



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL
TÉCNICAS DE PROGRAMAÇÃO – PROF. FISCHER

Lista 2 – Implementação de métodos

Faça uma classe Utilidades que tenha os metodos descritos em cada item abaixo. Como exemplo, utilize a Classe “ControleRemoto” feita em sala de aula.

- 1) **Somatório:** Seu método deverá receber um vetor como parâmetro e retornar a soma de todos elementos do vetor, sendo que as primeiras duas e as últimas posições do vetor de entrada não devem ser consideradas;
- 2) **Números pares:** Seu método deverá imprimir todos os números pares dentre um número inicial e final informado como parâmetro do método.
- 3) **Média:** Seu método deverá receber dois vetores como entrada e retornar a média dos valores presentes nos vetores.
- 4) **Média ponderada:** Seu método deverá receber um vetor de notas e um vetor de pesos correspondentes. Seu método deve retornar a média ponderada das notas. Exemplo: [7, 8, 6], [2, 3, 1] -> 7.33
- 5) **Contagem de elementos:** Seu método deverá receber dois vetores e um elemento a ser contado. A função deve retornar a quantidade de vezes que esse elemento aparece nos vetores. Exemplo: [1,3, 3, 2, 5],[1, 2, 3, 2, 4], 2 -> 3
- 6) **Cópia de vetor:** Seu método deverá receber um vetor e retornar uma cópia exata desse vetor.
- 7) **Ordenação crescente:** Seu método deverá receber um vetor e retornar o vetor ordenado em ordem crescente. Exemplo: [5, 2, 9, 1, 7] -> [1, 2, 5, 7, 9]
- 8) **Verificação de ordenação:** Seu método deverá receber um vetor e verificar se ele está ordenado em ordem crescente. Se sim retornar true.
- 9) **K-ésimo maior valor:** Seu método deverá receber um vetor e um K e retornar o K-ésimo maior valor presente no vetor.
- 10) **K-ésimo menor valor:** Seu método deverá receber um vetor e um K e retornar o K-ésimo menor valor presente no vetor.

- 11) **Verificação de igualdade:** Seu método deverá receber dois vetores e verificar se eles são iguais, ou seja, se possuem os mesmos elementos na mesma ordem.
- 12) **Potenciação:** Seu método deverá receber dois números, a base e o expoente, e retornar a potência da base elevada ao expoente.
- 13) **Primo:** Seu método deverá receber um número inteiro e verificar se ele é primo, ou seja, se é divisível apenas por 1 e por ele mesmo.
- 14) **Busca por ocorrências:** Seu método deverá receber dois vetores e deverá retornar um novo vetor com todos os elementos que aparecem em ambos os vetores.
- 15) **Remoção de elemento do vetor:** Seu método deverá receber um vetor e um elemento a ser removido. A função deve remover todas as ocorrências desse elemento no vetor. Um novo vetor, apenas com elementos mantidos, deve ser retornado: Exemplo: [1, 2, 3, 2, 4], 2 -> [1, 3, 4]
- 16) **Verificação de palíndromo:** Seu método deverá receber uma string e verificar se ela é um palíndromo, ou seja, se pode ser lida da mesma forma de trás para frente. Exemplo: "ana" -> True.
- 17) **Verificação de número primo:** Seu método deverá receber um número inteiro e verificar se ele é um número primo. Exemplo: 7 -> True.
- 18) **Inversão de string:** Seu método deverá receber uma string e retornar a mesma string invertida. Exemplo: "hello" -> "olleh".
- 19) **Cálculo de juros compostos:** Seu método deverá receber um valor inicial, uma taxa de juros e um número de períodos. A função deve retornar o valor final após a aplicação dos juros compostos. Exemplo: 1000, 0.05, 3 -> 1157.63
- 20) **Verificação de número perfeito:** Seu método deverá receber um número inteiro e verificar se ele é um número perfeito, ou seja, se a soma dos seus divisores próprios é igual a ele mesmo. Exemplo: 6 -> True
- 21) **Diminuição do vetor:** Seu método deverá receber um número qualquer que represente e criar um vetor com o tamanho informado. Este vetor deverá ter sucessivas diminuições do seu tamanho em um unidade até que seu tamanho seja igual a 1.
- 22) **Imprimir elementos:** Faça um método para cada item abaixo e imprima as formas apresentadas. Permita que o usuário escolha o número de linhas e colunas desejadas. No exemplo abaixo, para cada matriz foram utilizadas 5 linhas e 5 colunas. Porém, independentemente do número de linhas e colunas, as formas devem ser mantidas. As matrizes sempre serão quadradas.

a)

| | | | | |
|---|---|---|---|---|
| 0 | | | | |
| 0 | 1 | | | |
| 0 | 1 | 2 | | |
| 0 | 1 | 2 | 3 | |
| 0 | 1 | 2 | 3 | 4 |

b)

| | | | | |
|---|---|---|---|---|
| @ | * | @ | * | @ |
| * | @ | * | @ | * |
| @ | * | @ | * | @ |
| * | @ | * | @ | * |
| @ | * | @ | * | @ |

b)

| | | | | |
|---|---|---|---|---|
| 4 | 3 | @ | 1 | 0 |
| 0 | 1 | @ | 3 | 4 |
| @ | @ | @ | @ | @ |
| 0 | 1 | @ | 3 | 4 |
| 4 | 3 | @ | 1 | 0 |

c)

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | * | * | * | 3 |
| 2 | * | * | * | 2 |
| 3 | * | * | * | 1 |
| 4 | 3 | 2 | 1 | 0 |

d)

| | | | | |
|---|---|--|---|---|
| # | # | | ! | ! |
| # | # | | ! | ! |
| = | = | | = | = |
| * | * | | % | % |
| * | * | | % | % |

e)

| | | | | |
|---|---|---|---|---|
| 0 | * | * | * | 4 |
| * | 1 | * | 3 | * |
| * | * | 2 | * | * |
| * | 1 | * | 3 | * |
| 0 | * | * | * | 4 |