

LÓGICA DE PROGRAMAÇÃO – LISTA DE EXERCÍCIOS 10

1. Escreva um método chamado *fatorial* que receba um valor do tipo *int* como parâmetro e retorne como resposta um valor do tipo *long*. Defina este método em uma classe chamada *Calculos*.

2. Escreva um programa em Java que calcule e escreva a soma dos termos da série:

$$S = \frac{100}{0!} + \frac{99}{1!} + \frac{98}{2!} + \frac{97}{3!} + \dots + \frac{80}{20!}$$

Use o método definido no exercício 1 para o cálculo do fatorial.

3. Faça um programa em Java que calcule e mostre o resultado da série:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots,$$

O valor de *x* deve ser informado pelo usuário. A série deve ser calculada até uma precisão de 0,001. Utilize o método definido no exercício 1 para o cálculo do fatorial.

4. Escreva um método que receba dois valores do tipo *int* e calcule o MMC deles. O método deve retornar uma resposta do tipo *int*. Coloque este método na classe *Calculos*.

5. Escreva um programa em Java que calcule a soma de duas frações. Para isso, o programa deve pedir ao usuário que informe o valor do numerador e do denominador de cada fração (todos são do tipo *int*). Considerando que *n1* é o numerador da primeira fração, *d1* é o denominador da primeira fração, *n2* é o numerador da segunda fração e *d2* é o denominador da segunda fração, o resultado da soma será:

$$dr = \text{MMC de } d1 \text{ e } d2$$

$$nr = \frac{dr}{d1} \times n1 + \frac{dr}{d2} \times n2$$

onde *dr* é o denominador do resultado e *nr* é o numerador do resultado. Utilize o método escrito no exercício 4 para calcular o MMC.

6. Escreva um programa em Java que leia uma String contendo uma expressão e calcule o resultado dela. A expressão escrita será formada sempre por dois operandos e um operador, na forma:

$$\text{operando1 operador operando2}$$

em que *operando1* e *operando2* podem ser números inteiros quaisquer e operador pode ser +, -, * ou /, indicando a operação simbolizada por cada sinal. Pode haver um número de espaços em branco qualquer no meio da expressão.

7. Escreva um programa em Java que repita o que foi feito no exercício 6, até que o usuário digite “sair”.
8. Acrescente na classe *Calculos* um método que verifique se um número é primo ou não. Este método deve receber um valor do tipo *int* como parâmetro e deve devolver uma resposta do tipo *boolean*.
9. Use o método definido no exercício 9 em um programa Java que leia um número inteiro do usuário e mostre todos os divisores daquele número que são primos.
10. Baseado no exercício 9 da lista 7, escreva um método que receba uma data como *String* como parâmetro e verifique se ela é válida ou não. O método deve retornar uma resposta do tipo *boolean* (*true* se a data é válida e *false* caso contrário). Coloque este método na classe *Data* criada na aula do dia 22/06.
11. Escreva um método que receba uma data como *String* e devolva o dia desta data como *int*. Este método deve ser definido na classe *Data*, usada no exercício anterior. Caso a data informada seja inválida, o método deve retornar -1.
12. Escreva um método que receba uma data como *String* e devolva o mês desta data como *int*. Este método deve ser definido na classe *Data*, usada no exercício anterior. Caso a data informada seja inválida, o método deve retornar -1.
13. Escreva um método que receba uma data como *String* e devolva o ano desta data como *int*. Este método deve ser definido na classe *Data*, usada no exercício anterior. Caso a data informada seja inválida, o método deve retornar -1.
14. Utilizando os métodos da classe *Data* já definidos, escreva um programa em Java que permita somar um determinado número de dias, meses ou anos a uma data. Primeiramente, o programa deve solicitar ao usuário que informe uma data (como um valor *String*, no formato especificado no exercício 9 da lista 7). Caso a data seja inválida, o programa deve mostrar uma mensagem (utilize o método feito no exercício 10 para verificar se a data é válida ou inválida). Caso a data seja válida, o programa deve pedir ao usuário que informe uma quantidade a ser somada a data. O programa deve ler um *String*, que deverá ter um número inteiro seguido da palavra *dias*, *meses* ou *anos*. Por exemplo, se o usuário digitar “5 dias” devem ser somados 5 dias na data (cuidado, pois o número de dias a ser somado pode levar a uma data para o próximo mês – seu programa deve saber lidar com isso). Caso o usuário digite “6 meses” devem ser somados 6 meses na data (cuidado, pois o número de meses pode levar a uma data para o próximo ano – seu programa deve saber lidar com isso). Caso o usuário digite “3 anos”, devem ser somados 3 anos na data. A data resultante deve ser mostrada na tela, utilizando o caractere “/” como separador.