

Lista de Exercícios

MC302 - Programação Orientada a Objetos
Instituto de Computação
Universidade Estadual de Campinas

Programação Concorrente
2011
André Santanchè e Ivo Koga

Questão 1

Crie uma classe Contador, como uma subclasse de Thread, que imprima números de 0 a 10. Crie uma classe TesteContador que deve definir o método main responsável pela criação e execução do thread Contador. Depois de finalizado este programa, faça as seguintes variantes:

- Altere as classes Contador e TesteContador de modo que a classe Contador seja definida como uma implementação da interface Runnable.
- Agora altere o método main da classe TesteContador para criar dois ou mais threads Contador e inicialize a execução dos mesmos.

Dicas:

- Na última parte do exercício, onde é pedido para alterar o método main da classe TesteContador, você deve inicialmente criar os threads e em seguida inicializá-los, de modo a obter um melhor resultado na visualização da execução concorrente.
- Como o loop do exemplo é apenas de 0 a 10 pode ser que não seja visualizado um interleaving entre as execuções dos threads. Caso isto ocorra aumente o tamanho do loop.

Questão 2

Escreva um programa que simule um chat entre dois indivíduos virtuais da seguinte maneira:

Considere que existem dois arquivos texto de nome `personagem1.txt` e `personagem2.txt` que contêm os diálogos dos dois personagens principais. Existem duas threads da mesma classe Chat simulando dois terminais de chat A e B. No primeiro ciclo de chat a thread A lê uma linha do arquivo de texto e manda para a thread B. A thread B fica aguardando até receber a mensagem de A. Neste momento, imprime a mensagem recebida na tela e repete o processo no sentido inverso (lê uma linha do seu arquivo e envia para A). Este processo se repete até que um dos arquivos chegue ao fim. Para fins de simplificação considere que existe um método estático `proximaLinha()` da classe Arquivo que recebe como parâmetro o nome do arquivo e retorna uma String contendo a próxima linha de texto. O próprio método controla em que linha ele está em cada arquivo.

Expansão:

Depois de implementar este programa estenda a implementação para três threads. Neste caso as mensagens deverão fluir de forma circular (A para B, B para C e C para A).

Questão 3

Implementar uma classe que gerencie uma pilha de caracteres preparada para atender duas threads: uma que coloca caracteres na pilha (Produtor) e outra que retira a caracteres da pilha (Consumidor). O processo deve funcionar de tal maneira que o Produtor só insere caracteres na pilha se ela não estiver cheia. Se ela estiver cheia, ele aguarda que o Consumidor retire alguma coisa da pilha para que vague uma posição. Em contrapartida, o Consumidor verifica se há elementos na pilha. Se ela estiver vazia, ele aguarda que algo seja colocado para que ele possa consumir.

Escreva um programa de teste no qual o Produtor gere uma sequência de 1.000 números aleatórios para colocar na pilha. O Consumidor deve imprimir o que ele retira na tela.

Questão 4

Escreva um programa para simular o processo de vendas e produção de uma empresa, que vende 10 tipos de produtos diferentes codificados de 0 a 9. Este programa deverá definir duas classes cujos objetos serão executados em threads diferentes:

Vendas - Solicita pelo teclado o código do produto a ser vendido e sua quantidade e despacha o pedido para a produção. Este processo se repete indefinidamente.

Producao - Aguarda pedidos de produção, os registra e quando possível aciona a sua produção em lotes. Os produtos pedidos só são produzidos em lotes de 20 unidades. Para cada 20 unidades pedidas de um tipo de produto X o módulo de produção deve disparar o seguinte método:

produzLote20	Recebe como parâmetro o código do produto e o número do lote e efetua a produção do mesmo.
---------------------	--

Considere que este é um método estático da classe Fabrica que já existe e que você não precisa implementar.

Por exemplo, suponha que inicialmente é feito um pedido de 7 unidades de X, depois outro pedido de 15 unidades de X. Ao receber o segundo pedido, a classe de produção dispara produzLote20 para produzir 20 unidades de X. As duas unidades restantes farão parte do próximo lote de produção.

Note-se que o método produzLote20 produz um tipo de produto para cada vez que é chamado, ou seja, não é possível se somar unidades de produtos diferentes no mesmo lote de produção. Além disso, o método sempre produz exatamente 20 unidades (nem uma a mais, nem uma a menos).

O número do lote enviado para o método produzLote20 é um número sequencial que inicia de zero e é incrementado independentemente do tipo de produto produzido.

Existe uma política da fábrica de que não se podem acumular mais de 3.000 unidades pedidas (independente do tipo) sem produção. Ou seja, por este motivo, a sua classe deve ser capaz de realizar uma pausa nas vendas de produtos cada vez que o total de unidades solicitadas ultrapassar este limite.

Escreva um programa principal que instancie e execute as duas classes (Vendas e Producao) em threads independentes.

Questão 5

Escreva um programa para simular o funcionamento de um restaurante. Este programa deve definir duas classes, cujos objetos funcionarão em threads independentes, conforme descrito a seguir.

Caixa - Solicita pelo teclado uma String que contém o nome do prato comprado e coloca na lista de pedidos pendentes. Este processo se repete indefinidamente. Cada item da lista possui o nome de um único prato. Como só existe espaço para 20 pedidos na lista de pedidos pendentes, ao alcançar este número o caixa deve dar uma pausa nas vendas e aguardar que sejam retirados pedidos da lista para poder prosseguir.

Cozinha - Retira pedidos da lista de pedidos pendentes e aciona o seu preparo. Por uma questão de eficiência todos os pedidos para o mesmo prato devem ser feitos em conjunto. Por este motivo, a cada rodada o objeto Cozinha deve retirar da lista de pedidos pendentes todos os pedidos com o mesmo nome do prato e acionar o método fazPrato que simula a produção de pratos. Este processo se repete indefinidamente.

fazPrato	Recebe como parâmetros o nome do prato e a quantidade de pratos a ser produzida e providencia a sua produção. Este método produz um único tipo de prato cada vez que é chamado.
-----------------	---

Considere que este é um método estático da classe Cozinheiro que já existe e que você não precisa implementar.

As classes Caixa e Cozinha devem ser construídas de forma que sejam executadas em paralelo. Escreva um programa principal que instancie e execute as duas classes (Caixa e Cozinha) em threads independentes.

Podem ser implementadas classes auxiliares que sejam necessárias no processo.