UNIVERSIDADE CATÓLICA DE BRASÍLIA – UCB
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA
EDSON FRANCISCO DA FONSECA

Lista de Exercícios 4
Programação Concorrente

1. Uma padaria possui 3 padeiros e apenas um forno. Os padeiros trabalham ao mesmo tempo e cada um segue a rotina de fazer a massa do pão e assar o pão. O padeiro não realiza nenhuma outra atividade enquanto o pão assa. Como o forno é pequeno, comporta apenas um pão por vez. Implemente um programa que simule o comportamento dos padeiros. Para tanto:
   - Crie as classes Padeiro (thread), Pão e Forno;
   - O pão deve ser criado pelo método “fazer a massa” do padeiro;
   - O pão deve possuir um atributo assado (booleano) que deve ser alterado pelo forno;
   - Utilize temporizadores (sleep) aleatórios (random) para simular o tempo de execução das ações;
   - Imprima mensagens em pontos estratégicos para acompanhar o comportamento da aplicação;
   - Não implemente nenhum mecanismo de controle de concorrência (monitor, lock e semáforo) ou flag no forno e observe se os padeiros colocarão mais de um pão por vez no forno.

2. Refaça o exercício anterior, mas agora garanta que apenas um pão seja assado por vez. Para tanto:
   - Implemente o mecanismo de controle de concorrência por monitor.

3. Refaça o exercício anterior, mas troque o mecanismo de controle de concorrência. Para tanto:
   - Implemente lock.

4. Refaça o exercício anterior, evoluindo para que cada pão possua um peso em gramas (1kg = 1.000g) e o tempo de forno não seja aleatório, mas proporcional ao peso do pão (10 milésimos de segundos por grama). O tempo para fazer o pão também deve ser proporcional ao peso (5 milésimos de segundos por grama). Considere que o fogão possui um botijão de gás com autonomia de 5 minutos, ou seja, é possível o forno funcionar por 5 minutos com um botijão de gás. Ao ser utilizado o forno consome gás até que o gás termine. Quando o gás termina o gerente da padaria deve ser chamado para trocá-lo. Quando o gerente termina de trocar o gás, ele deve informar aos padeiros. Para tanto:
   - Volte a utilizar monitor;
   - Crie as classes Botijão e Gerente (thread);
   - Utilize um mecanismo de notificação (wait e notify/notityAll);
   - Verifique o que ocorre ao utilizar notify e notityAll com atenção para o caso de um padeiro possuir pão assando. Veja o que ocorre após a troca do gás.

5. Refaça o exercício anterior, mas agora tente garantir que quando o botijão de gás for trocado o forno volte/continue assando o pão que estava nele antes do gás acabar. Para tanto:
   - Utilize prioridades (.setPriority()) diferentes. Tente colocar os padeiros com baixa prioridade e no momento que o gás acabar durante o assar de um pão, aumentar muito a prioridade deste padeiro. Lembre-se de voltar o valor da prioridade após o pão assar;
   - Verifique o que ocorre ao utilizar notify e notityAll.

6. Refaça o exercício anterior, mas troque o mecanismo de controle de concorrência. Para tanto:
   - Implemente lock;
   - Retire as prioridades (.setPriority()) diferentes;
   - Utilize um mecanismo de notificação (await e signal/signalAll) com 2 condições diferentes, uma para os padeiros e outra para o gerente;
   - Tente implementar uma fila (FIFO) no lock.

7. O gerente da padaria cursa graduação em Administração de Empresa na UCB. Graças aos conhecimentos adquiridos, resolveu fazer reengenharia dos processos de produção para reduzir o tempo do padeiro parado enquanto o pão assa e ao mesmo tempo introduzir novos produtos no menu da padaria, além do pão simples, que são pão recheado, pão confeitado e pão esculpido artesanalmente. No novo processo ao terminar de fazer um pão, o padeiro verifica se o forno está livre, se não tiver, vai rechear este pão e verificar se o forno está livre, se não tiver, vai confeitar este pão e verificar se o forno está livre, se não tiver, vai esculpir artesanalmente este pão e, desta vez, esperar até que o forno esteja livre. Em qualquer momento que o forno esteja livre o pão é assado. Refaça o exercício anterior, aceitando o novo processo de produção. Os tempos de rechear, confeitar e esculpir o pão deve ser proporcional ao peso, sendo respectivamente 3, 5 e 7 milésimos de segundos por grama. Para tanto:
   - Crie um atributo “tipo” no pão para indicar o tipo: simples, recheado, confeitado e esculpido;
   - Reveja o método de bloqueio do lock (blocante). Considere utilizar também um não blocante (tryLock).

8. O gerente da padaria concluiu o curso de Administração e resolveu cursar Ciência da Computação na UCB, onde descobriu as lojas autônomas, sem funcionários, em que o próprio cliente se serve. Então, resolveu tornar sua padaria autônoma. No novo modelo há uma vitrine para exposição de 30 pães e o cliente pega o pão que lhe interessar. Contudo, o gerente tinha preocupação com a segurança e funcionamento do sistema automático de detecção de compra. Então, limitou a entrada de clientes na padaria a no máximo 50 por vez e se este número fosse atingido os clientes formariam uma fila na entrada da padaria.
   Sem esquecer seus conhecimentos de processo e preocupado com o frescor e qualidade de seus produtos, o gerente resolve que os padeiros somente irão produzir novos pães se houver espaço disponível na vitrine. Ainda preocupado com processo, qualidade do serviço e conforto do cliente, o gerente resolve que caso o cliente deseje um pão (independente do tipo) e não haja nenhum pão na vitrine este irá se sentar confortavelmente na área de

descanso da padaria e será informado por um padeiro assim que um pão for posto na vitrine. Para tanto:

- Antes de implementar as novas funcionalidades, melhore a organização da aplicação (POO), criando a classe Padaria com os atributos gerente, padeiros (List) e forno. Crie um construtor que receba o número de padeiros (3). Teste e garanta que esteja funcionando corretamente como antes;
- Crie um atributo vitrine do tipo array (Pao[] vitrine) na padaria. Altere o construtor da padaria para que também receba a capacidade da vitrine (30). Considere a questão de thread-safe;
- Altere o construtor da padaria para que também receba o limite de clientes na padaria (50) e garanta que esse limite seja respeitado. Pense na possibilidade de utilizar semáforo.