

Você é capaz de implementar os métodos solicitados abaixo?

**Método 1:** Na classe que representa uma lista estática (classe criada em aula), implemente um método **RECURSIVO** que recebe um elemento por parâmetro e retorna a quantidade de vezes que este elemento aparece na lista. Você deve fazer os métodos público e privado. Você deve utilizar a seguinte assinatura para o método público (utilizando exceções na declaração, caso seja necessário):

```
public int contaElementos(E el)
```

**Método 2:** Implemente em uma classe chamada **Metodo2** o seguinte método:

```
public boolean checkBrackets(Stack<Character> s1)
```

Esse método verifica se uma expressão matemática tem os parênteses agrupados de forma correta, isto é:

**(1)** se o número de parênteses à esquerda e à direita é igual; e

**(2)** se todo parêntese aberto é seguido, posteriormente, por um fechamento de parêntese.

Por exemplo, as expressões " $((A+B))$ " e " $A+B($ " violam a regra (1) e as expressões " $)A+B(-C$ " e " $(A+B)) - (C+D$ " violam a regra (2). Um exemplo de expressão correta seria: " $((A+B) - (C+D))$ ".

O método recebe como parâmetro uma **pilha que contém os caracteres de uma expressão matemática**. O método retorna verdadeiro se os parênteses da expressão estão agrupados de forma correta, ou falso, caso contrário. **Uma pilha armazena apenas uma única expressão**. As expressões **são armazenadas na pilha da direita para a esquerda**, ou seja, os caracteres da direita (ou seja, do final da expressão) são empilhados primeiro.

A imagem abaixo ilustra duas possíveis pilhas de entrada. A pilha da esquerda armazena a expressão " $A+B($ ", enquanto a pilha da direita armazena a expressão " $(A+B)$ ".