Programação Concorrente

Threads em Java

André Luis Martinotto

UCS UNIVERSIDADE DE CAXIAS DO SUL

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

- Utiliza-se Threads em Java através da classe Thread.
  - 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 e criada no momento em que se chama o metodo 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

Interface Runnable

- Java não existe herança múltipla.
  - A **Interface Runnable** permite que threads sejam utilizadas com classes que já herdam propriedades de uma superclasse.

Interface Runnable

- O uso de Threads 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 2 - MinhaThread.java

```
public class MinhaThread  implements Runnable {
    private int espera;

    public MinhaThread(String nome, int espera) {
        Thread.currentThread().setName(nome);
        this.espera = espera;
    }

    public void run(){
       try{
         Thread.currentThread().sleep(espera);
         System.out.println("Sou a Thread " + Thread.currentThread().getName());
       }
       catch (InterruptedException e){};
    }
```

Exemplo 2 - MinhaThread.java

```
public static void main (String [] args){
        MinhaThread m1 = new MinhaThread("Thread 1",500);
        MinhaThread m2 = new MinhaThread("Thread 2",250);
        MinhaThread m3 = new MinhaThread("Thread 3",50);
        Thread t1 = new Thread(m1);
        Thread t2 = new Thread(m2);
        Thread t3 = new Thread(m2);
        t1.start();
        t2.start();
        t3.start();
        try{
              t1.join();
              t2.join();
              t3.join();
        }catch(InterruptedException e){}
    }
}
```

Sincronizando Threads

- **Metodos synchronized**
  - Não permite que mais de uma thread acesse um método ao mesmo tempo
  - Os métodos synchronized são ações atômicas
  - Métodos synchronized fornecem o que se chama na área de sistemas operacionais **exclusão mútua!**

Exemplo 3 - 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 3 - 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){}
}
}
```