Filipe de Oliveira Ataíde – Mat.: 20181014040022

Pilha Rubro Negra (Obs: trabalho apresentado posteriormente à paralisação)

Main.java

```
import java.util.Scanner;

class Main {
	public static void main(String[] args) throws Exception{
		Scanner sc = new Scanner(System.in);
		System.out.print("Quant. Array: ");
		int t = sc.nextInt();
		int opt = 1;
		Array S = new Array(t);
		Object o;
		while(opt != 0){
			try{
				S.items_rubra();
				S.items_negra();
				System.out.println("[1] Push (Rubra)");
				System.out.println("[2] Pop (Rubra)");
				System.out.println("[3] Size (Rubra)");
				System.out.println("[4] Push (Negra)");
				System.out.println("[5] Pop (Negra)");
				System.out.println("[6] Size (Negra)");
				opt = sc.nextInt();
				switch(opt){
					case(1):
						System.out.println("Push: ");
						o = sc.next();
						S.push_rubra(o);
						break;
					case(2):
						S.pop_rubra();
						break;
```

```
                                        case(3):
                                                System.out.println("Size: " + S.size_rubra());
                                                break;
                                        case(4):
                                                System.out.println("Push: ");
                                                o = sc.next();
                                                S.push_negra(o);
                                                break;
                                        case(5):
                                                S.pop_negra();
                                                break;
                                        case(6):
                                                System.out.println("Size: " + S.size_negra());
                                                break;
                                        }
                                }
                                catch(EmptyStackException e){
                                        System.out.println("Pilha vazia.");
                                }
                        }
                }
        }
```

Array.java

```
public class Array implements Tad{
	private Object S[];
	private int t1 = -1;
	private int t2;

	public Array(int t){
		S = new Object[t];
		t2 = t;
	}

	public int size_rubra(){
		return t1+1;
	}
	public int size_negra(){
		return S.length - t2;
	}

	public void duplicate(){
		Object[] N = new Object[S.length*2];
		for(int i = 0; i <= t1; i++){
			N[i] = S[i];
		}
		for(int i = S.length-1, j = N.length-1; i >= t2; i--, j--){
			N[j] = S[i];
		}
		t2 = t2 + (N.length - S.length);
		S = N;
	}

	public void push_rubra(Object o){
		if(t1 == t2-1){
			duplicate();
		}
```

```
        t1 = t1 + 1;
        S[t1] = o;
}

public void push_negra(Object o){
        if(t2 == t1+1){
                duplicate();
        }
        t2 = t2 - 1;
        S[t2] = o;
}

public Object pop_rubra() throws EmptyStackException{
        if(t1 == -1){
                throw new EmptyStackException("Pilha vazia.");
        }
        Object temp = S[t1];
        S[t1] = null;
        t1 = t1 - 1;
        return temp;
}

public Object pop_negra() throws EmptyStackException{
        if(t2 == S.length){
                throw new EmptyStackException("Pilha vazia.");
        }
        Object temp = S[t2];
        S[t2] = null;
        t2 = t2 + 1;
        return temp;
}

public void items_rubra(){
        if(t1 == -1){
```

```
                return;
            }
            System.out.print("Rubra: |");
            for(int i = 0; i <= t1; i++){
                System.out.print(" " + S[i] + " |");
            }
            System.out.print("\n");
        }

        public void items_negra(){
            if(t2 == S.length){
                return;
            }
            System.out.print("Negra: |");
            for(int i = S.length-1; i >= t2; i--){
                System.out.print(" " + S[i] + " |");
            }
            System.out.print("\n");
        }
}
```

Tad.java

```
public interface Tad {
	public int size_rubra();
	public int size_negra();
	public void push_rubra(Object o);
	public void push_negra(Object o);
	public Object pop_rubra() throws Exception;
	public Object pop_negra() throws Exception;
	public void items_rubra();
	public void items_negra();
}
```

EmptyStackException.java

```
public class EmptyStackException extends Exception{
	public EmptyStackException(String err){
		super(err);
	}
}
```