

Domain Driven Design

Eliane Marion

FIAP

2025

Manipulação de Arquivos





01

APROFUNDANDO OS
CONCEITOS

Pacote java.io

Para a manipulação de arquivos em Java é necessário utilizar o pacote `java.io` que é um dos pacotes mais importantes da linguagem, pois fornece classes e interfaces para entrada e saída de dados em vários formatos, como arquivos, rede, teclado, dentre outros.

Vamos conhecer as principais classes desse pacote.

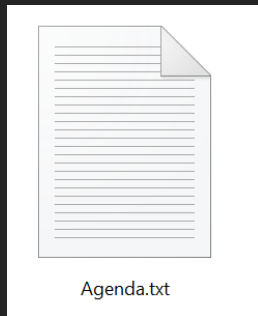


A classe File

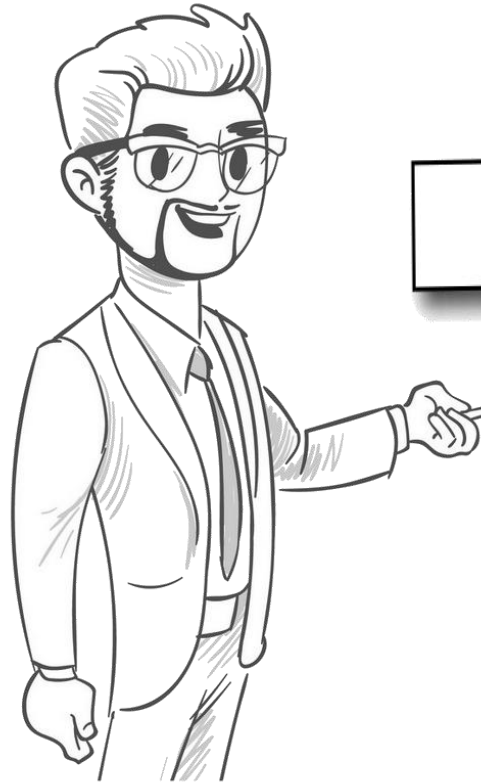
A classe File representa um arquivo ou diretório no sistema de arquivos do computador, permitindo que você crie, delete, liste e manipule arquivos e diretórios.

Para criar um objeto File, você precisa passar o caminho do arquivo ou diretório como argumento para o construtor.

```
File file = new File("C:\\Agenda.txt");
```



A classe File tem vários métodos úteis para interagir com arquivos e diretórios



`exists()`

`isDirectory()`

`canRead()`

`canWrite()`

`isFile()`

`mkDir()`

`delete()`



A Classe FileWriter e FileReader

São utilizadas para ler e escrever dados em arquivos de texto, lembrando que ambos trabalham com caracteres.

Leitura

```
File file = new File("C:\\Agenda.txt");
FileReader reader = new FileReader(file);
int data = reader.read();
while (data != -1)
{
    System.out.print((char) data);
    data = reader.read();
}
reader.close();
```

Escrita

```
File file = new File("C:\\saida.txt");
FileWriter writer = new FileWriter(file);
writer.write("Olá, mundo!");
writer.close();
```

Outras classes importantes

BufferedReader e BufferedWriter

São usadas para ler e gravar arquivos de texto de maneira eficiente, lendo e escrevendo uma linha por vez. Elas usam um buffer para armazenar os dados, o que torna a leitura e escrita mais rápida.

FileInputStream e FileOutputStream

São usadas para ler e gravar dados binários em um arquivo, como imagens e arquivos de áudio

ObjectInputStream e ObjectOutputStream

São usadas para ler e gravar objetos em arquivos. Isso permite que você armazene objetos Java em arquivos para uso posterior ou para transferência entre diferentes aplicações

ArquivoCupom

Vamos criar uma Classe com o nome ArquivoCupom e nela vamos implementar os métodos de escrita e leitura do arquivo

```
public static void escriptor(String path) throws IOException {  
    BufferedWriter buffWrite = new BufferedWriter(new FileWriter(path));  
    String linha = "";  
    Scanner in = new Scanner(System.in);  
    System.out.println("Escreva algo: ");  
    linha = in.nextLine();  
    buffWrite.append(linha + "\n");  
    buffWrite.close();  
}
```

Entendendo o método escritor

```
public static void escritor(String path)
```

O método escritor é do tipo void e recebe uma String path, esta string é o caminho do arquivo que será escrito.

```
BufferedWriter buffWrite = new BufferedWriter(new FileWriter(path));
```

Precisamos instanciar um objeto do tipo **BufferedWriter**, essa classe é a responsável por gerar um buffer que será utilizado para realizar a escrita do arquivo por meio do método append.

ArquivoCupom

```
public static void leitor(String path) throws IOException {  
    BufferedReader buffRead = new BufferedReader(new FileReader(path));  
    String linha = "";  
    while (true) {  
        if (linha != null) {  
            System.out.println(linha);  
        } else  
            break;  
        linha = buffRead.readLine();  
    }  
    buffRead.close();  
}
```

Entendendo o método escritor

```
public static void leitor(String path)
```

O método leitor é do tipo void e recebe uma String path, esta string é o caminho do arquivo que será manipulado.

```
BufferedReader buffRead = new BufferedReader(new FileReader(path));
```

Precisamos instanciar um objeto do tipo **BufferedReader**, essa classe é a responsável por gerar um buffer que será utilizado para realizar a leitura do arquivo por meio do método `readLine()`.

Principal

Para testar nosso código vamos criar uma Classe com o nome **Principal** e nela vamos implementar o método **main** e chamar nosso métodos da classe **ArquivoCupom**.

```
public class Principal {  
    new *  
    public static void main(String[] args) throws IOException {  
  
        String path = "C:\\\\Users\\eliana\\Documents\\2024\\fiap\\fiap-livraria\\Cupom.txt";  
        ArquivoCupom.escriptor(path);  
        ArquivoCupom.leitor(path);  
    }  
}
```

Não se esqueça de alterar o caminho do seu arquivo

Imprimindo o Cupom

Agora vamos utilizar o método estudado para imprimir nosso cupom fiscal. Na classe Registro de Vendas vamos implementar um método imprimirCupom

```
public void imprimirCupom(String path) throws IOException {  
    BufferedWriter buffWrite = new BufferedWriter(new FileWriter(path));  
    try{  
        buffWrite.append("  CUPOM FISCAL  \n");  
        buffWrite.append("-----\n");  
        buffWrite.append("  ITENS COMPRADOS  \n");  
        buffWrite.append("-----\n");  
        buffWrite.append("Título adquirido   - Preço\n");  
        lista.forEach(e ->  
        {  
            try {  
                buffWrite.append(e.getTitulo() + " - " + e.getPreco() + "\n");  
            } catch (IOException ex) {  
                throw new RuntimeException(ex);  
            }  
        }  
    });  
}
```

Imprimindo o Cupom

Continuação

```
        buffWrite.append("-----\n" +  
            "\nValor total da sua compra: R$ " + total);  
    }catch (IOException e){  
        System.out.println(e.getMessage());  
    }finally {  
        buffWrite.close();  
    }  
}
```

Chamando o método imprimirCupom

Na classe CriarLivros acrescente as duas últimas linhas

```
RegistroVendas carrinho = new RegistroVendas();  
carrinho.adicionar(favorito);  
carrinho.adicionar(meuLivro);  
carrinho.adicionar(livroDigital);  
carrinho.adicionar(sql);  
carrinho.exibir();  
String path = "C:\\Users\\eliana\\Documents\\2024\\fiap\\fiap-livraria\\Cupom.txt";  
carrinho.imprimirCupom(path);
```


Seu arquivo Cupom.txt

```
CUPOM FISCAL
-----
ITENS COMPRADOS
-----
Título adquirido - Preço
0 Hobbit -      37.27
UML 2.0 -      58.98
Senhor dos anéis - 0.0
SQL Magazine - v25 - 25.0
-----
Valor total da sua compra: R$ 121.25
```

Seu arquivo ficou assim !!!



02

Outros exemplos



Gravando e lendo dados de um
arquivo em Java

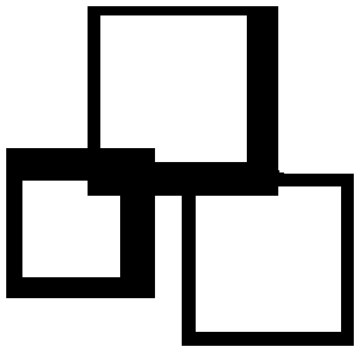
Trabalhando com arquivos
texto em Java



Java: Arquivos e
fluxos de dados



Arquivos Java, Caminho



OBRIGADA

To be continued...

