07

Mãos na massa: Escrita em um arquivo

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

Escrevendo em um arquivo

1) No projeto **java-io**, no pacote **br.com.alura.java.io.teste**, copie a classe `TesteLeitura`, dando o nome `TesteEscrita`. Após isso, na classe `TesteEscrita`, apague a leitura do arquivo:

```
public class TesteEscrita {

    public static void main(String[] args) throws IOException {

        InputStream fis = new FileInputStream("lorem.txt");
        Reader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        br.close();
    }

}
```

2) Modifique o código, onde houver ***input***, dessa vez será ***output*** e onde houver ***reader***, será ***writer***. Modifique também o nome das variáveis e o nome do arquivo:

```
public class TesteEscrita {

    public static void main(String[] args) throws IOException {

        OutputStream fos = new FileOutputStream("lorem2.txt");
        Writer osw = new OutputStreamWriter(fos);
        BufferedWriter bw = new BufferedWriter(osw);

        bw.close();
    }

}
```

3) Através do método `write`, do `BufferedWriter`, escreva no arquivo. Se você quiser escrever uma linha abaixo da outra, você antes precisa criá-la, através do método `newLine`, por exemplo:

```
public class TesteEscrita {

    public static void main(String[] args) throws IOException {

        OutputStream fos = new FileOutputStream("lorem2.txt");
        Writer osw = new OutputStreamWriter(fos);
        BufferedWriter bw = new BufferedWriter(osw);
```

```
            bw.write("Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod");
            bw.newLine();
            bw.write("tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam");

            bw.close();
        }

    }
```

Após a execução da classe, atualize o projeto no Eclipse e veja o novo arquivo criado.

Copiando um arquivo

4) Novamente no projeto **java-io**, no pacote **br.com.alura.java.io.teste**, copie a classe `TesteLeitura`, dando o nome `TesteCopiarArquivo`. Após isso, na classe `TesteCopiarArquivo`, com a leitura do arquivo já feita, estabeleça o fluxo de escrita:

```
public class TesteCopiarArquivo {

    public static void main(String[] args) throws IOException {

        InputStream fis = new FileInputStream("lorem.txt");
        Reader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        OutputStream fos = new FileOutputStream("lorem2.txt");
        Writer osw = new OutputStreamWriter(fos);
        BufferedWriter bw = new BufferedWriter(osw);

        String linha = br.readLine();

        while (linha != null) {
            System.out.println(linha);
            linha = br.readLine();
        }

        br.close();
    }

}
```

5) A leitura do arquivo já está sendo feita. Agora, ao invés de imprimir a linha no console, escreva-a no arquivo. Não esqueça de criar uma nova linha e de fechar o `BufferedWriter`:

```
public class TesteCopiarArquivo {

    public static void main(String[] args) throws IOException {

        InputStream fis = new FileInputStream("lorem.txt");
        Reader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        OutputStream fos = new FileOutputStream("lorem2.txt");
        Writer osw = new OutputStreamWriter(fos);
```

```
        BufferedWriter bw = new BufferedWriter(osw);

        String linha = br.readLine();

        while (linha != null) {
            bw.write(linha);
            bw.newLine();
            linha = br.readLine();
        }

        br.close();
        bw.close();
    }

}
```

6) A leitura não necessariamente precisa ser de um arquivo. Acima, se você utilizar o `System.in` , você irá gravar no arquivo o que o usuário digitar. Mas para conseguir parar o console, verifique também se a linha não estará vazia. Teste:

```
public class TesteCopiarArquivo {

    public static void main(String[] args) throws IOException {

        InputStream fis = System.in;
        Reader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);

        OutputStream fos = new FileOutputStream("lorem2.txt");
        Writer osw = new OutputStreamWriter(fos);
        BufferedWriter bw = new BufferedWriter(osw);

        String linha = br.readLine();

        while (linha != null && !linha.isEmpty()) {
            bw.write(linha);
            bw.newLine();
            linha = br.readLine();
        }

        br.close();
        bw.close();
    }

}
```

7) Do contrário, com o `System.out` , você consegue imprimir no console o que foi lido. Para tal, assim que criar uma nova linha, dê um *flush* no `BufferedWriter` . Por exemplo:

```
public class TesteCopiarArquivo {

    public static void main(String[] args) throws IOException {

        InputStream fis = System.in;
        Reader isr = new InputStreamReader(fis);
```

```
            BufferedReader br = new BufferedReader(isr);

            OutputStream fos = System.out;
            Writer osw = new OutputStreamWriter(fos);
            BufferedWriter bw = new BufferedWriter(osw);

            String linha = br.readLine();

            while (linha != null && !linha.isEmpty()) {
                bw.write(linha);
                bw.newLine();
                bw.flush();
                linha = br.readLine();
            }

            br.close();
            bw.close();
        }

    }
```