

Disciplina: Programação III

Tema: Arquivos em java

PROFESSOR: DANILO MONTEIRO

EMAIL: PROF.DANILO.MONTEIRO@GMAIL.COM



UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Objetivos da aula

Promover o conhecimento sobre arquivos em java

Leitura e escrita de arquivos

A **leitura e escrita de arquivos por um programa** pode se muito útil, seja para servir como arquivo de configurações e evitar a necessidade de modificação frequente do código fonte do próprio programa para cada cenário ou simplesmente para servir para entrada e saída de dados.

Por que é importante ?

Você pode querer criar um arquivo de configuração para seu sistema;

Você pode querer armazenar dados de maneira simples sem um banco de dados;

Você pode querer ler outros arquivos ou escrever um arquivo para outro programa (escrever ou ler um arquivo para o Excel por exemplo) [uma coisas que eu mais fiz com arquivos!]

Você pode querer ler ou escrever em um cookie de algum site da internet [fazia muito isso também nos meus sistemas antigamente]

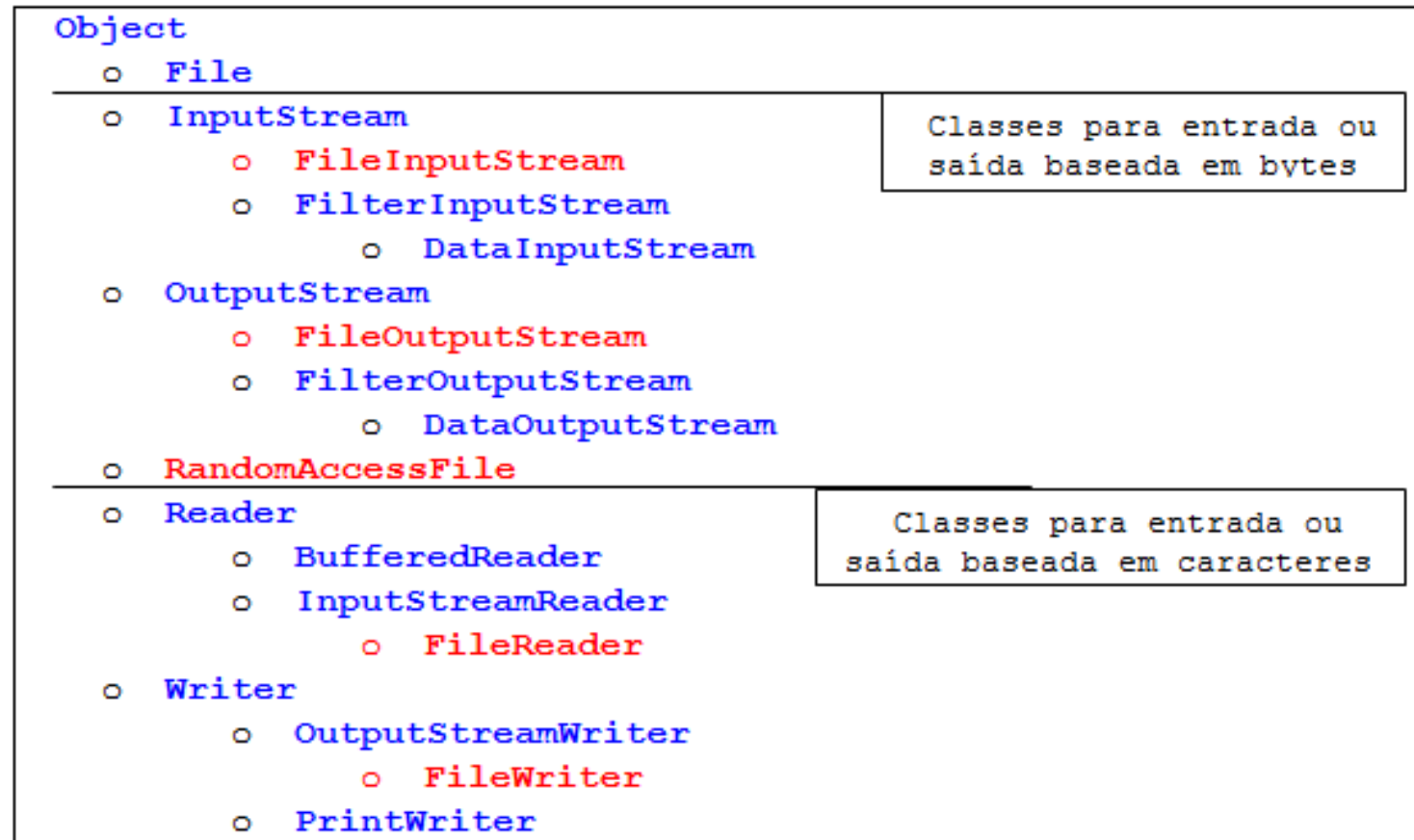
- Cookies armazenam dados seus em relação a sites que você visita, sabe quando você visita o mercado livre e fica aquelas propagandas no facebook do que você procurou? É alguém usou arquivos para aquilo 😊

Para isso nos temos...

A classe File

- Com diversos métodos, construtores e detalhes
- Vale a pena olhar o que faz a classe na sua documentação do java
- <https://docs.oracle.com/javase/7/docs/api/java/io/File.html>
- (tire uns 10 minutos para ler e entender as informações nesse site, se não souber inglês, coloca no tradutor 😊)

Além dela nos temos outras :



Alguns detalhes sobre essas outras 😊

Os arquivos são abertos criando-se objetos através de uma das classes de fluxo, citando:

- `FileInputStream`: para entrada baseada em bytes de um arquivo;
- `FileOutputStream`: para saída baseada em bytes para um arquivo;
- `RandomAccessFile`: para entrada e saída baseada em bytes de e para um arquivo;
- `FileReader`: para entrada baseada em caracteres de um arquivo;
- `FileWriter`: para saída baseada em caracteres para um arquivo.

A seguir vamos codar algumas coisas...

Você pode pegar os códigos no meu github

<https://github.com/dan1lo/ProgramacaoIII>


```
import java.io.File;
import java.io.IOException;

public class Principal {

    public static void main(String[] args) {

        File arquivo = new File("p3.txt");
        // este comando prepara para criar o arquivo p3.tx
        // eu poderia também colocar o caminho da pasta usando ... File("C:\\pasta\\nome.txt

        try {
            arquivo.createNewFile();
            System.out.println("Arquivo criado");
            // este comando cria um novo arquivo na pasta do projeto
        } catch (IOException e) {
            // TODO Auto-generated catch block
            System.out.println(" Deu problema ao criar um arquivo");
            e.printStackTrace();
        }

        if( ! arquivo.exists() ){
            System.out.println("o arquivo nao existe");
        } else {

            System.out.println("Arquivo já existe");
        }

    }
}
```

Este primeiro código utiliza a classe `File` para criar um arquivo na pasta do projeto (ou em outra pasta que você deseje).

Primeiro se estancia a localização(`path`) e o nome do arquivo, posteriormente se cria o arquivo (`createNewFile`), e com o método `exists()` eu verifico se ele foi criado.

O método `createNewFile()` que retorna `true` se o arquivo for criado sem problemas ou `false` se ele já existir. Lembrando que o método `createNewFile()` pode lançar uma exceção (`IOException`) então ele deve estar em um bloco `try...catch`.

» Este Computador » OS (C:) » Usuários » danil » eclipse-workspace » AulaArquivos »

Nome	Data de modificação	Tipo	Tamanho
.settings	17/04/2019 10:15	Pasta de arquivos	
bin	17/04/2019 10:16	Pasta de arquivos	
src	17/04/2019 10:16	Pasta de arquivos	
.classpath	17/04/2019 10:15	Arquivo CLASSPATH	1 KB
.project	17/04/2019 10:15	Arquivo PROJECT	1 KB
p3	17/04/2019 10:24	Documento de Texto	0 KB

isy: <terminated> Principal [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (17 de abr de 2019 10:24:19)
Arquivo criado

Este programa tem um equivoco, toda eu vez eu vou usar o método de criar o arquivo, mesmo se ele estiver sido criado, que tal se vocês arrumassem esse programa para ele apenas criar se o arquivo não existir ? 😊

Vou deixar esse pequeno desafio para vocês 😊

Com o comando `mkdir` eu posso criar o diretório...

Exemplo a seguir...









```
1 import java.io.File;
2 import java.io.IOException;
3
4 public class PrincipalDiretorios {
5     public static void main(String[] args) {
6         // podemos criar diretorios também
7         //
8         File directorio = new File( "novo"); // dou o nome do directorio e o caminho, neste caso tá dentro
9         // da pasta do projeto
10        if( directorio.mkdir() ){
11            System.out.println("Diretório criado");
12        }else{
13            System.out.println("Diretório não criado");
14        }
15        try{
16            File file = new File(diretorio, "file.txt");// aqui eu posso botar
17            // o arquivo file dentro do directorio
18            if( file.createNewFile() ){
19                System.out.println("Arquivo criado");
20            }else{
21                System.out.println("Arquivo não criado");
22            }
23        }catch(IOException ex){
24            ex.printStackTrace();
25        }
26
27    }
28 }
```

<terminated> PrincipalDiretorios [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (17 de abr de 2019 10:53:48)


Diretório criado

Arquivo criado

Este Computador > OS (C:) > Usuários > danil > eclipse-workspace > AulaArquivos >

Nome	Data de modificação	Tipo	Tamanho
 .settings	17/04/2019 10:15	Pasta de arquivos	
 bin	17/04/2019 10:47	Pasta de arquivos	
 novo	17/04/2019 10:53	Pasta de arquivos	
 src	17/04/2019 10:47	Pasta de arquivos	
 .classpath	17/04/2019 10:15	Arquivo CLASSPATH	1 KB
	17/04/2019 10:41	Documento de Texto	1 KB
 .project	17/04/2019 10:15	Arquivo PROJECT	1 KB
 p3	17/04/2019 10:24	Documento de Texto	0 KB

> Este Computador > OS (C:) > Usuários > danil > eclipse-workspace > AulaArquivos > novo

Nome	Data de modificação	Tipo	Tamanho
 file	17/04/2019 10:53	Documento de Texto	0 KB

Eu posso verificar se o caminho é um diretório com o método `isDirectory()` que retorna um `true` caso seja um diretório.

Assim como eu posso listar os arquivos dentro de uma pasta com o método `list()` que retorna um array de string

Por fim, eu posso pegar o nomes dos arquivo que retornam com o método `list` e interar(lembra da aula de array?) e mostrar todos os nomes dos arquivos na tela.

Exemplo a seguir


```
1 import java.io.File;
2
3 public class PrincipalListarDir {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         File dir = new File("novo");
9         if( dir.isDirectory() ){
10             String[] arqs = dir.list();
11             for(String nome : arqs){
12                 System.out.println(nome);
13             }
14         }
15     }
16
17 }
18
```

<terminated> PrincipalListarDir [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (17 de abr de 2019 10:58:33)

file.txt

Podemos deletar um arquivo também com o método `delete()`

Nó código a seguir, criamos o arquivo “arquivo_deletar”, verificamos se ele existe e posteriormente deletamos ele, verificando

```

1 import java.io.File;
2 import java.io.IOException;
3 public class PrincipalDeletarArquivo {
4     public static void main(String[] args) {
5         File arquivo = new File("arquivo_deletar.txt");
6         try {
7             arquivo.createNewFile();
8             System.out.println("Arquivo criado");
9         } catch (IOException e) {
10             System.out.println(" Deu problema ao criar um arquivo");
11             e.printStackTrace();
12         }
13         if( ! arquivo.exists() ){
14             System.out.println("o arquivo nao existe");
15         } else {
16             System.out.println("Arquivo já existe");
17         }
18         if( arquivo.delete() ){ // comando para deletar um arquivo
19             System.out.println("O arquivo foi deletado");
20         } else {
21             System.out.println("O arquivo não foi deletado");
22         }
23
24         if( ! arquivo.exists() ){//verifico se ele existe novamente
25             System.out.println("o arquivo nao existe");
26         } else {
27
28             System.out.println("Arquivo já existe");
29         }
30     }
31 }

```

Problems @ Javadoc Declaratio

<terminated> PrincipalDeletarArquivo

Arquivo criado
 Arquivo já existe
 O arquivo foi deletado
 o arquivo nao existe

Você pode usar o método `renameTo` da classe `File` para renomear uma classe, que tal tentar ?

Procure na documentação que passei no início da aula, é simples de se usar 😊

Vamos agora escrever e ler em
um arquivo:

Para escrever nos vamos usar a classe `bufferwriter`

<https://docs.oracle.com/javase/7/docs/api/java/io/BufferedWriter.html>

Escrever

```
1+ import java.io.BufferedWriter;
5
6 public class PrincipalEscrever {
7
8     public static void main(String[] args) throws IOException {
9         BufferedWriter arquivo = new BufferedWriter(new FileWriter("p3.txt"));
10        // assim apaga tudo no arquivo e escreve novamente
11        // caso vc queira adicionar na ultima linha usar:
12        //BufferedWriter arquivo = new BufferedWriter(new FileWriter("p3.txt", true));
13        String linha = "";
14        Scanner in = new Scanner(System.in);
15        System.out.println("Escreva algo para colocar no arquivo p3.txt: ");
16        linha = in.nextLine(); // para pegar o que o user escreveu
17        try {
18            arquivo.append(linha + "\n");//aqui eu escrevo
19            System.out.println("escreve no arquivo");
20        } catch (IOException e) {
21            e.printStackTrace();
22        } finally {
23            arquivo.close();
24            System.out.println("fechar o arquivo sempre, mesmo se deu erro ou não");
25        }
26
27    }
28 }
29
```

Agora vamos ler um arquivo com a classe buffer read

<https://docs.oracle.com/javase/7/docs/api/java/io/BufferedReader.html>

Ler arquivo

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Principaller {

    public static void main(String[] args) throws IOException {
        BufferedReader arquivo = new BufferedReader(new FileReader("p3.txt"));
        String linha = "";
        do { // enquanto linha for diferente de vazio, ele tem que rodar ao menos uma vez
            linha = arquivo.readLine(); // ler linha atual
            if (linha!=null) {
                System.out.println(linha); // escreve linha atual na tela
            } else {

                System.out.println("-----Fim do arquivo-----"); //

            }

        } while (linha != null);

        arquivo.close(); // fecha o arquivo
    }
}
```

Você já sabe escrever e ler em arquivos
em java!

Leitura complementar para entendimento mais profundo

<https://www.caelum.com.br/apostila-java-orientacao-objetos/pacote-java-io/#inputstream-inputstreamreader-e-bufferedReader>

<https://www.inf.ufes.br/~vitorsouza/wp-content/uploads/java-br-curso-basico-novo-slides10.pdf>

<https://forum.imasters.com.br/topic/382481-resolvido%C2%A0lerescrever-strings-com-randomaccessfile/>