



Lista 6

1. Criando Threads e Definindo Prioridades

Crie **três threads**, nomeando-as como "Alta Prioridade", "Média Prioridade" e "Baixa Prioridade", e defina suas prioridades como `MAX_PRIORITY`, `NORM_PRIORITY` e `MIN_PRIORITY`, respectivamente.

Cada thread deve imprimir seu nome 10 vezes. Observe a ordem de execução e analise se a prioridade influencia o comportamento do sistema.

2. Sincronização de Acesso a um Contador Compartilhado

Crie uma classe **Contador** com um atributo `int valor`, inicializado em 0. Duas threads (**Incrementador** e **Decrementador**) devem, respectivamente, **incrementar e decrementar o valor 10000 vezes**.

Sem synchronized, o resultado final pode ser inconsistente. **Com synchronized**, garanta que a saída sempre seja 0.

3. Comunicação entre Threads (Produtor-Consumidor)

Implemente um buffer compartilhado onde um **Produtor** insere números de 1 a 10, e um **Consumidor** lê e remove esses números.

- Use `wait()` e `notify()` para evitar que o **Consumidor** leia um buffer vazio ou o **Produtor** tente inserir quando o buffer estiver cheio.
- O **Produtor** deve esperar caso o buffer esteja cheio e só continuar quando houver espaço disponível.

4. Evitando Deadlocks em Transferências Bancárias

Implemente um sistema bancário com **duas contas** (**ContaA** e **ContaB**), ambas com saldo inicial de 1000.

- Duas threads tentarão transferir 200 de uma conta para a outra ao mesmo tempo.
- **Problema:** Deadlock pode ocorrer se ambas as threads bloquearem os recursos na ordem errada.
- **Solução:** Utilize um **método ordenado de bloqueio** para evitar deadlock.

5. Simulando um Sistema de Pedidos Online

Implemente um sistema onde um **Cliente** faz pedidos e um **Atendente** os processa.

- O **Cliente** deve adicionar **5 pedidos** a uma lista compartilhada.
- O **Atendente** deve processar um pedido por vez e removê-lo da lista.
- Use `wait()` para impedir que o **Atendente** tente processar um pedido quando a lista estiver vazia.
- Use `notify()` para acordar o **Atendente** sempre que um novo pedido for adicionado.