

Programação Concorrente em Java

Alcides Calsavara

Threads e Semáforos

Programação Concorrente em Java

Classe Semaphore

Pacote: java.util.concurrent Instanciação: Semaphore s = new Semaphore(n); // n é inteiroOperações básicas: s.acquire(); // s.esperar() s.release(); // s.sinalizar() OBS: É obrigatório fazer tratamento de exceção (try-catch).

♦ Arquivos: Teste01.java e Teste02.java

Classe Thread

♦ Pacote:

java.lang

✦ Relação de herança com threads de aplicação:

class Cliente extends Thread

♦ Método de execução inicial (herdado de Thread):

```
public void run()
```

♦ Criação e escalonamento:

```
Cliente c = new Cliente();
c.start();
```

♦ Arquivos: Tarefa.java, Teste03.java e Teste04.java

Miscelânea

♦ Fazer uma thread dormir por t milissegundos:

```
Thread.sleep(t);
```

- *♦ Gerar um inteiro qualquer entre 0 e n-1:*

 - pacote: java.util
 - ♦ operação nextInt(n): retorna um inteiro entre 0 e n-1
- ♦ Arquivo: Aleatorio.java

Miscelânea

♦ Número de processadores (cores)

```
// get the runtime object
Runtime runtime = Runtime.getRuntime();
// get the number of processors
int numberOfProcessors = runtime.availableProcessors();
```

♦ Arquivo: NumeroProcessadores.java

Miscelânea

♦ Tempo do Sistema

long t1 = System.currentTimeMillis();

♦ Arquivo: TempoSistema.java