

Curso Básico Java – Lista de Exercícios 1 – Partes 1 a 4

1. Programação básica (estruturada)

1) Faça um programa para calcular o valor das seguintes expressões:

- $S_1 = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \dots + \frac{99}{50}$
- $S_2 = \frac{2^1}{50} + \frac{2^2}{49} + \frac{2^3}{48} + \dots + \frac{2^{50}}{1}$
- $S_3 = \frac{1}{1} - \frac{2}{4} + \frac{3}{9} - \frac{4}{16} + \frac{5}{25} - \dots - \frac{10}{100}$

Resultado Esperado

S1 = 95.5007946616706
S2 = 1.5608286920413398E15
S3 = 0.6456349206349207

2) Observe a seguinte propriedade que alguns números maiores que 1000 e menores que 9999 possuem:

Número: abcd
(ab) + (cd) = (ef)
(ef)² = abcd
Exemplo: 3025
30+25 = 55
55² = 3025

Resultado Esperado

2025
3025
9801

Faça um programa que imprima todos os números que satisfaçam esta propriedade.

3) Faça um programa para calcular o Máximo Divisor Comum entre 2 números passados como argumentos para a execução do programa. Observe as seguintes propriedades do MDC:

- MDC(x, y) = MDC(x – y, y), se x > y;
- MDC(x, y) = MDC(y, x);
- MDC(x, x) = x.

Dica

Para converter uma string "10"
para o número 10 use:
`Integer.parseInt("10");`

Exemplo:

MDC(3,5)=MDC(5,3)=MDC(2,3)=MDC(3,2)=MDC(1,2)=MDC(2,1)=MDC(1,1)=1

2. Vetores e Entrada de Dados Básica

4) Considere um vetor de 10 números inteiros positivos maiores que zero e um único número X inteiro, também positivo e maior que zero. Faça um programa para:

- ler pelo teclado o vetor;
- ler pelo teclado o número X;
- encontrar e imprimir o par de posições consecutivas cujas componentes possuam a maior distância entre elas;
- verificar se o vetor está em ordem crescente, decrescente ou não ordenado;
- dizer quantos números no vetor são maiores que X, menores que X e iguais a X.

5) Verifique se as componentes de um vetor de 10 componentes inteiros lidos pelo teclado formam uma progressão aritmética, informando se sim ou se não. Caso forme, imprima o termo inicial e a razão.

6) Faça um programa que leia duas matrizes de reais A e B, com respectivos tamanhos 5 x 4 e 4 x 6, e imprima a matriz resultante A x B (multiplicação das matrizes), de tamanho 5 x 6.

3. Programação Orientada a Objetos (sem herança)

7) Crie uma classe que representa um ponto no plano cartesiano. Em seguida, crie uma classe que representa um triângulo, reusando a classe anterior por composição. Finalmente, escreva um programa que receba do usuário as coordenadas dos vértices do triângulo e imprima seu perímetro.

8) Implemente uma classe chamada `Aleatorio` que representa um número aleatório (como aquela utilizada nos slides do curso). Esta classe deve possuir:

- Um objeto da classe `java.util.Random` compartilhado por todas as instâncias da classe. Ao criar este objeto, passe o valor `System.currentTimeMillis()` como parâmetro ao seu construtor;
- Uma constante `VALOR_MAXIMO_DEFAULT` (a ser utilizada somente pela própria classe) que define o valor máximo padrão caso um não seja especificado no construtor de `Aleatorio`;
- Dois construtores: um sem parâmetros e outro que recebe um valor inteiro como parâmetro, que deve ser utilizado como valor máximo para o número aleatório gerado;
- Um atributo que armazene o número aleatório gerado durante a construção do objeto e um método que retorne este valor.

Crie uma outra classe com um método `main()` que gere e imprima 10 números aleatórios utilizando a classe `Aleatorio`.

9) No exercício anterior, uma vez que um número aleatório é gerado e armazenado em uma instância da classe `Aleatorio`, ele não é mais modificado. Agora, gostaríamos de adicionar um método `renovar()` a esta classe que troca o valor numérico armazenado dentro dela, porém obedecendo o valor máximo especificado (ou não) no momento da construção daquele objeto. Modifique a classe nesse sentido.

10) Crie uma classe que representa um funcionário, registrando seu nome, salário e data de admissão. Em seguida, crie uma classe que represente um departamento de uma empresa, registrando o nome e os funcionários que nele trabalham (para uso de vetores, considere um máximo de 100 funcionários). Por fim, crie uma classe que representa uma empresa, registrando seu nome, CNPJ e departamentos (considere um máximo de 10 departamentos). Faça um programa que:

- Crie uma empresa;
- Adicione a esta empresa alguns departamentos;
- Adicione aos departamentos alguns funcionários;
- Dê aumento de 10% a todos os funcionários de um determinado departamento;
- Transfira um funcionário de um departamento para outro.

É esperado que seu código seja bem encapsulado. Por exemplo, para adicionar um departamento em uma empresa (ou um funcionário a um departamento), não se deve acessar o vetor (ou lista) de departamentos diretamente, mas sim ter um método na classe que representa a empresa para adicionar um departamento.