

Soft Blue

Fundamentos de Java

Programação Multithread e Sincronismo

Soft Blue

www.softblue.com.br

Todos os direitos de cópia reservados. Não é permitida a distribuição física ou eletrônica deste material sem a permissão expressa e por escrito do autor.

Tópicos Abordados

- O que são threads
- Criando e executando threads
- Escalonador
- Ciclo de vida das threads
- Prioridade em threads
- Sincronização de threads
- Comunicação entre threads

O Que São Threads

- “Processos Leves”
- Uma thread representa uma linha de execução em um processo
- Um processo pode ter uma ou mais threads executando simultaneamente
- As threads compartilham recursos do processo

Criando Threads

- Uma thread pode ser criada através da extensão da classe **Thread**

```
public class MinhaThread extends Thread {  
    public void run () {  
        //código da thread  
    }  
}
```

```
MinhaThread t = new MinhaThread();  
t.start();
```

Criando Threads

- Outra alternativa é através da implementação da interface **Runnable**

```
public class MinhaThread implements Runnable {  
    public void run () {  
        //código da thread  
    }  
}
```

```
Thread t = new Thread(new MinhaThread());  
t.start();
```

Executando Threads

- Independente da forma como é criada, o código que será executado pela thread deve ser implementado dentro do método **run()**
- Quando o método **run()** termina, a thread também termina

Escalonador

- A JVM possui um escalonador que define como será a execução das threads
- Se o computador possui apenas 1 processador, apenas 1 thread pode rodar ao mesmo tempo

Ciclo de Vida de Uma Thread



Prioridades em Threads

- Threads podem ter prioridades
 - Não existe garantia de que as prioridades serão seguidas
- Constantes
 - `MIN_PRIORITY` (1)
 - `NORM_PRIORITY` (5)
 - `MAX_PRIORITY` (10)
- O método `yield()` pode ajudar outras threads de mesma prioridade a executar

Sincronização de Threads

- Muitas vezes é necessário que várias threads acessem o mesmo objeto
 - Race Conditions
- Sincronizar as threads é necessário para evitar acesso simultâneo às regiões críticas (critical sections)

Sincronização de Threads

- A sincronização de threads em Java é feita através do uso de monitores
- Monitor é um objeto Java qualquer que cuida de uma região crítica
- Só é permitida a execução de uma thread por vez
- As outras threads ficam aguardando

Usando o Synchronized

- É possível sincronizar o método todo (método **synchronized**)

```
public synchronized void metodo() {  
    //código sincronizado  
}
```

- É possível sincronizar apenas um bloco (bloco **synchronized**)

```
synchronized(monitor) {  
    //código sincronizado  
}
```

Thread-Safe

- Um recurso é thread-safe quando existe a garantia de que ele funcionará adequadamente na presença de várias threads
 - Algumas coleções antigas do Java eram thread-safe
 - Variáveis locais são sempre thread-safe

Comunicação Entre Threads

- Às vezes a atividade de uma thread depende da atividade de outra
 - Comunicação é necessária
- Métodos

Método	Descrição
wait()	Faz a thread esperar até que outra thread a notifique ou que determinado tempo tenha passado
notify()	Notifica uma <i>thread</i> que está aguardando
notifyAll()	Notifica todas as <i>threads</i> que estão aguardando (uma delas acessa a região crítica e as outras voltam a esperar)

Problemas de Sincronização

- Starvation
 - Uma thread nunca consegue executar
- Deadlock
 - Duas ou mais threads estão paradas aguardando por algo que nunca vai acontecer
- Travamento do sistema

Problemas Clássicos

- “Produtor e Consumidor”
- “Leitor e Escritor”
- “Jantar dos Filósofos”
- “Barbeiro Adormecido”

Considerações Finais

- Quando o assunto é thread, muito pouco é garantido
- Cuidado com o sincronismo dos dados
- Encontrar problemas de sincronismo é bastante difícil
- Quando for programar multithread, tome bastante cuidado e saiba o que você está fazendo

Colocando em Prática...



Agora que você já aprendeu a teoria, acesse as vídeo-aulas práticas e pratique os assuntos abordados neste módulo!

[Clique aqui para acessar as vídeo-aulas práticas](#)
