

Programação Concorrente em Java

Aula 08

Blocos Sincronizados

Índice

- A palavra reservada **synchronized**
- Métodos de instância sincronizados
- Métodos estáticos sincronizados
- Blocos sincronizados em métodos de instância
- Blocos sincronizados em métodos estáticos

Synchronized

- Blocos sincronizados em Java são marcados com a palavra `synchronized`.
- Um bloco sincronizado em Java é relativo a algum objeto.
- Todo bloco sincronizado só pode ter uma thread executando em seu interior por vez.
- Todas as threads que tentarem executar em seu interior ficarão bloqueadas até a thread dentro do bloco sincronizado sair.

Synchronized

- A palavra reservada **synchronized** pode ser usada em 4 tipos diferentes de estruturas:
 - Métodos de instância
 - Métodos estáticos
 - Blocos de código dentro de métodos de instância
 - Blocos de código dentro de métodos estáticos
- Estes blocos são sincronizados em diferentes objetos. O tipo de bloco que você irá precisar irá depender da situação.

Métodos de instância sincronizados

- Exemplo de método de instância sincronizado:

```
public synchronized void add(int value){  
    this.count += value;  
}
```

- Um método de instância sincronizado é sincronizado na instância dona do método.
- Cada instância, então, tem seu próprio método sincronizado.
- Apenas uma única thread pode ter acesso ao método sincronizado.
- Uma thread por instância.

Métodos sincronizados estáticos

- Assim como os métodos de instância:

```
public static synchronized void add(int value){  
    count += value;  
}
```

- Métodos estáticos sincronizados são sincronizados em relação à **classe** ao qual pertencem.
- Como apenas um único objeto estático existe por classe, então apenas uma única thread pode entrar por vez no método estático.

Blocos sincronizados em métodos de instância

- As vezes, não é necessário sincronizar um método inteiro.
- Em Java, é possível sincronizar apenas uma pedaço do método (bloco de código).
- Blocos sincronizados dentro de métodos fazem esse trabalho:

```
public void add(int value){  
  
    //some code..  
  
    synchronized(this){  
        this.count += value;  
    }  
  
    //some more code  
}
```

Blocos sincronizados em métodos de instância

- Esse bloco irá executar quase exatamente como um método sincronizado de instância.
- O bloco recebe um parâmetro “this” que se refere exatamente a instância da classe que implementa o método add.
- Sendo assim, esse bloco é dito **sincronizado no monitor do objeto**.
- Um método sincronizado de instância usa um objeto o qual pertence como um **objeto monitor**.
- Apenas uma única thread pode executar dentro do bloco sincronizado no mesmo objeto monitor.

Blocos sincronizados em métodos de instância

- Métodos equivalentes em termos de sincronização:

```
public class MyClass {  
    public synchronized void log1(String msg1, String msg2){  
        log.writeln(msg1);  
        log.writeln(msg2);  
    }  
  
    public void log2(String msg1, String msg2){  
        synchronized(this){  
            log.writeln(msg1);  
            log.writeln(msg2);  
        }  
    }  
}
```

O que acontece nesse caso?

Blocos sincronizados em métodos estáticos

- Mesmo exemplo em métodos estáticos (pertencem a classe)

```
public class MyClass {  
    public static synchronized void log1(String msg1, String msg2){  
        log.writeln(msg1);  
        log.writeln(msg2);  
    }  
  
    public static void log2(String msg1, String msg2){  
        synchronized(MyClass.class){  
            log.writeln(msg1);  
            log.writeln(msg2);  
        }  
    }  
}
```

O que acontece nesse caso?

Exemplo

```
public class Counter{  
    long count = 0;  
  
    public synchronized void add(long value){  
        this.count += value;  
    }  
}
```

```
public class CounterThread extends Thread{  
    protected Counter counter = null;  
    public CounterThread(Counter counter){  
        this.counter = counter;  
    }  
    public void run() {  
        for(int i=0; i<10; i++){  
            counter.add(i);  
        }  
    }  
}
```

```
public class Example1 {  
    public static void main(String[] args){  
        Counter counter = new Counter();  
        Thread threadA = new CounterThread(counter);  
        Thread threadB = new CounterThread(counter);  
        threadA.start();  
        threadB.start();  
    }  
}
```

```
public class Example2 {  
    public static void main(String[] args){  
        Counter counterA = new Counter();  
        Counter counterB = new Counter();  
        Thread threadA = new CounterThread(counterA);  
        Thread threadB = new CounterThread(counterB);  
        threadA.start();  
        threadB.start();  
    }  
}
```

Referências

- <http://tutorials.jenkov.com/java-concurrency/synchronized.html>

















