

# LPOO – Lista de Exercícios

## Conceitos Básicos de Programação em Java

1. Escreva o famoso *Hello World* de C, que imprime a frase "Hello World" na tela. Logo após, escreva um programa em Java com a mesma funcionalidade. Compare as diferenças de código, compilação e execução.
2. Altere o nome do arquivo criado anteriormente para *OutroNome.java*. Logo após, tente compilá-lo e executá-lo. Qual o padrão básico que você detectou em relação aos nomes de arquivos Java?
3. Os tipos primitivos do Java são: *int*, *double*, *float*, *boolean*, *char*, *byte*, *short* e *long*. Crie um programa Java que declara algumas dessas variáveis, inicializa com um valor fixo e as imprime logo após.
4. Por haver vários tipos primitivos, em alguns momentos podemos confundir e realizar atribuições que não são permitidas pelo Java. Crie um programa Java que declara variáveis de diferentes tipos e realiza atribuições entre elas. Analise as tabelas de *casting* e de tipos abaixo e entenda os motivos pelos quais suas atribuições funcionaram ou não.

PARA:	byte	short	char	int	long	float	double
DE:							
byte	----	<i>Impl.</i>	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
short	(byte)	----	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
char	(byte)	(short)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
int	(byte)	(short)	(char)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
long	(byte)	(short)	(char)	(int)	----	<i>Impl.</i>	<i>Impl.</i>
float	(byte)	(short)	(char)	(int)	(long)	----	<i>Impl.</i>
double	(byte)	(short)	(char)	(int)	(long)	(float)	----

TIPO	TAMANHO
boolean	1 bit
byte	1 byte
short	2 bytes
char	2 bytes
int	4 bytes
float	4 bytes
long	8 bytes
double	8 bytes

5. Crie um programa Java que imprime todos os números de 150 a 300 utilizando as estruturas de repetição *for*, *while* e *do..while*;
6. Crie um programa Java que imprime a soma de 1 até 1000 utilizando as estruturas de repetição *for*, *while* e *do..while*;
7. Crie um programa Java que imprime todos os múltiplos de 3 entre 1 e 100, utilizando as estruturas de repetição *for*, *while* e *do..while*;
8. Crie um programa Java que imprime todos os fatoriais de 1 a 10, utilizando as estruturas de repetição *for*, *while* e *do..while*;

9. No item anterior, aumente a quantidade de números que terão os fatoriais impressos até que as respostas devolvidas estejam incorretas. Descubra o motivo que levou a essa incorretude e altere o código para que ele passe a funcionar corretamente para essa entrada;
10. Crie um programa Java que, dada uma variável  $x$  com algum valor inteiro, um novo valor é escrito em  $x$  com base nas seguintes regras:
- Se  $x$  é par,  $x = x/2$ ;
  - Se  $x$  é ímpar,  $x = 3 * x + 1$ ;
  - Imprime  $x$ .

Seu programa deve parar de imprimir  $x$  quando  $x$  alcançar o valor 1.

*Obs.: é possível provar que  $x$  sempre alcançará o valor 1 em algum momento.*

11. Crie um programa Java que imprime a seguinte tabela, usando *for*s encadeados:

```
1
2 4
3 6 9
4 8 12 16
n n*2 n*3 ... n*n
```

12. Crie um programa Java que ache o primeiro número, entre 1 e 1 milhão, que é divisível por 11, 13 e 17. Utilize o comando *break*.
13. Crie um programa Java que some todos os números, de 1 até 100, exceto os múltiplos de 5. Utilize o comando *continue*.
14. Qual a diferença entre pré-incrementar e pós-incrementar uma variável?
15. Identifique e corrija os erros em cada um dos seguintes fragmentos de código. [*Nota:* pode haver mais de um erro em cada trecho de código]

```
(a) if ( age >= 65 );
    System.out.println("Age is greater than or equal to 65");
    else
        System.out.println("Age is less than 65");
```

```
(b) int x = 1, total;
    while ( x <= 10)
    {
        total += x;
        ++x;
    }
```

```
(c) while ( x <= 100)
    total += x;
    ++x;
```

```
(d) while ( y < 0 ) {
    System.out.println ( y );
    ++y;
```

16. Os motoristas se preocupam com a quilometragem obtida por seus automóveis. Um motorista monitorou vários tanques cheios de gasolina registrando a quilometragem dirigida e a quantidade de combustível em litros utilizados para cada tanque cheio. Desenvolva um programa Java que recebe como entrada os quilômetros dirigidos e os litros de gasolina consumidos (ambos como inteiros)

para cada tanque cheio. O programa deve calcular e exibir o consumo em quilômetros/litro para cada tanque cheio e imprimir a quilometragem combinada e a soma total de litros de combustível consumidos até esse ponto. Todos os cálculos de média devem produzir resultados de ponto flutuante. Utilize a classe *Scanner* e defina uma maneira de finalizar a leitura dos tanques.

17. Desenvolva um programa Java que determina se um cliente de uma loja de departamentos excedeu o limite de crédito de uma conta-corrente. Para cada cliente, os seguintes fatos estão disponíveis:

- (a) Número de conta;
- (b) Saldo no início do mês;
- (c) Total de todos os itens cobrados desse cliente no mês;
- (d) Total de créditos aplicados ao cliente no mês;
- (e) Limite de crédito autorizado

O programa deve inserir todos esses fatos como inteiros, calcular o novo saldo ( $= \text{saldo inicial} + \text{despesas} - \text{creditos}$ ), exibir o novo saldo e determinar se o novo saldo excede ao limite de crédito do cliente. Para aqueles clientes cujo limite de crédito foi excedido, o programa deve exibir a mensagem "Limite de crédito excedido".

18. Uma grande empresa paga seu pessoal de vendas por comissão. O pessoal de vendas recebe R\$ 200,00 por semana mais 9% de suas vendas brutas durante essa semana. Por exemplo, um vendedor que realiza um total de vendas de mercadorias de R\$ 5.000,00 em uma semana recebe R\$ 200,00 mais 9% de R\$ 5.000,00, ou um total de R\$ 650,00. Foi-lhe fornecida uma lista dos itens vendidos por cada vendedor. Os valores desses itens são como segue:

Item	Valor
1	239.99
2	129.75
3	99.95
4	350.89

Desenvolva um programa Java que recebe entrada de itens vendidos por um vendedor durante a última semana e calcula e exibe os rendimentos do vendedor. Não existe nenhum limite para o número de itens que pode ser vendido.

19. Desenvolva um programa Java que determina o salário bruto de três empregados. A empresa paga as horas normais pelas primeiras 40 horas trabalhadas de cada funcionário e 50% a mais pelas horas trabalhadas além das 40 horas. você recebe uma lista de empregados, o número de horas trabalhadas que eles trabalharam na semana passada e o salário-hora de cada empregado. Seu programa deve aceitar a entrada dessas informações para cada empregado e, então, determinar e exibir o salário bruto do empregado. utilize a classe *Scanner* para inserir os dados.
20. O processo de localizar o maior valor é muito utilizado em aplicativos de computador. Por exemplo, um programa que determina o vencedor de uma competição de vendas inseriria o número de unidades vendidas por cada vendedor. O vendedor que vende mais unidades ganha a competição. Escreva um programa Java que aceita como entrada uma série de 10 inteiros e determina e imprime o maior dos inteiros.
21. Usando uma abordagem semelhante à abordagem do exercício anterior, encontre os dois maiores valores entre os 10 valores inseridos.
22. Um palíndromo é uma sequência de caracteres que é lida da esquerda para a direita ou da direita para a esquerda. Por exemplo, cada um dos seguintes inteiros de cinco dígitos é um palíndromo:

12321, 55555, 45554 e 11611. Escreva um programa Java que lê um inteiro de cinco dígitos e determina se ele é ou não um palíndromo. Se o número não for de cinco dígitos, exiba uma mensagem de erro e permita que o usuário insira um novo valor.

23. Escreva um programa Java que insere um número inteiro que contém somente 0s e 1s (isto é, um número binário) e imprime seu equivalente decimal.

24. Escreva um programa Java que lê um inteiro não negativo, calcula e imprime seu fatorial.

25. Escreva um aplicativo que estima o valor da constante matemática  $e$  utilizando a fórmula a seguir. Permita ao usuário inserir o número de termos a calcular.

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

26. Escreva um programa Java que localiza o menor de vários números inteiros. Assuma que o primeiro valor lido especifica o número de valores a serem inseridos pelo usuário.

27. Escreva um programa Java que calcula o produto dos números inteiros ímpares de 1 a 15.

28. Um varejista online vende cinco produtos cujos preços no varejo são como a seguir: produto 1, US\$ 2,98; produto 2, US\$ 4,50; produto 3, US\$ 9,98; produto 4, US\$ 4,49; e produto 5, US\$ 6,87. Escreva um aplicativo que leia uma série de pares de números como segue:

(a) Número de produto;

(b) Quantidade vendida.

Seu programa deve utilizar uma instrução *switch* para determinar o preço de varejo de cada produto. Você deve calcular e exibir o valor de varejo total de todos os produtos vendidos.

29. Calcule o valor de  $\pi$  a partir de uma série infinita

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Imprima uma tabela que mostra o valor  $\pi$  aproximado calculando os 200.000 primeiros termos dessa série. Quantos termos você tem de utilizar antes de obter um valor que comece com 3.14159.

30. Uma crítica à instrução *break* e à instrução *continue* é que elas não são estruturadas. Na verdade, essas instruções sempre podem ser substituídas por instruções estruturadas, embora fazer isso possa ser difícil. Descreva de maneira geral como você removeria qualquer instrução *break* de um loop em um programa e a substituiria por alguma equivalente estruturada.