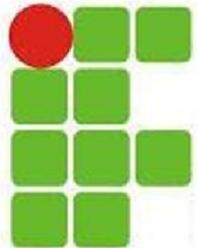
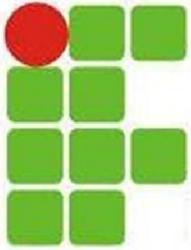

Programação Orientada a Objetos

Professor: Diego Oliveira



Conteúdo 17:
Persistência em Arquivo

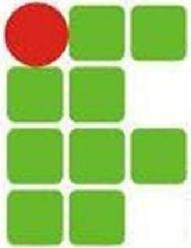




Pacote java.io

- Quando trabalhamos com Leitura e Escrita em disco, estamos trabalhando com Input/Output, comumente chamado de I/O
- O Java tem um pacote específico para armazenar as classes que trabalham com I/O, é o **java.io**
- As classes abstratas **InputStream** e **OutputStream** definem o comportamento dos fluxos de entrada e saída



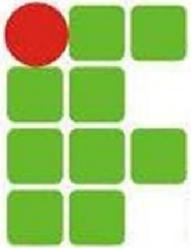


Classe File

- Dentro do pacote java.io, temos a classe File, que representa um arquivo no disco
- Vejamos alguns de seus métodos:

```
File arquivo = new File("meuArquivo.txt");
System.out.println("Existe? " + arquivo.exists());
System.out.println("Pode ler? " + arquivo.canRead());
System.out.println("Pode escrever? " + arquivo.canWrite());
System.out.println("Caminho completo? " + arquivo.getAbsolutePath());
System.out.println("É arquivo? " + arquivo.isFile());
System.out.println("É pasta? " + arquivo.isDirectory());
System.out.println("É oculto? " + arquivo.isHidden());
System.out.println("Tamanho (bytes)? " + arquivo.length());
```





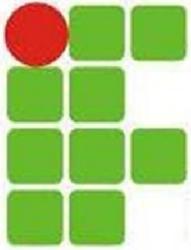
Classe File

- Saída do código anterior:

```
Output - POO-Exercicio17 (run) ✖
Existe? true
Pode ler? true
Pode escrever? true
Caminho completo? C:\Java\PROJETOS_NETBEANS\POO-Exercicio17\meuArquivo.txt
É arquivo? true
É pasta? false
É oculto? false
Tamanho (bytes)? 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Existem alguns outros métodos interessantes, utilize o CTRL+SPACE na IDE para descobrir alguns



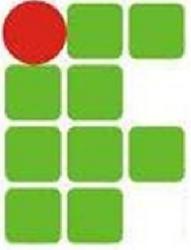


Escrita em Arquivo

- Método de escrita de um texto em um arquivo 'txt':

```
public class WriteFile {  
    public static void escreva(String caminho, String texto)  
        throws IOException {  
        File arquivo = new File(caminho);  
        PrintWriter pw = new PrintWriter(arquivo);  
        pw.write(texto);  
        pw.close();  
    }  
}
```



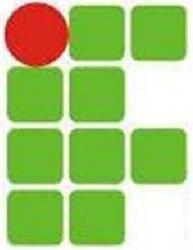


Leitura em Arquivo

- Método de leitura de um texto salvo em um arquivo 'txt':

```
public class ReadFile {  
    public static String leia(String caminho) throws IOException {  
        String texto = "";  
        String str = "";  
        File arquivo = new File(caminho);  
        BufferedReader in = new BufferedReader(new FileReader(arquivo));  
        while ((str = in.readLine()) != null) {  
            texto += str + "\n";  
        }  
        in.close();  
        return texto; ←  
    }  
}
```

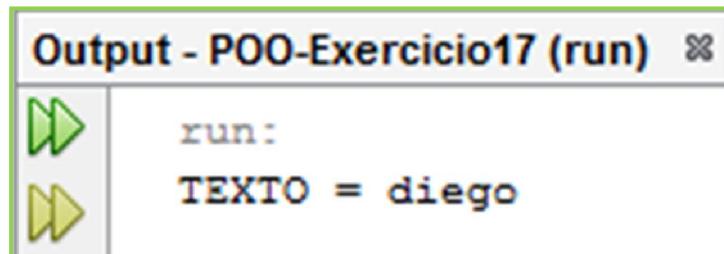
Salve o retorno da chamada deste método em uma STRING



Testando os Métodos

- Testando os métodos de Leitura e Escrita em arquivo através de uma classe de teste:

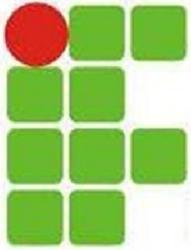
```
public class TesteArquivo {  
    public static void main(String[] args) throws IOException {  
        String texto = "diego";  
        WriteFile.escreva("meuArquivo.txt", texto);  
        String textoLido = ReadFile.leia("meuArquivo.txt");  
        System.out.println("TEXTO = " + textoLido);  
    }  
}
```



Output - POO-Exercicio17 (run) ✖

```
run:  
TEXTO = diego
```





Arquivo Criado

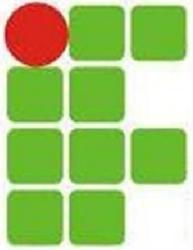
- Arquivo criado com o texto passado:

The image shows a Windows file explorer window and a text editor window. The file explorer window displays the contents of the directory 'C:\Java\PROJETOS_NETBEANS\POO-Exercicio17'. The files listed are:

Name	Date modified	Type
build	23/02/2015 08:17	File folder
nbproject	18/02/2015 19:28	File folder
src	23/02/2015 08:16	File folder
build.xml	18/02/2015 19:28	XML Document
manifest.mf	18/02/2015 19:28	MF File
meuArquivo.txt	23/02/2015 08:19	TXT File

The text editor window shows the content of 'meuArquivo.txt' as 'diego' on line 1, column 1. The status bar indicates 'Ln:1 Col:1 Sel:0|0' and 'Dos\Windows ANSI INS'.



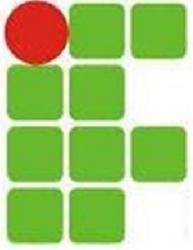


Exercício

- No exercício de cadastro de alunos com GUI adicione os métodos de leitura e escrita em arquivo para salvar permanentemente as informações dos alunos
- Para isso utilize os métodos aprendidos na aula de Strings para separar os alunos lidos do texto por um símbolo (#), depois inicialize novos objetos alunos com os dados lidos separados por vírgulas (nome, idade, mat)



• **VALE 1,0 PONTO EXTRA**



Perguntas?

