

# Programação orientada a objeto II

# ROTEIRO DE AULA PRÁTICA

**NOME DA DISCIPLINA:** Programação orientada a objeto II.

## Unidade 2

### Seção 2.1

#### OBJETIVOS

##### Definição dos objetivos da aula prática:

Aplicar as técnicas de desenvolvimento concorrente para construir um aplicativo orientado a objetos com a linguagem Java

#### INFRAESTRUTURA

##### Instalações:

Laboratório de informática

##### Materiais de consumo:

Descrição	Quantidade de materiais por procedimento/atividade
Computador	Máximo dois alunos por computador

##### Software:

Sim (X) Não ( )

Em caso afirmativo, qual? Eclipse OXYGEN .2 e Java SDK 9.

Pago ( ) Não Pago (X)

Tipo de Licença: OpenSource Eclipse Public License e Oracle Binary Code License.

##### Descrição do software:

O programa Eclipse OXYGEN .2 deve ser instalado com a opção: "Eclipse IDE for Java Developers" O java SDK deve ser instalado de forma padrão

<u>Equipamento de Proteção Individual (EPI):</u>
Não se aplica

## PROCEDIMENTOS PRÁTICOS

Neste momento você deve ajudar o professor por meio da descrição de todas as etapas que deverão ser realizadas para a execução dos procedimentos práticos. Considerando a carga horária da aula prática, você pode replicar a caixa de procedimento/atividade quantas vezes for necessário.

### Procedimento/Atividade n.1

#### Atividade proposta:

A tarefa consiste em ler diversos arquivos de texto de um diretório utilizando thread para que seja possível tornar o processo mais rápido. O arquivos contém os dados das multas geradas no dia com o formato placa do carro;valor da multa, como exemplo:

AAA1234;250,00

BBBB1234;100,00

#### Procedimentos para a realização da atividade:

Crie uma classe que possua um thread para fazer o processamento:

1. Criar uma classe que implementa a interface Runnable:
  - **public class** ProcessadorMultas **implements** Runnable
2. Declarar como atributos a representação da thread e o caminho dos arquivos:
  - **private** Thread th;
  - **private** String caminhoArquivos;
3. No construtor instanciar a classe thread indicando a própria classe como método da thread:
  - th = **new** Thread(**this**);
4. No método "run()" fazer a leitura dos arquivos do diretório e exibir na terminal os dados:
  - **public void** run()

Caso tenha dúvidas quanto a implementação verifique uma solução no Quadro 1.

Quadro 1: Resolução do exercício de sobre thread.

```
package U2S1;  
  
import java.io.File;  
import java.io.FileNotFoundException;
```

```

import java.io.FileReader;
import java.util.Scanner;

public class ProcessadorMultas implements Runnable{

    private Thread th;
    private String caminhosArquivos;
    private int qtdFiles;
    private int arquivoAtual;
    public ProcessadorMultas(String pCaminhosArquivos)
    {
        th = new Thread(this);
        caminhosArquivos = pCaminhosArquivos;
    }

    public void iniciar()
    {
        th.start();
    }

    public void run() {

        File diretorio = new File(caminhosArquivos);
        File[] listaDeArquivos = diretorio.listFiles();
        qtdFiles = listaDeArquivos.length;
        for (int i = 0; i < listaDeArquivos.length; i++) {
            arquivoAtual = i;
            if (listaDeArquivos[i].isFile()) {
                String nomeArquivos = listaDeArquivos[i].getName();
                processaMulta(caminhosArquivos+nomeArquivos);
            }
        }

    }

    public void processaMulta(String nomeArquivo)
    {
        Scanner in;
        try {
            in = new Scanner(new FileReader(nomeArquivo));
            while(in.hasNext())
            {
                String[] dados = in.next().split(";");
                String placa = dados[0];
                String valor = dados[1];
                System.out.println("Placa do carro " + placa + " Valor da
multa "+ valor);
            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

```

```
public static void main(String[] args) {  
  
    ProcessadorMultas pa1 = new ProcessadorMultas("c:\\multas\\");  
    pa1.inciar();  
}  
}
```

Fonte: elaborado pelo autor.

#### Checklist:

Para verificar se a tarefa foi concluída com sucesso, os seguintes itens devem ser contemplados:

1. Existe uma classe que implementa interface Runnable com o método "run()";
2. Foi inicializada a thread e o processo ocorre de maneira concorrente;
3. O aluno criou uma pasta chamada multas na raiz c:\;
4. O aluno criou arquivos dentro desse diretório com o formato especificado;
5. O programa está executando sem erros;
6. Os métodos possuem alta coesão.

## RESULTADOS

#### Resultados da aula prática:

Ao final da aula o aluno deverá entregar um projeto em Java que faça o processamento de arquivos de texto usando processamento paralelo. Ao conseguir completar a tarefa o aluno deve ser capaz de criar threads e fazer processamento paralelos.