



LISTA DE EXERCÍCIO I

1. Assinale a alternativa que corresponde ao código correto para deixar uma thread dormir por 5 segundos.

A) Thread.sleep(5);

B) thread.wait(5000);

C) thread.sleep(5);

D) Thread.sleep(5000);

2. Analise o código a seguir:

```
TarefaMultiplicacao tarefa = new TarefaMultiplicacao(...);
```

```
Thread threadMultiplicador = new Thread(tarefa);
```

Sobre a tarefa que a thread recebe, selecione a alternativa correta.

A) É um Runnable.

B) É uma classe qualquer.

C) A tarefa deve ter o método main.

D) É um Comparable.

3. Em relação a um programa com várias Threads, marque a alternativa correta sobre a ordem de execução.

A) A ordem de execução será diferente em máquinas diferentes, mas na mesma máquina será sempre a mesma.

B) Não é possível determinar a ordem de execução, que pode ser sempre diferente inclusive na mesma máquina.

C) A ordem de execução será a mesma somente para máquinas com mesmo sistema operacional.

D) A ordem de execução pode ser determinada, mas isso depende da implementação da máquina virtual.



4. Considere que um programa que crie e chame o método `start()` em três threads que respectivamente imprimem os valores 1, 2 e 3. Qual a ordem que esses valores serão impressos?

A) 3,2,1

B) Nada será impresso.

C) 1, 2, 3

D) Não é possível determinar a ordem.

5. Assinale a alternativa que apresenta a maneira correta de pegar a instância da Thread atual (aquela que está sendo executada).

A) Thread atual = `Object.currentThread();`

B) Thread atual = `Thread.currentThread();`

C) Thread atual = `(Thread) this;`

D) Thread atual = `Thread.currentThread();`

6. Qual o nome do bloco ou modificador que deve ser colocado em um método para que não possa ser executado por duas Threads ao mesmo tempo? Selecione a alternativa correta.

A) `static`

B) `strictfp`

C) `synchronized`

D) `final`

7. Assinale a alternativa que apresenta o significado de operação atômica.

A) Cuja execução não pode ser interrompida na metade.

B) Que está associada a apenas uma Thread

C) Cuja execução é feita na memória principal sem o uso de cache.

D) Cuja execução por várias Threads é alternada.



8. assinale a alternativa que faz uso correto da palavra chave synchronized:

A) public void metodo() {
}

B) public void metodo() {
 synchronized(this){
 }
}

C) public class TarefaBuscaNome {
 public synchronized TarefaBuscaNome() {
 }
}

D) public class TarefaBuscaNome {
 public synchronized String nome;
}

9. Assinale a alternativa que representa a maneira correta de fazer com que uma thread A espere a execução da thread B.

A) Coloque um wait() na Thread B.

B) Coloque um wait() na Thread A e um notify() na Thread B.

C) Coloque o modificador synchronized na thread B.

D) Coloque um notify() na Thread A

10. Crie um programa na linguagem Java para ler dois números e qual operação matemática deve ser utilizada. Em seguida o programa deverá apresentar o resultado do cálculo. Toda a operação matemática deverá ser executada via thread.

11. Implemente uma thread para contar a quantidade de consoantes e vogais de uma frase.



12. Escreva um programa para ler um valor X e um valor Z (se Z for menor que X deve ser lido um novo valor para Z). Crie uma thread para contar quantos números inteiros devemos somar em sequência (a partir do X inclusive) para que a soma ultrapasse o valor de Z o mínimo possível. Escrever o valor final da contagem.

Exemplo:

X	Z	Reposta	
3	20	5	(3+4+5+6+7=25)
2	10	4	(2+3+4+5=14)
30	40	2	(30+31=61)

13. Uma agência bancária possui vários clientes, todavia a agência possui apenas um caixa eletrônico em funcionamento, para a realização de saques e transferências. Para realizar a operação de saque o cliente gasta 8 segundos para finalizar seu saque e para a transferência o cliente do banco gasta 5 segundos. Implemente um sistema em que o caixa eletrônico será o nosso objeto e os clientes serão threads que tentaram realizar as operações de saque e transferência no caixa eletrônico.

14. Implemente um mecanismo que verifica se o caixa eletrônico do exercício anterior (13) está sem cédulas para saque, em caso afirmativo, deverá ser inicializada uma thread com o objetivo de alimentar novas cédulas no caixa eletrônico. Durante esse processo o caixa eletrônico ficará indisponível, esperando o terminado de alimentação das cédulas, apenas ao final do processo os clientes podem voltar a utilizar o caixa eletrônico.

15. Implemente um servidor utilizando ServerSocket que irá receber de seus clientes os seguintes comandos:

- **Desenhar:** o servidor deverá retornar um desenho de forma geométrica no console do cliente.

- **Listar:** o servidor deverá listar no cliente todos os nomes em ordem alfabética contidos um arquivo texto (criar um arquivo texto com nomes a escolha);

- **Média:** O servidor deverá listar no cliente o número médio de letras por nome contido no arquivo texto (criado anteriormente);

- **Buscar(&parâmetro):** O servidor deverá listar no cliente todos os nomes que contém a letra passado por parâmetro (&parâmetro). Exemplo: Buscar(p) lista todos os nomes com “p”.

No servidor cada cliente deverá estar conectado em thread e todas as execuções dos comandos do cliente também deverão estar em threads.