

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Campus Guarapuava
Tecnologia em Sistemas para Internet

Resolução de Problemas
Prof. Dr. Eleandro Maschio

Estruturas de Dados Lineares

Exercício 1

BALANCEAMENTO – Modele uma classe que tenha, como atributo, um texto que represente uma expressão matemática (recebido no construtor). Implemente um método que retorne se essa expressão está balanceada quanto às chaves, colchetes e parênteses.

Considere que os colchetes são internos às chaves e que os parênteses são internos aos colchetes. Assim, não se pode ter chaves dentro de parênteses.

Nenhuma outra propriedade precisa ser avaliada (como, por exemplo, os operadores e operandos).

Exemplos de expressões válidas:

```
(a + b)
{a * [c - b * (e + f)]} - 2
```

Exemplos de expressões inválidas:

```
(a + b
a + b)
{a * [(c - b * (e + f)]} - 2
```

Exercício 2

LISTA ALEATÓRIA – Modele uma classe cujo construtor gere uma lista de n elementos inteiros aleatórios. Sobrescreva o método `toString()` para que retorne a representação dessa lista em texto.

Exercício 3

VALOR DE CORTE – Na mesma classe do exercício anterior, implemente um método que receba, como parâmetro, um número inteiro e remova da lista todos os números menores do que esse valor.

Exercício 4

REMOVE MÚLTIPLOS – Na mesma classe do exercício anterior, implemente um método que receba um número inteiro e remova da lista todos os elementos que estiverem em posições múltiplas desse valor. Por exemplo, caso o número fornecido seja 3, serão removidos os elementos das posições 3, 6, 9 e, assim, sucessivamente (note que os índices são 2, 5, 8). Se o número fornecido for zero, a lista permanece inalterada.

Exercício 5

MAIS PRÓXIMO DA MÉDIA – Na mesma classe do exercício anterior, implemente um método que retorne o número mais próximo da média dos elementos. Caso haja números igualmente próximos da média, retorne o primeiro encontrado.

Exercício 6

REDUZ – Na mesma classe do exercício anterior, implemente um método que receba um inteiro n e que reduza a lista a essa quantidade de elementos.

Exercício 7

FATIA – Na mesma classe do exercício anterior, implemente um método que receba dois inteiros, *inicio* e *fim*. Altere a lista de forma que ela se resuma a uma fatia que vá de *inicio* a *fim*. Não use

métodos predefinidos como `slice()`.

Exercício 8

INVERTE – Na mesma classe do exercício anterior, implemente um método que inverta a lista. Com isso, o primeiro elemento se tornará o último, o segundo será o penúltimo e, assim, sucessivamente. Não use métodos predefinidos como `reverse()`.

Exercício 9

AMPLITUDE – Na mesma classe do exercício anterior, implemente um método que retorne a diferença entre o maior e o menor elemento da lista.

Exercício 10

JOSEPHUS – Modele e implemente uma variação do Problema de Josephus. Um grupo de pessoas são dispostas em círculo e cada uma escolhe um inteiro positivo. Sorteia-se uma pessoa e toma-se o número que ela escolheu para iniciar uma contagem, a partir da pessoa seguinte, em sentido horário. Aquele que estiver na posição correspondente ao número escolhido é eliminado do jogo, então o número que essa pessoa escolheu é usado para continuar a contagem. A cada pessoa eliminada, usa-se o número por ela escolhido para determinar a próxima pessoa eliminada. Retorne a última pessoa restante nesse jogo.

Como Citar

Todos os exercícios desta lista são autorais.

MASCHIO, Eleandro. **Estruturas de Dados Lineares**. Guarapuava: Universidade Tecnológica Federal do Paraná, 2024. 2 p. Material didático da disciplina de Resolução de Problemas.