

# AfroDev

Java Básico - Aula 5 - Parte 2

# Java IO

- IO -> Input/Output
- “Habilita” a comunicação do JAVA com o mundo
- Permite que o JAVA faça “output” para o mundo
- Permite que o “mundo” faça o “input” para o JAVA
- Pense... você escreveu um programa. E daí... ele não precisa se comunicar com nada ou ninguém?
- Java por interagir com disco, dispositivos, outros dispositivos ou memória diretamente

# Streams

- Tipos primitivos (aqueles que já vimos)
- Objetos simples ou complexos (built-in Java ou aqueles que você mesmo criou)
- Mas... como ele sabe? Como ele consegue?
- <https://docs.oracle.com/javase/tutorial/essential/io/streams.html>
- <https://docs.oracle.com/javase/8/docs/api/java/io/InputStream.html>
- <https://docs.oracle.com/javase/8/docs/api/java/io/OutputStream.html>

# Byte Streams

- Mais primitivo possível
- Faz a leitura e a gravação de cadeia de bytes
- <https://docs.oracle.com/javase/tutorial/essential/io/bytestreams.html>

# Character Streams

- Mais indicado para arquivos textos
- A API já fica mais amigável para esse fim. Por exemplo: `readLine()`; Fica claro que esse método vai ler 1 única linha do arquivo e não uma quantidade de bytes.
- <https://docs.oracle.com/javase/tutorial/essential/io/charstreams.html>

# Buffered Streams

- “Decora” (técnica de adicionar um comportamento sem afetar o comportamento original) a leitura/escrita para fins de performance
- Cria um espaço temporário na memória, um buffer
- Já têm uma quantidade finita de dados a disposição
- <https://docs.oracle.com/javase/tutorial/essential/io/buffers.html>
- Você conseguiu ver que não mudou “nada”?

# Data Streams

- Mais amigável para dados
- Dados primitivos do Java e String
- <https://docs.oracle.com/javase/tutorial/essential/io/datastreams.html>
- E se tivesse que ler um tipo que não fosse primitivo ou String?

# Object Streams

- Consegue lidar com “qualquer” objeto
- Qualquer objeto serializável
- Um objeto é serializável quando ele implementa essa interface Serializable.
- Você consegue pensar em algum problema de segurança com a frase acima?
- <https://docs.oracle.com/javase/tutorial/essential/io/datastreams.html>



# Java IO - Exercícios

- Escreva um programa que faça a cópia de **uma foto** no seu disco
- Leia o conteúdo em:  
<https://docs.oracle.com/javase/tutorial/essential/io/datastreams.html>
- Altere o programa exemplo com a seguinte lógica:
  - Se o valor total for maior que 10 “dinheiros”, escreva no arquivo mais uma coluna com a mensagem “produto caro”
  - Se o valor total for menor ou igual a 10 “dinheiros”, escreva no arquivo mais uma coluna com a mensagem “produto ok”
  - Na leitura do arquivo, imprima no console a nova coluna criada.
- Abra o arquivo “invoicedata” num editor de texto. O conteúdo está normal?

# Java IO - Exercícios

- Crie um objeto qualquer. Escreva ele em um arquivo no disco
- Faça a leitura desse objeto

# Java IO - Vamos pensar

- Um programa Java pode “conversar” com um programa escrito em outra linguagem?
- Para leitura/escrita de arquivos, você precisou se preocupar com o sistema operacional?
- Ler um arquivo no Linux ou no Windows faz com que você tenha que alterar o código?
- Um double de outra linguagem pode ser atribuído a um double do Java?

# Revisão

- Aprendemos o que é IO em Java é o jeito do Java conversar com o mundo
- A leitura/escrita vai desde bytes (mais baixo nível) até objetos (mais alto nível)
- Aprendemos que para serializar/desserializar um objeto é preciso que ele implemente a interface Serializable
- Existe um “decorator” que acelera o processo de leitura/escrita através de um buffer