

Programação Orientada a Objetos

Prof. Dr. Anderson V. de Araujo



Módulo IV - Conceitos Avançados

Unidade IV - Trabalhando com Arquivos e fluxos



Manipulação de Arquivos em Java

- Java oferece várias classes e métodos para lidar com operações de arquivo;
- Arquivos de texto:
 - Classes `FileReader` e `FileWriter` para ler e escrever texto.
- Arquivos Binários:
 - Classes `FileInputStream` e `FileOutputStream` para ler e escrever dados binários.
- Uso do bloco try-catch para lidar com exceções.

Leitura de Arquivos em Java

- `FileReader`: permite ler dados de um arquivo de texto;
- `FileInputStream`: permite ler dados de um arquivo de arquivos binários;

```
public class LeituraArquivo {  
    public static void main(String[] args) {  
        try {  
            FileReader reader = new FileReader("arquivo.txt");  
            int character;  
  
            while ((character = reader.read()) != -1) {  
                System.out.print((char) character);  
            }  
  
            reader.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Escrita em Arquivos em Java

- `FileWriter`: permite escrever dados em um arquivo de texto;
- `FileOutputStream`: permite escrever dados de um arquivo de arquivos binários;

```
public class EscritaArquivo {  
    public static void main(String[] args) {  
        try {  
            FileWriter writer = new FileWriter("arquivo.txt");  
            writer.write("Olá, mundo!");  
            // escrever mais texto  
            writer.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
public class ManipulacaoArquivosBinarios {  
    public static void main(String[] args) {  
        try {  
            FileInputStream input = new FileInputStream("imagem.jpg");  
            FileOutputStream output = new FileOutputStream("copia_imagem.jpg");  
  
            byte[] buffer = new byte[1024];  
            int bytesRead;  
            while ((bytesRead = input.read(buffer)) != -1) {  
                output.write(buffer, 0, bytesRead);  
            }  
  
            input.close();  
            output.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```


Manipulação de Arquivos e Diretórios

- `File`: fornece métodos para manipular arquivos e diretórios;
 - Criar um diretório, verificar se um arquivo existe, listar arquivos em um diretório;
 - Uso dos métodos `isDirectory()`, `exists()`, `createNewFile()`, `listFiles()`, etc.

```
public class ManipulacaoArquivos {  
    public static void main(String[] args) {  
        // Criar um diretório  
        File directory = new File("meuDiretorio");  
        directory.mkdir();  
        // Verificar se um arquivo existe  
        File file = new File("meuDiretorio/arquivo.txt");  
        if (file.exists()) {  
            System.out.println("O arquivo existe!");  
        }  
        // Listar arquivos em um diretório  
        File[] files = directory.listFiles();  
        for (File f : files) {  
            System.out.println(f.getName());  
        }  
    }  
}
```

Licenciamento



Respeitadas as formas de citação formal de autores de acordo com as normas da ABNT NBR 6023 (2018), a não ser que esteja indicado de outra forma, todo material desta apresentação está licenciado sob uma [Licença Creative Commons - Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).