



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Departamento de Estatística e Informática (DEINFO)

Disciplina: Paradigmas de Programação Prof: Lucas Albertins

Lista de Exercícios

2ª VA

Programação Orientada a Objeto

1. Exercícios 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 20, 24, 25, 26, 27, 28 (*Problem Set Cap12*) do livro do Sebesta

Programação Orientada a Aspecto

2. Que tipos de problemas no código fonte a orientação a aspecto pode resolver?
3. Explique a principal vantagem do uso de aspectos no desenvolvimento de software.
4. Análise o seguinte aspecto e explique o que ele faz:

```
public aspect JapaneseMannersAspect {  
  
    pointcut callSayMessageToPerson (String person)  
  
        : call (* HelloWorld.sayToPerson (String, String))  
  
        && args (*, person);  
  
    void around (String person)  
  
        : callSayMessageToPerson (person) {  
  
        proceed (person + "-san");  
  
        }  
  
}
```

Programação Concorrente

5. Exercícios 13.4.2, 13.4.4, 13.4.5, 13.4.7 do livro do David Watt
6. Qual a forma de utilização de semáforos em Java para implementar programas concorrentes?



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Departamento de Estatística e Informática (DEINFO)

Disciplina: Paradigmas de Programação Prof: Lucas Albertins

7. Explique problemas do uso de threads em Java para implementar programas concorrentes orientados a objeto, e quais são as formas de minimizar tais problemas e suas consequências.
8. Explique o conceito de exclusão mútua e técnicas que implementem esta característica.
9. Implemente as seguintes aplicações tanto em Java quanto em Go:
 - a. **Produtor-Consumidor:** o problema do produtor-consumidor simples (um consumidor e um produtor), depois expanda para vários produtores e vários consumidores compartilhando o mesmo buffer;
 - b. **Máquina de venda automática:** uma máquina de venda automática que recebe moedas e retoma chocolates. Imagine que diferentes compradores possuem moedas de valor 1, 2 e 5. Os compradores devem escolher o item, depois inserir dinheiro e esperar que a máquina entregue o item. Lembre-se de projetar a máquina e os compradores como processos separados;
 - c. **controle de acesso a impressoras 3D:** durante um hackaton, impressoras 3D são altamente utilizadas. Escreva uma simulação de hackers tentando acessar impressoras 3D. Assuma que existem 3 impressoras 3D e 7 hackers interessados em utilizá-las. Se um hacker não consegue acesso a impressora por mais de 5 segundos, ele fica chateado e sai do hackaton. Hackers usam impressoras durante um intervalo aleatório de 1 a 10 segundos e normalmente eles precisam usar a impressora no mínimo duas vezes porque nada sai perfeito na primeira vez.

Programação Funcional

10. Exercícios 1-20 (*Programming Exercises Cap15*) do livro do Sebesta usando Haskell como a linguagem de implementação.
11. Exercícios 14.1.1, 14.1.2, 14.3.1 até 14.3.12 (cap14) do livro do David Watt

Programação Lógica

12. Exercícios 3, 4, 5 (*Problem Set Cap16*) e 1, 2, 3, 4, 5, 6 (*Programming Exercises Cap16*) do livro do Sebesta
13. Exercícios do cap15 do livro do David Watt



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Departamento de Estatística e Informática (DEINFO)

Disciplina: Paradigmas de Programação Prof: Lucas Albertins