



Paradigma da Programação

Autoria: Prof. Paulo Cabrita

Objectivos:

No final da aula o aluno deverá saber:

- O que é uma Thread.
- As duas formas de criar Threads: a classe derivada da Thread e implementação da interface Runnable
- A função dos métodos run, start e sleep.
- O que são User threads e Daemon threads.
- Identificar o thread que executa o código
- Utilizar as classes Timer e TimerTask.

Plano da aula:

- Apresentar e explicar o exemplo seguinte:

```
public class Funcionario {
    String nome;
    int delay;
    int pecasProduzidas = 0;
    static int TOTAL_PECAS = 50;

    public Funcionario(String nome, int
delay) {
        this.nome = nome;
        this.delay = delay;
    }

    public void trabalhar () {
        System.out.println(nome + " -
Comecei.");
        while (pecasProduzidas < TOTAL_PECAS) {
            try {
                Thread.sleep(delay);
                System.out.println(nome + ": " +
++pecasProduzidas);
            } catch (InterruptedException ex) {
                System.out.println(ex);
            }
        }
        System.out.println(nome + " -
Terminei.");
    }

    /** main em que não existe concorrência.
    */
    public static void main(String[] args)
    {
        Funcionario manuel = new
Funcionario("Manuel", 3000);
        manuel.trabalhar ();
        Funcionario pedro = new
Funcionario("\tPedro", 5000);
        pedro.trabalhar ();
        System.out.println("Main
terminado.");
    }
}
```

Exercícios:

1. Crie uma aplicação que simula a produção de peças por dois funcionários. Utilize `Thread.sleep` para simular o tempo de demora. O Manuel produz uma peça a cada 3 segundos e o Pedro uma peça a cada 5. Porque não funciona bem? Transforme agora a classe `Funcionário` num `Thread`.
2. Implemente agora o mesmo exercício utilizando objectos `Runnable`.
3. Implemente também utilizando as classes `Timer` e `TimerTask`.
4. Criar agora uma classe que faça a avaliação de desempenho dos funcionários colocando-os a trabalhar durante uma hora e avaliando as peças produzidas no final. Utilize para isso `threads Daemon`.

Exercícios suplementares

5. Criar uma aplicação que faz três contagens decrescentes começando em valores passados como argumento. A aplicação deve esperar que os contadores terminem e imprimir uma mensagem quando todos os contadores tiverem terminado.