

TODO O CODIGO FONTE DO PROJETO

CLASSE STACK:

```
package AA1;

public class Stack<A> {

    /* Container para os elementos da pilha */
    class Element {
        private A val; // valor do elemento
        private Element next; // proximo elemento da pilha

        //cria um elemento com o valor el e uma referencia para next_
        public Element(A el, Element next_){
            this.val = el;
            this.next = next_;
        }
        public A getVal() {
            return val;
        }

        public Element getNext() {
            return next;
        }
    }

    private Element top; // Ultimo elemento da pilha

    // Contrutor da pilha
    // Cria uma pilha vazia
    public Stack() {
        this.top = null;
    }

    // Cria um novo elemento com o valor 'el', cujo proximo elemento
    // é o this.top (ultimo elemento da pilha) e o torna ultimo elemento da
    pilha(this.top)
    public void push(A el) {
        Element newtop = new Element(el, this.top);
        this.top = newtop;
    }

    // Remove o ultimo elemento e atualiza a referencia
    //this.top para ser o proximo elemento(this.top.getNext())
    public A pop() throws EmptyException {
        //Lança um erro caso a pilha esteja vazia
        if (this.top == null)
            throw new EmptyException("A LISTA ESTA VAZIA! ESTA OPERAÇÃO NAO PODE
SER EXECUTADA");

        A el = this.top.getVal();
        this.top = this.top.getNext();
        return el;
    }
}
```

```

// Percorre cada elemento da pilha imprimindo seu conteudo
public void print(String name) {
    System.out.print(name + ": [ ");
    Element act = this.top;
    while (act != null) {
        System.out.print(" " + act.getVal());
        act = act.getNext();
        if (act == null)
            break;
        System.out.print(", ");
    }
    System.out.println(" ]");
}

// Mostrando o item que esta no topo no momento
public A peek() throws EmptyException {
    if (this.top == null)
        throw new EmptyException("A LISTA ESTA VAZIA! ESTA OPERAÇÃO NAO PODE
SER EXECUTADA");

    return this.top.getVal();
}
}

```

CLASSE TESTE:

```

package AA1;

public class Teste {
    public static void execute() {
        Stack<String>stack = new Stack<String>();
        try {
            stack.push("João");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Maria");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Antonio");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Unibrasil");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Marcos");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Eugenio");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Leão");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Joaquim");
            System.out.println(stack.peek() + " adicionado");
            stack.push("José");
            System.out.println(stack.peek() + " adicionado");
            stack.push("Carlos Eduardo Albuquerque");
            System.out.println(stack.peek() + " adicionado");
            System.out.println();
            stack.print("Stack");
        }
    }
}

```

```

        System.out.println();
        System.out.println("1 pop: " + stack.pop());
        System.out.println("2 pop: " + stack.pop());
        System.out.println("3 pop: " + stack.pop());
        System.out.println("4 pop: " + stack.pop());
        System.out.println("5 pop: " + stack.pop());
        System.out.println("6 pop: " + stack.pop());
        System.out.println("7 pop: " + stack.pop());
        System.out.println("8 pop: " + stack.pop());
        System.out.println("9 pop: " + stack.pop());
        System.out.println("10 pop: " + stack.pop());
        stack.print("Pilha Vazia");
        System.out.println("11 pop: " + stack.pop());
        System.out.println();
    }
    catch(Exception e ) {
        System.err.println(e);
    }
}
}

```

CLASSE EmptyException:

```

package AA1;

/* Exception para pilha vazia */
public class EmptyException extends Exception{

    public EmptyException (String errorMsg) {
        super(errorMsg);
    }
}

```

CLASSE MAIN:

```

package AA1;

class Main{
    public static void main(String[] args) {

        Teste.execute();
    }
}

```

FIGURAS DE EVIDÊNCIA:

```
João adicionado
Maria adicionado
Antonio adicionado
Unibrasil adicionado
Marcos adicionado
Eugenio adicionado
Leão adicionado
Joaquim adicionado
José adicionado
Carlos Eduardo Albuquerque adicionado

Stack: [ Carlos Eduardo Albuquerque, José, Joaquim, Leão, Eugenio, Marcos, Unibrasil, Antonio, Maria, João ]

1 pop: Carlos Eduardo Albuquerque
2 pop: José
3 pop: Joaquim
4 pop: Leão
5 pop: Eugenio
6 pop: Marcos
7 pop: Unibrasil
8 pop: Antonio
9 pop: Maria
10 pop: João
Pilha Vazia: [ ]
AA1.EmptyException: A LISTA ESTA VAZIA! ESTA OPERAÇÃO NAO PODE SER EXECUTADA
```