

Threads

Threads são fluxos de execução que rodam dentro de um processo (aplicação). Normalmente os threads compartilham regiões de memória, mas não necessariamente. Processos permitem que o seu SO execute mais de uma aplicação ao mesmo tempo, enquanto que threads permitem que sua aplicação execute mais de um método ao mesmo tempo (programas multithreads).

Processos → Fork-join (Processo filho → cópia completa do pai)

Thread ↔ processo leve (não é cópia completa do pai)

Criando Thread usando herança ou interfaces

Herança

Usando herança, sua classe já é uma Thread, que quando tiver seu método start() chamado, vai executar tudo que estiver no método run().

```
public class TesteThreads extends Thread{

    public TesteThreads(String nome){
        super(nome);    // chama o método construtor da classe pai
        (no caso a classe Thread)
    }

    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(getName()+" na etapa"+i);
            try{
                sleep((int) (Math.random()*2000));
            }catch(Exception e){}
        }
        System.out.println("Thread terminada: "+getName());
    }
}

public class CorridaTreads {
    public static void main (String args[]){
        TesteThreads a,b;
        a = new TesteThreads("A1");
        a.start();
        b = new TesteThreads("B1");
        b.start();
        try{a.join();}catch(Exception e){}
        try{b.join();}catch(Exception e){}
    }
}
```

Interfaces

Usando interfaces, define-se uma classe cujo método run() vai ser executado por uma Thread. Para isto, deve-se dizer que implementa a interface Runnable, que é composta do método Run(). Quando uma Thread é construída usando como argumento uma classe que implementa a interface Runnable, chamar o método start para esta Thread faz com que o método run() da classe

comece a ser executado nesta thread.

```
public class TesteThreads implements Runnable{
    public String str;
    public TesteThreads(String nome){
        str = nome;
    }

    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(str +" na etapa "+i);
            try{
                Thread.sleep((int) (Math.random()*2000));
            }catch(Exception e){}
        }
        System.out.println("Thread terminada: "+str);
    }
}

public class CorridaThreads {
    public static void main (String args[]){
        Thread t1,t2;
        TesteThreads a,b;
        a = new TesteThreads("A1");
        t1 = new Thread(a);
        t1.start();
        b = new TesteThreads("B1");
        t2 = new Thread(b);
        t2.start();
        try{t1.join();}catch(Exception e){}
        try{t2.join();}catch(Exception e){}
    }
}
```