



03. FileWrite e PrintStream

📅 Date	@10/10/2022
▼ Categoria	Java
▼ Curso	Java e java.io: Streams Reader e Writers

Revisão

Primeira data February 21, 2022

Tópicos

- Saída com FileWriter
 - Sobre o FileWriter
 - PrintWriter e PrintStream
 - Saída para arquivo de texto
 - Escrita independente do sistema
 - Mãos na massa: Leitura e escrita mais simples
 - Para saber mais: currentTimeMillis
 - O que aprendemos?
-

Para saber mais: currentTimeMillis

Um pouco mais sobre `System`.

Nessa aula, vimos mais um método da classe `System`, o método `lineSeparator`:

```
String newLine = System.lineSeparator();
```

Essa String possui os caracteres especiais que representam uma nova linha no seu sistema operacional (no Windows é `\r\n`, no Unix é `\n`).

Além disso, a classe `System` possui outros métodos, alguns são poucos usados, outros mais. Entre os mais usados está o método `currentTimeMillis()`:

```
long millis = System.currentTimeMillis();
```

Ao executar essa linha, recebemos os milissegundos que passaram desde 1 de janeiro de 1970. Essa data é considerada o início da *Era Unix* ou *Unix Epoch*, que o Java também usa. Ou seja, essa data é o marco zero no sistema de calendário usado nos sistemas operacionais UNIX.

Agora sendo mais prático, você verá esse método para medir o tempo de execução de algum trecho de código. Por exemplo, podemos medir a escrita:

```
public class TesteEscritaFileWriter {

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

        long ini = System.currentTimeMillis();

        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();

        long fim = System.currentTimeMillis();

        System.out.println("Passaram " + (fim - ini) + " milissegundos");

    }
}
```

O que aprendemos?

Nessa aula, vimos:

- Várias alternativas para estabelecer uma saída para um arquivo de texto, como:

- A classe `FileWriter`

- Podemos passar um `FileWriter` para um `BufferedWriter`.

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

- A classe `PrintStream`

- Foi adicionado no Java na versão 1.0

- A classe `PrintWriter`

- Foi adicionado no Java na versão 1.1

- Alguns detalhes da classe `System`, como :

- O atributo `System.out` é do tipo `PrintStream`

- O método `System.lineSeparator()` devolve os caracteres que representam uma nova linha

- O método `System.currentTimeMillis()` devolve os milissegundos que passaram desde 1 de janeiro de 1970