

Sistemas Operacionais

Threads em Java

André Luis Martinotto

Threads em Java

- Todos os programas Java consistem em pelo menos um fluxo de execução
 - Java fornece suporte no nível de linguagem de programação para a criação e gerência de threads.
- Existem duas maneiras de se implementar uma thread em Java:
 - Usando **Herança**
 - Usando **Interfaces**
- As threads no java são disparadas, executadas e controladas pela Máquina Virtual Java.

Classe Threads

- Pertence ao pacote `java.lang.Thread`.
- Execução dos seguintes passos :
 - Criar uma nova classe (subclasse de `Thread`)
 - Sobreescrever o método `run()`
 - Criar um objeto da nova subclasse
 - Chamar o método `start()`
- A thread é criada no momento em que se chama o método `start()`

Exemplo 1 - MinhaThread.java

```
public class MinhaThread extends Thread {  
    private int espera;  
  
    public MinhaThread(String nome, int espera) {  
        this.setName(nome);  
        this.espera = espera;  
    }  
  
    public void run(){  
        try{  
            sleep(espera);  
        }catch (InterruptedException e){};  
        System.out.println("Sou a thread " + this.getName());  
    }  
}
```

- **sleep(int x)** faz a thread dormir por x milisegundos

Exemplo 1 - MinhaThread.java

```
public static void main (String [] args){  
    MinhaThread t1 = new MinhaThread("Thread 1",500);  
    MinhaThread t2 = new MinhaThread("Thread 2",250);  
    MinhaThread t3 = new MinhaThread("Thread 3",50);  
    t1.start();  
    t2.start();  
    t3.start();  
    try{  
        t1.join();  
        t2.join();  
        t3.join();  
    }catch(InterruptedException e){}  
}
```

- **join():** faz com que o programa espere que a thread termine a sua execução

Exemplo 2 - MinhaThread.java

```
public class MinhaThread extends Thread {  
    private int intervalo;  
  
    public MinhaThread(int intervalo) {  
        this.intervalo = intervalo;  
    }  
  
    public void run(){  
        for(int i=0; i<50; i++){  
            System.out.println(this.getName() + ":" + i);  
            try{  
                sleep( (long) (Math.random() * intervalo));  
            }catch(Exception e){}  
        }  
    }  
}
```

Exemplo 2 - MinhaThread.java

```
public static void main (String [] args){  
    MinhaThread t1 = new MinhaThread(1000);  
    MinhaThread t2 = new MinhaThread(1000);  
    t1.start();  
    t2.start();  
}  
}
```

Interface Runnable

- Permite que threads sejam utilizadas com classes que já herdam propriedades de uma superclasse
 - Lembre-se que no java não existe herança múltipla
- Consiste na execução dos seguintes passos :
 - Deve-se definir uma classe que implementa a interface Runnable
 - Deve-se criar um objeto desta classe e após criar um objeto da classe Thread passando o objeto como parâmetro
 - É necessário redefinir o método run()

Exemplo 3 - MinhaThread.java

```
public class MinhaThread implements Runnable {  
    private int intervalo;  
  
    public MinhaThread(int intervalo) {  
        this.intervalo = intervalo;  
    }  
  
    public void run(){  
        for(int i=0; i<50; i++){  
            System.out.println(Thread.currentThread().getName() + ":" + i);  
            try{  
                Thread.currentThread().sleep((long)(Math.random()*  
                    intervalo));  
            }catch(Exception e){}  
        }  
    }  
}
```

Exemplo 3 - MinhaThread.java

```
public static void main (String [] args){  
    MinhaThread mt1 = new MinhaThread(1000);  
    Thread t1 = new Thread(mt1);  
    MinhaThread mt2 = new MinhaThread(1000);  
    Thread t2 = new Thread(mt2);  
    t1.start();  
    t2.start();  
}  
  
}
```

Sincronizando Threads

- **Metodos synchronized**
 - Previne que mais de uma thread acesse esse método ao mesmo tempo
 - Se mais de um método no programa é synchronized nenhum desses métodos será chamado ao mesmo tempo
 - Os métodos synchronized são ações atômicas
 - Métodos synchronized fornecem o que se chama na área de sistemas operacionais **exclusão mútua!**

Exemplo 4 - MinhaThread.java

```
public class MinhaThread extends Thread {  
  
    public MinhaThread(String nome) {  
        this.setName(nome);  
    }  
  
    public static synchronized void escreveNome(){  
        for (int i=0; i<5; i++){  
            try{  
                sleep(1000);  
            }catch (InterruptedException e){ };  
            System.out.println("Sou a thread " + Thread.currentThread().getName());  
        }  
    }  
}
```

Exemplo 4 - MinhaThread.java

```
public void run(){
    MinhaThread.escreveNome();
}

public static void main (String [] args){
    MinhaThread t1 = new MinhaThread("Thread 1");
    MinhaThread t2 = new MinhaThread("Thread 2");
    MinhaThread t3 = new MinhaThread("Thread 3");
    t1.start();
    t2.start();
    t3.start();
    try{
        t1.join();
        t2.join();
        t3.join();
    }catch(InterruptedException e){ }
}
```

Exemplo 5 - Pilha.java

```
public class Pilha {  
    private int [] pilha;  
    private int indice;  
    private int tam;  
  
    public Pilha(int tam) {  
        this.indice = 0;  
        this.tam = tam;  
        this.pilha = new int[tam];  
    }  
}
```

Exemplo 5 - Pilha.java

```
public synchronized void push(int num) throws Exception {  
    if ( indice == tam-1) throw new Exception("Fila cheia!");  
    System.out.println("Produzindo: " + num );  
    pilha[indice] = num;  
    indice++;  
}
```

```
public synchronized void pop() throws Exception {  
    if (indice == 0 ) throw new Exception("Fila vazia!");  
    indice--;  
    System.out.println("Consumindo: " + pilha[indice] );  
}
```

```
}
```

Exemplo 5 - Produtor .java

```
public class Produtor extends Thread {  
    Pilha pilha;  
    public Produtor(Pilha pilha){  
        this.pilha = pilha;  
    }  
  
    public void run() {  
        while( true ){  
            try {  
                this.sleep((long)(Math.random() * 5000));  
                pilha.push((int)(Math.random()*1000));  
            } catch (Exception e) {System.out.println(e);}   
        }  
    }  
}
```


Exemplo 5 - Consumidor .java

```
public class Consumidor extends Thread {  
    Pilha pilha;  
    public Consumidor(Pilha pilha){  
        this.pilha = pilha;  
    }  
  
    public void run(){  
        while ( true ){  
            try {  
                this.sleep((long)(Math.random() * 5000));  
                pilha.pop();  
            } catch (Exception e) {System.out.println(e);}   
        }  
    }  
}
```

Exemplo 5 - Main.java

```
public class Main {  
  
    public static void main ( String [] args ){  
        Pilha pilha = new Pilha(40);  
        Produtor produtor = new Produtor(pilha);  
        Consumidor consumidor = new Consumidor(pilha);  
        produtor.start();  
        consumidor.start();  
    }  
}
```