

Lista de Exercícios 2 de Programação Orientada a Objetos

Exercícios de OO

1. 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.
2. 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.
3. Adicione a funcionalidade de exibição do total de contatos registrados na agenda disponível em <https://github.com/leomurta/agenda>
4. Adicione a funcionalidade de busca por contato na agenda disponível em <https://github.com/leomurta/agenda>
5. Adicione a funcionalidade de ordenação dos contatos na agenda disponível em <https://github.com/leomurta/agenda>
6. Adicione o registro de mais de um telefone por contato na agenda disponível em <https://github.com/leomurta/agenda>
7. Adicione o registro de endereço comercial e residencial por contato na agenda disponível em <https://github.com/leomurta/agenda>
8. 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.
9. Qual a diferença entre abstração, encapsulamento e modularidade?

Exercícios de Tratamento de Exceções

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

Exercícios de Coleções

12. Refaça os exercícios 13 e 14 da lista 1 usando List no lugar de vetor.
13. Refaça o exercício 17 da lista 1 usando Map para guardar a tradução dos números decimais para romanos.
14. Refaça o exercício 1 desta lista usando Set para guardar o conjunto de filhos de uma pessoa no lugar dos seus pais.

Exercícios de Threads

15. 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.
16. 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.
17. 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();
}
```

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