

Mãos na massa: Leitura e escrita mais simples

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

FileWriter

1) No projeto **java-io**, no pacote **br.com.alura.java.io.teste**, copie a classe **TesteEscrita**, dando o nome **TesteEscritaFileWriter**. Após isso, na classe **TesteEscritaFileWriter**, comente ou apague as linhas de criação do **FileOutputStream**, **OutputStreamWriter** e **BufferedWriter**:

```
public class TesteEscritaFileWriter {  
  
    public static void main(String[] args) throws IOException {  
  
        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();  
    }  
}
```

2) Instancie um **FileWriter**, passando o arquivo a ser escrito como parâmetro para o construtor. Em seguida, modifique o código, substituindo a instância **bw** pela instância do **FileWriter**:

```
public class TesteEscritaFileWriter {  
  
    public static void main(String[] args) throws IOException {  
  
        FileWriter fw = new FileWriter("lorem2.txt");  
  
        fw.write("Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod");  
        fw.newLine();  
        fw.write("tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam");  
  
        fw.close();  
    }  
}
```

3) O **FileWriter** não possui o método **newLine**, então substitua-o e, através do método **write**, escreva o caractere especial que representa uma nova linha, mas esse caractere varia de acordo com o sistema operacional. O Java possui um método que retorna esse caractere especial baseado no seu sistema operacional, o método **lineSeparator()**, de **System**:

```
public class TesteEscritaFileWriter {  
  
    public static void main(String[] args) throws IOException {  
  
        FileWriter fw = new FileWriter("lorem2.txt");
```

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

fw.close();
}
}
```

4) Para ter acesso ao método `newLine`, você pode voltar a utilizar o `BufferedWriter` para a escrita no arquivo, passando para ele o `FileWriter` criado anteriormente:

```
public class TesteEscritaFileWriter {

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

        FileWriter fw = new FileWriter("lorem2.txt");
        BufferedWriter bw = new BufferedWriter(fw);

        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();
    }
}
```

5) Você também pode criar a instância em uma só linha:

```
public class TesteEscritaFileWriter {

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

        BufferedWriter bw = new BufferedWriter(new FileWriter("lorem2.txt"));

        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();
    }
}
```

PrintStream e PrintWriter

7) Novamente no projeto `java-io`, no pacote `br.com.alura.java.io.teste`, copie a classe `TesteEscritaFileWriter`, dando o nome `TesteEscritaPrintStreamPrintWriter`. Após isso, na classe `TesteEscritaPrintStreamPrintWriter`, remova a criação da instância do `BufferedWriter`:

```
public class TesteEscritaPrintStreamPrintWriter {
```

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

    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();

}

}
```

8) Instancie um `PrintStream`, passando o arquivo a ser escrito como parâmetro para o construtor. Em seguida, modifique o código, substituindo a instância `bw` pela instância do `PrintStream`:

```
public class TesteEscritaPrintStreamPrintWriter {

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

        PrintStream ps = new PrintStream("lorem2.txt");

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

        ps.close();

    }

}
```

9) O `PrintStream` não possui o método `write` e sim o método `print`. Ele também não tem o método `newLine`, então no caso de precisar pular uma linha após ela ser escrita, há o método `println`:

```
public class TesteEscritaPrintStreamPrintWriter {

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

        PrintStream ps = new PrintStream("lorem2.txt");

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

        ps.close();

    }

}
```

10) Do mesmo jeito, faça com o `PrintWriter`:

```
public class TesteEscritaPrintStreamPrintWriter {

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

        PrintWriter pw = new PrintWriter("lorem2.txt");

        pw.println("Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod");

    }

}
```

```
pw.print("tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam");
```

```
pw.close();
```

```
}
```

```
}
```