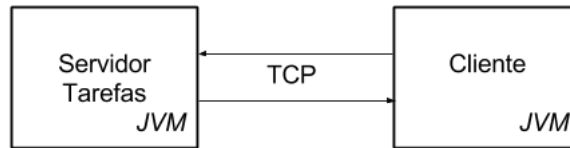


Dicas de Eclipse

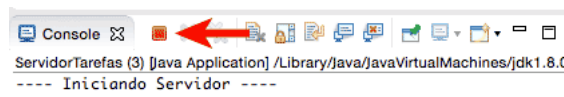
Nesse treinamento vamos trabalhar em uma aplicação *Client-Server*, ambos apresentados por sua JVM rodando:



Em outras palavras, o tempo todo você precisa inicializar e parar a execução da JVM dentro do Eclipse.

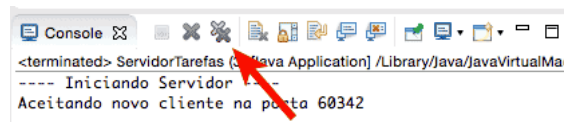
É fundamental que você domine esse trabalho com Eclipse, isto é, saber *inicializar e parar as JVMs ativas e saber vincular um console com a JVM em execução*. Seguem algumas dicas:

1) Quando você roda uma classe com método `main`, o processo da JVM é representado pelo console no Eclipse. Por exemplo, ao rodar a classe `ServidorTarefas` você pode ver um console ativo no Eclipse. Faça o teste e rode a classe (se não tiver rodando ainda):



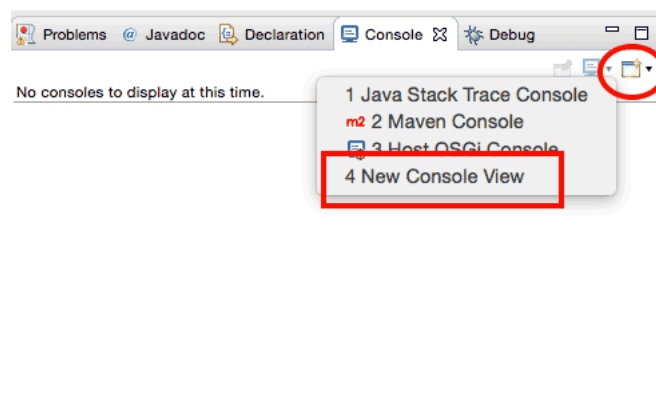
Para parar a JVM em execução basta apertar o botão vermelho em cima.

2) Ao apertar, você desliga a JVM mas *ainda não elimina o console*. Para **fechar todos os consoles inativos** aperte o botão **XX** :



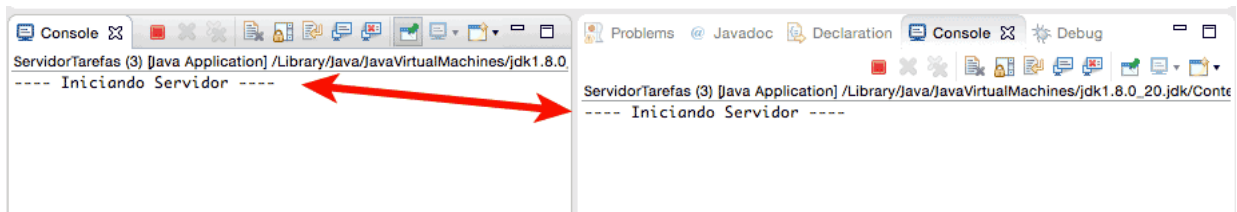
Usando o botão **XX** você consegue verificar facilmente se ainda há alguma JVM rodando, pois o botão apenas tira os consoles inativos. Use esse botão com frequência!

3) Outro ponto importante é saber como associar um console com uma JVM específica. Lembrando que você precisa rodar o tempo todo a JVM do servidor E a JVM do cliente. Nas próximas aulas usaremos sempre *duas janelas de Console* no Eclipse. Você pode abrir um novo console pelo botão no lado direito da *view console*:

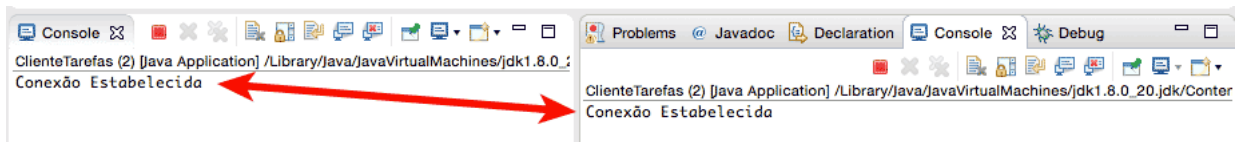


Abra uma nova console agora!

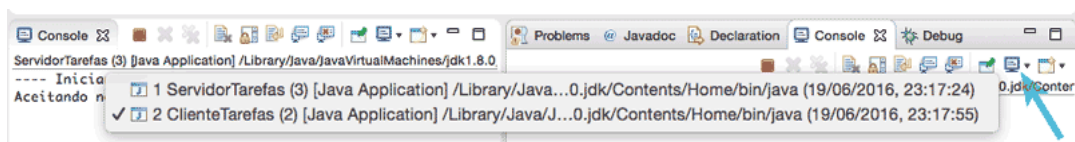
4) Agora pare todas as JVMs ativas e rode novamente a classe `ServidorTarefas` (apenas essa classe). Repare que os dois consoles mostram a mesma saída, da classe `ServidorTarefas` :



5) Agora rode também a classe `ClienteTarefas` :

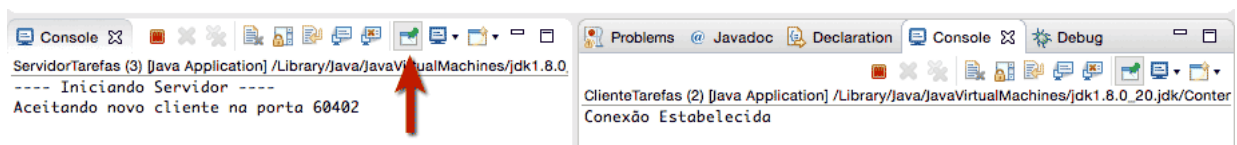


O que queremos é ver em cada console a saída de uma JVM diferente! Para tal, escolha qual *JVM deve ser associada com o console*:



Cada console deve então ficar vinculado com uma JVM dedicada (um console com a JVM servidora, outro com a JVM do cliente).

6) Podemos deixar o vínculo entre JVM e Console mais "firme" ao usar o botão *Pin* do console:



Teste o botão, ele garante que o console sempre mostrará a saída da mesma JVM.