

Lista de Exercícios de Programação Orientada a Objetos

Exercícios básicos

1. Determine as raízes de uma equação de 2º grau: $ax^2 + bx + c = 0$ (recordar que o discriminante $\Delta = b^2 - 4ac$, e que a raiz $r = (-b \pm \sqrt{\Delta})/2a$).
2. Calcule a distância entre dois pontos num espaço de 3 dimensões.
3. O programa a seguir estranhamente sempre escreve "A distancia e: 1.0". Identifique onde está o defeito.

```
import java.util.Scanner;

public class Distancia {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double x1, y1, x2, y2, distancia;

        System.out.println("Entre com as coordenadas x e y dos pontos nesta ordem:");
        x1 = teclado.nextFloat();
        y1 = teclado.nextFloat();
        x2 = teclado.nextFloat();
        y2 = teclado.nextFloat();

        distancia = Math.pow(Math.pow(x2-x1, 2) + Math.pow(y2-y1, 2), 1/2);
        System.out.println("A distância é: " + distancia);
    }
}
```

4. Para cada produto informado (nome, preço e quantidade), escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos pelo número de unidades compradas, segundo a tabela abaixo:
 - a. Até 10 unidades: valor total
 - b. de 11 a 20 unidades: 10% de desconto
 - c. de 21 a 50 unidades: 20% de desconto
 - d. acima de 50 unidades: 25% de desconto

Exercícios de estrutura de repetição

5. Construa a tabela de multiplicação de números de 1 a 10 (ex.: $1 \times 1 = 1$, $1 \times 2 = 2$, etc.).
6. Determine o número de dígitos de um número informado.
7. Considere os programas a seguir, que leem um código repetidamente e imprimem o código lido até que o código lido seja igual a -1. O código -1 não deve ser impresso.
 - a. Qual das duas soluções é a correta?
 - b. Como a solução incorreta poderia ser corrigida?

Programa A

```
import java.util.Scanner;

public classCodigo {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int codigo;

        System.out.println("Informe o código: ");
        codigo = teclado.nextInt();
        while (codigo != -1) {
            System.out.println("Código: " + codigo);
            System.out.println("Informe o código: ");
            codigo = teclado.nextInt();
        }
    }
}
```

Programa B

```
import java.util.Scanner;

public classCodigo {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int codigo;

        do {
            System.out.print("Informe o código: ");
            codigo = teclado.nextInt();
            System.out.println("Código: " + codigo);
        } while (codigo != -1);
    }
}
```

8. Calcule a série de Fibonacci para um número inteiro não negativo informado pelo usuário. A série de Fibonacci inicia com os números $F_0 = 0$ e $F_1 = 1$, e cada número posterior equivale à soma dos dois números anteriores ($F_n = F_{n-1} + F_{n-2}$). Por exemplo, caso o usuário informe o número 9, o resultado seria: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

9. Determine a saída do seguinte programa:

```
public class Loop {
    public static void main(String[] args) {
        for (int i = 2; i <= 8; i = i + 2) {
            for (int j = i; j <= 4; j++) {
                for (int k = 1; k <= j; k = k + 1) {
                    System.out.println(i + ", " + j + ", " + k);
                }
            }
        }
    }
}
```

10. Calcule o retorno de um investimento financeiro fazendo as contas mês a mês, sem usar a fórmula de juros compostos. O usuário deve informar quanto será investido por mês e qual será a taxa de juros mensal. O programa deve informar o saldo do investimento após um ano (soma das aplicações mês a mês considerando os juros compostos), e perguntar ao usuário se ele deseja que seja calculado o ano seguinte, sucessivamente. Por exemplo, caso o usuário deseje investir R\$ 100,00 por mês, e tenha uma taxa de juros de 1% ao mês, o programa forneceria a seguinte saída:

```
Saldo do investimento após 1 ano: 1280.9328043328942
Deseja processar mais um ano? (S/N)
...
```

11. Calcule a raiz quadrada aproximada de um número inteiro informado pelo usuário, respeitando o erro máximo também informado pelo usuário. Não utilize funções predefinidas.

Exercícios de variáveis compostas

12. Leia uma matriz 3 x 3 que representa um tabuleiro de jogo da velha e indique qual posição deveria ser jogada para ganhar o jogo (se possível) ou ao menos para evitar uma derrota.
13. Ordene um vetor de 100 números inteiros gerados aleatoriamente.
14. Crie dois vetores de 50 posições com valores inteiros aleatórios, ordene cada vetor individualmente, e combine os dois vetores gerando um novo vetor de 100 posições, de forma que esse novo vetor já seja criado ordenado.

Exercícios de subprogramação

15. Faça um método que calcule a média de um aluno de acordo com o critério definido neste curso. Além disso, faça um outro método que informe o status do aluno de acordo com a tabela a seguir:

Nota acima de 6 → “Aprovado”

Nota entre 4 e 6 → Conceito “Verificação Suplementar”

Nota abaixo de 4 → Conceito “Reprovado”

16. Leia do usuário o tempo em segundos e escreva em horas, minutos e segundos. Utilize cinco métodos, para a leitura e escrita de dados e para obtenção de horas, minutos e segundos a partir do tempo em segundos.
17. Leia um número decimal (até 3 dígitos) e escreva o seu equivalente em numeração romana. Utilize métodos para obter cada dígito do número decimal e para a transformação de numeração decimal para romana (Dica1: 1 = I, 5 = V, 10 = X, 50 = L, 100 = C, 500 = D, 1.000 = M; Dica2: utilize um vetor guardando a tradução para cada um dos dígitos).
18. Escreva um número por extenso aceitando números de até 9 dígitos, usando métodos para as traduções e vetores de Strings que guardam cada tradução (ex.: unidades = { “zero”, “um”, “dois”, ..., “nove” }).

Exercícios de OO

19. Identifique as classes e implemente um programa para a seguinte especificação: “O supermercado vende diferentes tipos de produtos. Cada produto tem um preço e uma quantidade em estoque. Um pedido de um cliente é composto de itens, onde cada item especifica o produto que o cliente deseja e a respectiva quantidade. Esse pedido pode ser pago em dinheiro, cheque ou cartão.”
20. Faça um programa de agenda telefônica, com as classes Agenda e Contato.

21. Faça um programa para controle de empréstimo de livros, com as classes Empréstimo, Livro e Pessoa.
22. Faça um programa para representar a árvore genealógica de uma família. Para tal, crie uma classe Pessoa que permita indicar, além de nome e idade, o pai e a mãe. Tenha em mente que pai e mãe também são do tipo Pessoa.
23. Faça um programa que calcule a área de uma figura geométrica. Aceite quatro tipos de figura geométrica: quadrado, retângulo, triângulo e círculo. Use herança e polimorfismo.
24. Adicione a funcionalidade de exibição do total de contatos registrados na agenda disponível em <https://github.com/leomurta/agenda>
25. Adicione a funcionalidade de busca por contato na agenda disponível em <https://github.com/leomurta/agenda>
26. Adicione a funcionalidade de ordenação dos contatos na agenda disponível em <https://github.com/leomurta/agenda>
27. Adicione o registro de mais de um telefone por contato na agenda disponível em <https://github.com/leomurta/agenda>
28. Adicione o registro de endereço comercial e residencial por contato na agenda disponível em <https://github.com/leomurta/agenda>
29. Explique com suas palavras como é o mecanismo de listener usado no Swing para permitir que os eventos de clique dos botões sejam tratados pelo programa.
30. Qual a diferença entre abstração, encapsulamento e modularidade?

Exercícios de Tratamento de Exceções

31. Refaça os exercícios 1, 2 e 3 para protegê-los de qualquer tipo de exceção que pode ocorrer durante a interação com o usuário.
32. Qual a diferença entre uma exceção (subclasses de Exception) e um erro (subclasses de Error)?

Exercícios de Coleções

33. Refaça os exercícios 13 e 14 usando List no lugar de vetor.
34. Refaça o exercício 17 usando Map para guardar a tradução dos números decimais para romanos.
35. Refaça o exercício 22 usando Set para guardar o conjunto de filhos de uma pessoa no lugar dos seus pais.

Exercícios de Threads

36. Faça um programa que leia um número “n” informado pelo usuário e diga quantos números primos há entre 0 e “n”. Esse seu programa deve rodar em 2 threads, de forma que o esforço computacional seja uniformemente dividido entre as threads.
37. Faça com que a funcionalidade de persistência da agenda disponível em <https://github.com/leomurta/agenda> passe a rodar em uma thread em separado, tanto para leitura dos dados, na abertura da janela, quanto para a escrita dos dados, no fechamento da janela.
38. Qual a diferença do efeito produzido pelos programas abaixo? Qual deles é mais eficiente, assumindo que o computador tem mais de um processador?

Programa A

```
Thread[] threads = new Thread[10];
for (int i = 0; i < threads.length; i++) {
    threads[i] = new Thread(new MeuRunnable());
    threads[i].start();
}

for (int i = 0; i < threads.length; i++) {
    threads[i].join();
}
```

Programa B

```
Thread[] threads = new Thread[10];
for (int i = 0; i < threads.length; i++) {
    threads[i] = new Thread(new MeuRunnable());
    threads[i].start();
    threads[i].join();
}
```

39. Para que serve o modificador synchronized? Em que situações ele deve ser usado? Por que não usar em todos os métodos do programa?
40. Qual a diferença entre o método sleep() e o método join() da classe Thread?