

1. Como o uso de várias threads Java permite escrever programas mais eficientes?

Com o uso do modelo multi-threads, é possível criar aplicações poderosas capazes de solucionar problemas complexos, como por exemplo, se for preciso processar um arquivo muito grande, com várias threads é possível fazer isso de modo síncrono, várias threads processando partes pequenas deste arquivo grande, faz com que o processamento do arquivo seja mais rápido.

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

É fortemente indicado pela comunidade, e, até mesmo no livro "Use a cabeça - Java", que só estendamos uma classe com Thread, caso seja realmente necessário um novo tipo de Thread. Caso contrário, recomenda-se a implementação do Runnable. O motivo é arquitetural. Uma vez que a classe é estendida com Thread, a mesma fica impossibilitada de estender de alguma outra classe, inviabilizando sua evolução pelo processo de herança. Agora, se implementarmos a interface Runnable, a classe fica livre para estender de alguma outra classe, viabilizando sua evolução.

Um ponto de atenção também se dá pelo fato de que, quando estendemos de Thread, cada uma de nossas threads cria um objeto único e associa a ele. E se implementarmos o Runnable, ele compartilhará o mesmo objeto com diversas threads.

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

```
myThrd.join();
```

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

```
myThrd.setPriority(Thread.NORM_PRIORITY + 3);
```

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

A palavra synchronized, faz com que apenas uma thread acesse o método por vez, evitando sobrescrita de dados.