

1. Introdução

Nesta prática vamos utilizar os conceitos de programação utilizando Sockets e Threads para criar uma “sala de bate-bapo” onde vários usuários podem se conectar. O objetivo é implementar dois **programas** diferentes: um servidor e um cliente.

Servidor

Nosso servidor será um programa no modelo produtor/consumidor. Teremos várias Threads produtoras, sendo que cada cliente que entrar na sala terá uma thread produtora correspondente. O que estas threads produzem? Elas produzem objetos da classe Mensagem, que é a primeira classe que você deve implementar. A classe Mensagem tem um texto, que é a mensagem enviado por um usuário, tem o nome (nick) do usuário que enviou, e tem a hora que esta mensagem chegou. Os objetos da classe mensagem serão colocados em uma **fila única** que será gerenciada pelo servidor. Você deve utilizar a mesma classe Fila da última atividade com Threads. Como serão produzidas as mensagens? Cada thread produtora tem um objeto Socket que está conectado diretamente com seu usuário correspondente. O método run da thread produtora ficará em um loop “infinito”, lendo mensagens vinda do cliente (via readUTF() da classe Socket). Ao receber uma mensagem, a produtora cria um objeto mensagem e o coloca na Fila de mensagens. Depois disso, ela volta a ficar na espera por uma nova mensagem do cliente.

Você deve criar uma classe principal Servidor. Um objeto Servidor será responsável por: 1) criar a Fila única de mensagens. 2) Receber solicitações de usuários para entrar na sala e criar a thread produtora correspondente ao usuário novo. 3) Criar a thread consumidora.

Como a classe Servidor recebe solicitações de usuários? Esta classe tem um método principal chamado Execute. Neste método teremos também um loop “infinito” onde um ServerSocket ficará na escuta de novas conexões. Ao receber uma solicitação o servidor cria uma nova conexão (Socket), cria um objeto da classe produtora, passa o objeto socket para esse objeto, passa a Fila de mensagens para o objeto, cria uma thread

produtora e coloca essa thread para rodar. Depois, o servidor volta novamente a ficar na escuta de novas conexões.

O servidor também deve criar **uma** thread consumidora. O que a consumidora faz? Ela apenas tira as mensagens da fila, uma por uma, e imprime o texto na própria tela do servidor. Obviamente, a consumidora tem de ter acesso à **fila única** de mensagens.

Cliente

Nosso programa cliente é mais simples que o servidor. O método main do cliente, em primeiro lugar, deve criar uma conexão com o servidor. Logo após, ele ficará em um loop infinito também. Neste loop, o programa lê uma mensagem do usuário (use o JOptionPane) e envia para o servidor.

Observe que esta é uma versão simplificada do Bate Papo por que o servidor não devolve as mensagens para os clientes. Porém, ao desenvolver este versão inicial, faltará pouco para você fazer a aplicação completa. Algumas das tarefas que você deve realizar neste sentido incluem: 1) manter no servidor uma lista de objetos da classe produtora (com isso, o consumidor pode percorrer esta lista para devolver as mensagens aos clientes). 2) Criar uma thread no programa cliente que ficará apenas recebendo e imprimindo na tela as mensagens provenientes do servidor.

Bom trabalho!!