



Ciência da Computação
Técnicas Avançadas de Programação – TAP
Professor: Anderson Elias

PROJETO TAP – Paralelismo e Concorrência (Threads)

1. O projeto é realizado de forma individual e deve desenvolver as competências que segue abaixo.
2. Montar um projeto em java para realizar a dinâmica de um cenário definido em sala de aula junto com o professor.
3. Este cenário não deve ser o mesmo de outro aluno. Obrigatoriamente tem de ser único.
4. O cenário fica a seu critério, mas, que se possa utilizar de algum contexto comum de nossa sociedade onde tenha uma oportunidade de conseguir demonstrar os eventos paralelos e concorrentes para uma melhor abordagem. Em outras palavras, algum cenário onde os processos que acontecem no mundo real de forma paralela e concorrente. Lembre-se: deve ser validado antes com o professor.
5. O projeto deverá utilizar um padrão de projeto simples como o MVC (*Model, View e Controller*) contendo cada classe/entidade em seu devido pacote além de um pacote de tratamento de exceção para reportar as mensagens de erros personalizadas.
6. Utilize classe genérica (*generics*) para identificar características do seu cenário a fim de prover o paralelismo e a concorrência e que seu padrão de projeto está de acordo com o âmbito do seu negócio.
7. Implemente uma boa prática para exceções do seu projeto, que por ventura podem ser levantadas e estas devem ser personalizadas com mensagens mais intuitivas para os usuários.
8. Enriqueça seu projeto com a implementações de interfaces advindas de *Collections*.
9. Seu projeto deverá ser totalmente com interfaces gráficas onde a(s) tela(s) deve(m) refletir o cenário abordado.
10. A simulação do ambiente do cenário em pleno funcionamento se dará por um tempo entre 2 e 3 minutos para que possamos em uma escala menor, refletir as operações realizadas como sendo 8hs de trabalho.
11. Obrigatoriamente é essencial o uso de componentes barras de progressos (*progress bar*) para que se possa visualizar o andamento do processamento da thread em questão. A quantidade de barras de progressos será de acordo com a quantidade de núcleos o computador tem. Isto é definido dinamicamente através da captura do número de núcleos.
12. As barras de progressos receberão novos processamentos até que o tempo limite máximo possa ser fechado.
13. O projeto deverá conter os pontos abordados em sala de aula para controle do paralelismo explícito como: Uso correto dos construtores, priorizar determinadas tarefas serem executadas antes de outras, pausar por um tempo determinado e também por tempo indeterminado as tarefas, levantar as tarefas que estejam inativas, sincronizar tarefas, permitir que apenas uma tarefa possa ser executada após uma já estiver finalizada ou além disto quando determinado tempo se passou, implementar interfaces que se utilize de thread ou que herde dela, interface gráfica com barras de progressos e rótulos, um tratamento de eventos de exceção que possa acontecer.
14. Tanto do lado esquerdo, quanto do lado direito da barra de progresso, deve apresentar informações do que está sendo processado.
15. O projeto deve conseguir ser executado e após o término de todas as execuções dos processos, o projeto deve apresentar um relatório informando os pontos contabilizados de cada situação do seu cenário.

Façam com bastante atenção.
Abraços.