

## Trabalho Prático 3 (TP 03) – Disciplina de Sistemas Distribuídos

Neste trabalho o aluno utilizará o conceito de tolerância a falhas para construir sua 3ª versão de uma solução distribuída.

Qualquer linguagem será aceita.

TEMA: Implementar o padrão de projeto de software Observer, contudo distribuído e tolerante a falhas no subject. Informações detalhadas sobre tal padrão em [http://en.wikipedia.org/wiki/Observer\\_pattern](http://en.wikipedia.org/wiki/Observer_pattern)

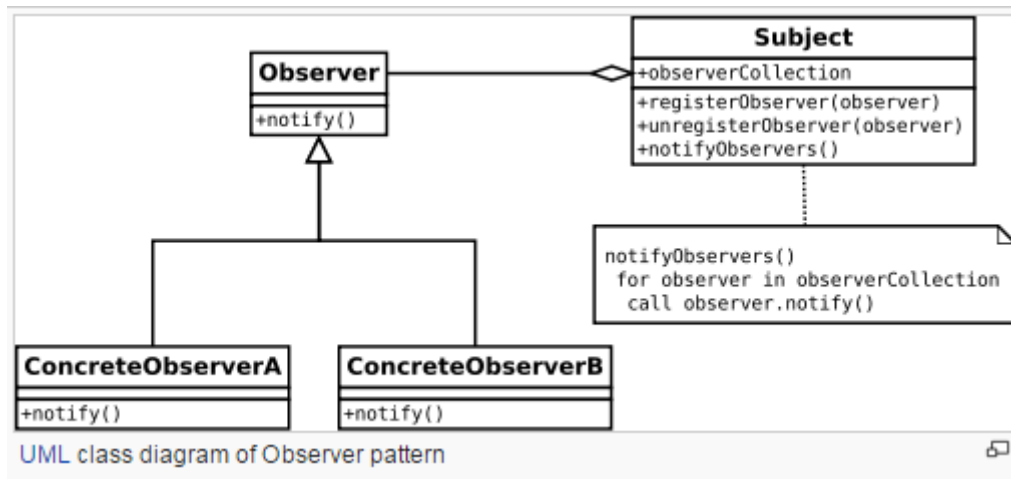


Figura retirada de [http://en.wikipedia.org/wiki/Observer\\_pattern](http://en.wikipedia.org/wiki/Observer_pattern)

Nossa aplicação terá uma máquina armazenando o subject que sofre alterações aleatórias de  $n$  processos remotos, portanto note que há concorrência de escrita e também há tolerância a falhas neste contexto. Para simular este requisito, iremos implementar um subject que é uma coleção de pontos, sendo que cada ponto possui as propriedades cor, tamanho e posição X-Y. Novos pontos são inseridos e removidos dos  $n$  processos remotos, assim como suas posições alteradas recorrentemente no subject.

Os observers são aplicações com interface gráfica onde os pontos do subject são plotados sistematicamente. Um observer se registra no subject e passa a receber notificações quando o subject é alterado. Cada conjunto de alterações no subject gera um frame que precisa ser notificado e encaminhado aos observers. Feche um frame a cada  $M$  milissegundos e mantenha dois frames (um para escrita e outro para ser encaminhado), evitando escritas durante envio de frame.

No nosso exemplo, os observers receberão todos os pontos novamente para serem replotados ou somente as mudanças ocorridas no subject a cada frame. Certamente, a segunda alternativa é mais complexa, será mais rápida e valerá mais pontos. Tente mostrar o tempo médio de recebimento de frames nos observers, realçando que o usuário irá observar tempos variados, o que indica que a QoS, sob ponto de vista dos usuários que visualizam os frames, não está boa!!

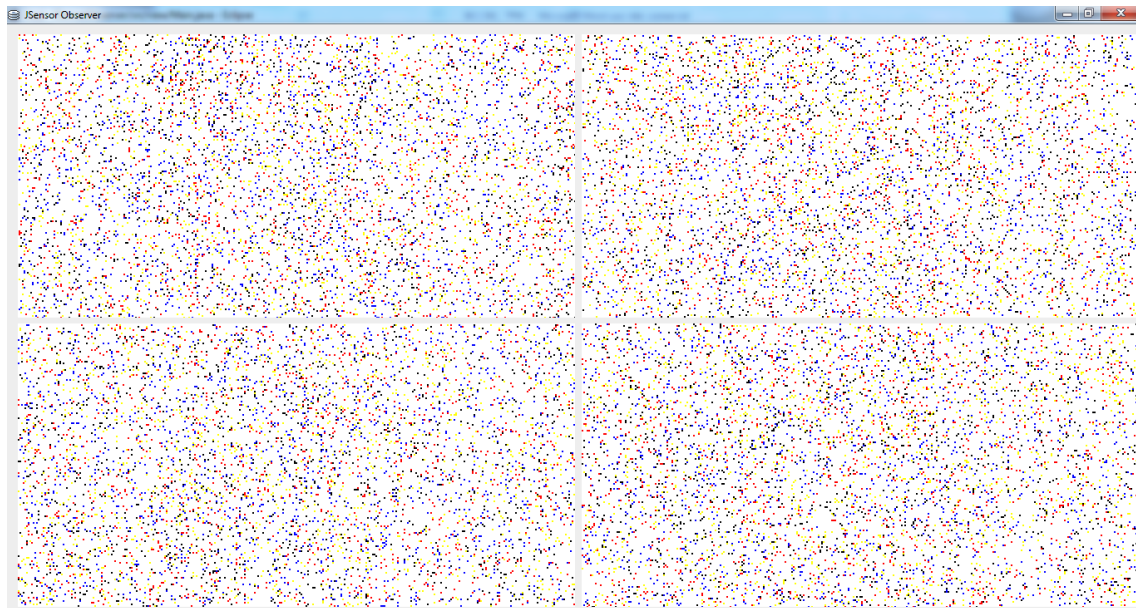


Figura 1: Exemplo de Observer com GUI com 4 áreas de rendering de pontos independentes

Haverá sempre um subject redundante e caso o primário falhe, o secundário assume. Haverá política de recuperação também, ou seja, o primário em off volta a estar pronto para uso, contudo o secundário já é o MASTER e contratos precisam ser reestabelecidos entre os SUBJECTS. Com 3 subjects o TP atinge níveis ótimos de resiliência, pois sempre haverá dois subjects após queda de algum. Note que na Figura 1 há 4 áreas para rendering. Isto foi feito para aumentar a concorrência da tarefa de rendering de pontos na GUI, pois podemos ter 4 e não apenas 1 thread fazendo rendering de cada frame recebido.

**USAR LAB 30 OU LAB 22 OU QUALQUER LAB. USAR MAIS DE 10 MÁQUINAS COM OBSERVERS E 2 COM SUBJECT.**

**NÃO USAR INPUTS PEQUENOS. SE PLANEJE PARA NÃO FICAR SEM LAB E SEM TESTAR SUA SOLUÇÃO!!**

A apresentação é fundamental, portanto atenção a sua confecção.

Neste TP03 devemos

- 1) Descrever a aplicação. Descrever como a solução foi implementada (usando mensagens, tolerância a falhas, tcp-udp, etc.). Conte sobre como o OBSERVER-SUBJECT distribuído e tolerante a falhas funciona. Conte como mediu os tempos e quais foram os gargalos do sistema. Use as métricas speedup, eficiência e outras para descrever a escalabilidade do seu sistema.
- 2) Testar a solução aumentando o número de máquinas Observers, fixando o número de processos remotos que alteram o subject e fixando a quantidade de pontos no Subject (note que os pontos podem mudar de cor, tamanho e posição. Apenas quantidade é proibitiva). Testar fixando o número de máquinas Observers no máximo e variando o número de pontos do subject (pode também variar, cor, posição e tamanho). Testar fixando o número de máquinas Observers no máximo, fixando a quantidade de pontos

no Subject no máximo (note que os pontos podem mudar de cor, tamanho e posição. Apenas quantidade é proibitiva) e variando a quantidade de processos remotos que alteram o subject. Monte gráficos dos cenários explicados anteriormente, explique a metodologia adotada e apresente justificativa para os resultados obtidos.