

Programação orientada a objetos II.

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: Programação orientada a objetos II

Unidade 2

Seção 2.2

OBJETIVOS

Definição dos objetivos da aula prática:

Desenvolver o conhecimento da utilização do sistema de controle de exceção do Java.

INFRAESTRUTURA

Instalações:

Laboratório de informática

Materiais de consumo:

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

Software:

Sim (☒) Não (☐)

Em caso afirmativo, qual? Eclipse OXYGEN .2 e Java SDK 10.

Pago (☐) Não Pago (☒)

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

Equipamento de Proteção Individual (EPI):

Não aplicável

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:

Fazer o processamento de arquivos XML de notas fiscais. O procedimento consiste em abrir o arquivo e exibi-lo no terminal.

Procedimentos para a realização da atividade:

Implementar o seguinte código da classe ProcessadorXML.

Quadro 1: Classe para processar XML.

```
package U2S2;

import java.io.File;
import java.util.Scanner;

public class ProcessadorXML {

    private String caminhoArquivo;

    public ProcessadorXML(String pCaminhoArquivo)
    {
        caminhoArquivo = pCaminhoArquivo;
    }

    public void lerArquivo()
    {
        File f = new File(caminhoArquivo);
        Scanner sc = new Scanner(f);
        String dados = sc.next();
        System.out.println(dados);
        sc.close();
    }

    public static void main(String[] args) {
        ProcessadorXML le = new ProcessadorXML("nf/1.xml");
        le.lerArquivo();
    }
}
```

Fonte: elaborado pelo Autor.

Alterar o código do Quadro 1 para implementar o tratamento de exceções com o try e o catch.

- Os métodos ou construtores devem ter os blocos **"try"** e **"catch"** com a função que envia para saída padrão a sequência de erros que foram gerados:

```
try {  
    //conteúdo do método ou construtor  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

- Para fechar os recursos se deve utilizar o bloco **"finally"**:

```
try {  
  
} catch (Exception e) {  
    e.printStackTrace();  
} finally {  
    if(sc != null)  
        sc.close();  
}
```

Como resultado todos os métodos possuem tratamento de exceção e o fechamento de recursos é realizado no bloco **"finally"**, como no Quadro 2.

Quadro 2: Tratamento com o **"finally"**.

```
package U2S2;  
  
import java.io.File;  
import java.util.Scanner;  
  
public class ProcessadorXML {  
  
    private String caminhoArquivo;  
  
    public ProcessadorXML(String pCaminhoArquivo)  
    {  
        try {  
            caminhoArquivo = pCaminhoArquivo;  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
    public void lerArquivo()  
    {  
        Scanner sc = null;  
        try {  
            File f = new File(caminhoArquivo);
```

```

        sc = new Scanner(f);
        String dados = sc.next();
        System.out.println(dados);
        sc.close();
    }catch (Exception e) {
        e.printStackTrace();
    }finally {
        if(sc != null)
            sc.close();
    }
}

public static void main(String[] args) {
    try {
        ProcessadorXML le = new ProcessadorXML("nf/1.xml");
        le.lerArquivo();
    }catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Fonte: Elaborado pelo Autor.

Checklist:

Para averiguar se a tarefa foi feita de forma completa:

- Verifique se o nome da classe está correto: ProcessadorXML.
- Verifique se os métodos ProcessadorXML() e lerArquivo() foram criados.
- Verifique se os métodos possuem o tratamento de exceção, com try, catch.
- Os recursos utilizados durante o código são fechados no bloco finally.

RESULTADOS

Resultados da aula prática:

Ao final da aula o aluno deverá entregar um projeto com tratamento de exceções. Nessa entrega deverá ser verificado se ele foi capaz de utilizar os recursos de tratamento de exceção gerando um código que informa quando foi gerado algum problema.