

1. Como o uso de várias threads Java permite escrever programas mais eficientes?
R. O uso de várias threads Java permite escrever programas mais eficientes quando tratamos de processos que devem ser assíncronos ou que não necessariamente necessitem ser sequenciais.

2. Na criação de um objeto executável, por que pode ser melhor estender Thread em vez de implementar Runnable?

R. Porque as boas práticas dizem que só devemos estender classes quando necessitamos melhorar ou modificar algo da classe mãe.

3. Mostre como podemos usar join() para esperar um objeto de thread chamado MyThrd terminar.

R. Considere a thread mt1 declarada conforme segue:

```
MyThread mt1 = new MyThread("Child #1");
```

Dentro de try (do bloco try-catch):

```
mt1.thread.join();
```

O uso de join exige um catch para InterruptedException.

4. Mostre como configurar uma thread chamada MyThrd com três níveis acima da prioridade normal.

R.

```
MyThrd myThrd = new MyThrd("Chann #1");  
myThrd.setPriority(Thread.NORM_PRIORITY+3);
```

5. Qual é o efeito da inclusão da palavra-chave synchronized em um método?

R. A palavra-chave synchronized sincroniza duas threads. Considere que são duas threads. Ela executará a primeira, encerra, depois executa a segunda. Ou seja, sequencialmente.