

Laboratório 12

Exercícios com monitores (em Java)

Computação Concorrente (MAB-117)
Prof. Silvana Rossetto

¹DCC/IM/UFRJ — 24 de novembro de 2016

Introdução

O objetivo deste Laboratório é praticar o uso dos mecanismos de sincronização em Java. Para cada atividade, siga o roteiro proposto e responda às questões colocadas.

Atividade 1

Objetivo: Problema do produtor/consumidor com monitor em Java.

Roteiro: Abra o arquivo **PC.java** e siga o roteiro abaixo.

1. Complete a implementação dos métodos `Inserir` e `Remover` da classe `Buffer`.
2. **Inclua código adicional para geração de log da execução de modo que seja possível verificar a sua correteza.**
3. Execute o programa **várias vezes** e verifique se a execução está correta.
4. Varie o número de threads consumidoras e produtoras, fazendo: (a) um produtor e um consumidor; (b) um produtor e vários consumidores; (c) vários produtores e um consumidor; (d) vários produtores e vários consumidores. **Verifique se a execução do programa está sempre correta.**
5. [Mostre o programa executando para a professora.](#)

Atividade 2

Objetivo: Propõe uma variação na implementação do problema produtor/consumidor.

Roteiro: Implemente a seguinte variação do problema produtor/consumidor: a cada execução de um consumidor, ele consome o buffer inteiro, e não apenas um único item (para isso ele deve esperar o buffer ficar completamente cheio). O produtor continua com a mesma lógica, i.e., insere um item de cada vez. Varie o número de threads consumidoras e produtoras, fazendo: (a) um produtor e um consumidor; (b) um produtor e vários consumidores; (c) vários produtores e um consumidor; (d) vários produtores e vários consumidores. **Inclua código adicional para geração de log da execução de modo que seja possível verificar a sua correteza.** [Mostre o programa executando para a professora.](#)

Atividade 3

Objetivo: Implementar e testar uma solução para o problema dos “leitores e escritores” usando o conceito de monitor em Java.

Roteiro: Considere o seguinte protótipo para a classe que implementa o monitor:

```
class LE {  
    //qtde de leitores e escritores lendo/escrivendo  
    private int leit, escr;  
    LE() { ... } //construtor  
    // Entrada para leitores  
    public void EntraLeitor () {...}  
    // Saida para leitores  
    public void SaiLeitor () {...}  
    // Entrada para escritores  
    public void EntraEscritor () {...}  
    // Saida para escritores  
    public void SaiEscritor () {...}  
}
```

1. Implemente o protótipo do monitor LE;
2. Implemente os códigos das threads leitoras e escritoras;
3. Implemente a classe principal;
4. **Inclua código adicional para geração de log da execução de modo que seja possível verificar a sua corretude.**
5. Teste sua aplicação.

Mostre o programa executando para a professora.