elemento
        pilha.desempilhar();

        // Exibindo todos os elementos da pilha

```

```
        pilha.exibir();  
    }  
}
```