

Lab9 – Using BlockingQueue – 24.2

Objetivo

Neste lab vocês irão experimentar o uso da interface `BlockingQueue` do pacote `java.util.concurrent` Package de Java. Nesse contexto, desenvolva este laboratório seguindo as etapas abaixo:

Etapla 1. No diretório **src/etapa1**, desenvolva um programa concorrente que implementa um pipeline com um produtor que gera números aleatórios entre 1 e 10 e um consumidor que avalia os números gerados e os exibe na saída padrão. Tanto o produtor quanto o consumidor devem receber como parâmetro um tempo (em milisegundos) que representa o tempo necessário para produzir/consumir o número. A fila compartilhada entre o produtor e consumidor deve ser uma `BlockingQueue` implementada no tipo `ArrayBlockingQueue`. Neste caso, tanto o produtor como o consumidor executam sem parar. Experimente diferentes tempos para produção e consumo dos números. No mesmo diretório, crie o arquivo `comentarios.txt` e comente o que acontece quando o tempo de produção é maior que o de consumo e vice versa.

Etapla 2. No diretório **src/etapa2**, a partir da solução anterior, modifique o código de tal forma que o produtor deve gerar 10.000 números aleatórios e o consumidor deve ter um timeout de 600

milissegundos na espera por números na fila. Dessa forma, espera-se que o programa não execute sem parar. No mesmo diretório, crie o arquivo `comentarios.txt` e comente o que acontece quando você executa com as configurações de tempo que você testou.

Etapa 3. No diretório `src/etapa3`, a partir da solução anterior, modifique o código de tal forma que a fila compartilhada entre o produtor e consumidor siga a política baseada em prioridade. Dessa forma, o consumidor examinará os itens seguindo sua prioridade ao invés de seu momento de inserção. No mesmo diretório, crie o arquivo `comentarios.txt` e comente o que acontece quando você executa esta nova versão do programa.

Apesar de não ser obrigatório, fiquem à vontade para usar `ExecutorService` para o gerenciamento das threads produtora e consumidora em suas soluções.

Visão geral do código base

<https://github.com/giovannifs/fpc/tree/master/2024.2/Lab9>

Entrega

Você deve criar e manter um repositório privado no GitHub com a sua solução. No entanto, a entrega do laboratório deverá ser realizada por meio de submissão online utilizando o script `submit-answer.sh`, disponibilizado na estrutura de arquivos do próprio laboratório. Uma vez que você tenha concluído sua resposta, seguem as instruções:

- 1) Crie um arquivo `lab9_matr1_matr2.tar.gz` com o seu código fonte. Para isso, supondo que o diretório raiz é `Lab9/src`, você deve executar:

```
tar -cvzf lab9_matr1_matr2.tar.gz Lab9/src
```

- 2) Submeta o arquivo `lab9_matr1_matr2.tar.gz` usando o script `submit-answer.sh`, disponibilizado no mesmo repositório do laboratório:

```
bash submit-answer.sh lab9 lab9_matr1_matr2.tar.gz
```

Prazo

01/abr/25 16:00