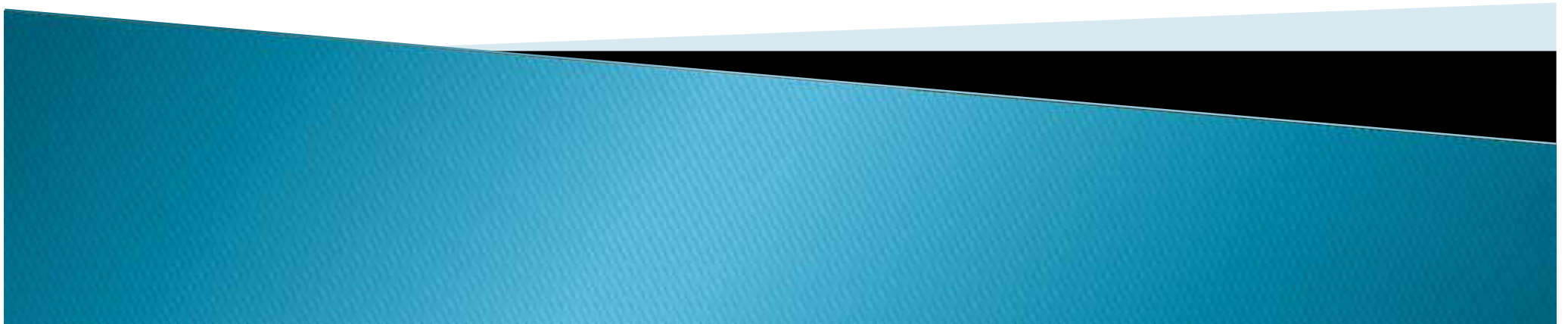


# Linguagem de Programação Orientada a Objetos 2

Streams em Java  
Prof. Tales Viegas

<https://fb.com/ProfessorTalesViegas>



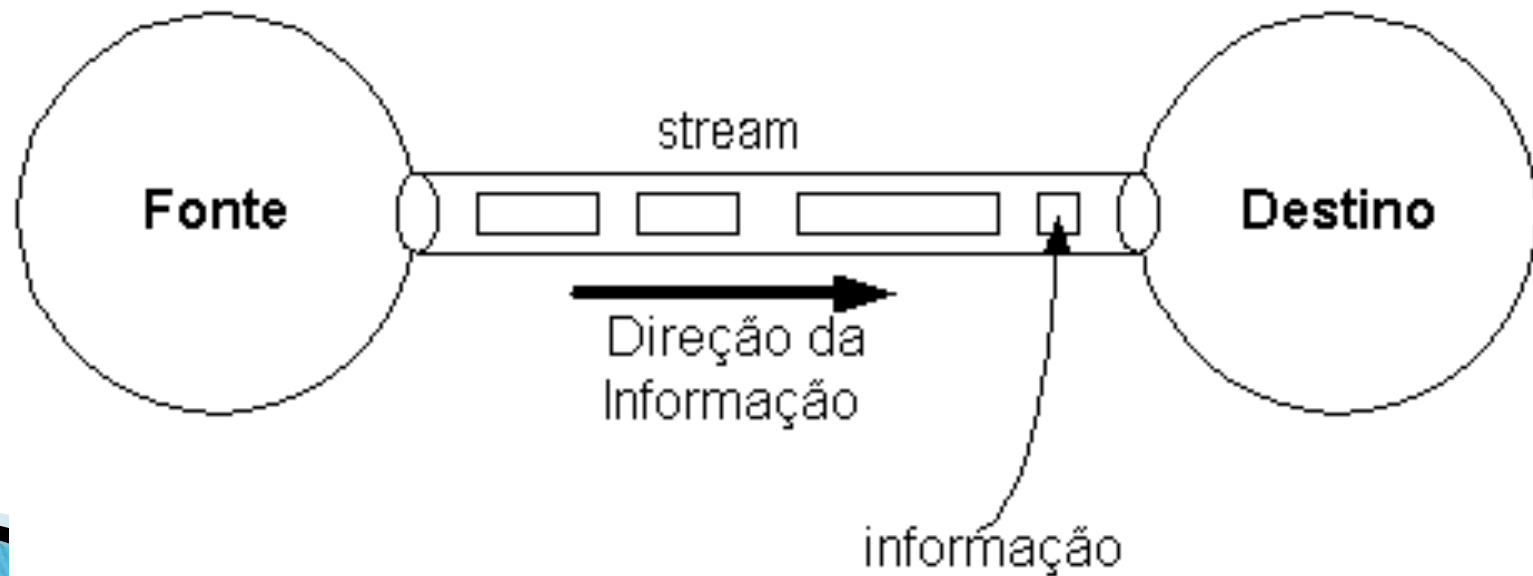
# Stream

- ▶ Abstração que representa uma fonte genérica de entrada de dados ou um destino genérico para escrita de dados
- ▶ É independente do dispositivo físico concreto
- ▶ Todas as classes que implementam um stream em Java são subclasses das classes abstratas **InputStream** ou **OutputStream**



# Streams

- ▶ Caminho por meio do qual os programas podem enviar uma sequência de bytes de uma fonte até um destino.



# Streams – Características

## ► Direção:

- Entrada: as classes possuem a palavra Input (ex. InputStream).
- Saída : as classes possuem a palavra output (ex. OutputStream).



# Streams – Características

## ▶ Fonte e Destino:

- Memória (array de bytes / caracteres)
- Arquivos
- Console (teclado, mouse, vídeo, impressora)
- Conexão de rede



# Streams – Características

## ▶ Tipo de Informação :

- Bytes puros
- Caracteres Unicode: informações do tipo texto. As classes possuem a palavra Reader ou Writer (ex. FileReader)
- Tipos nativos: inteiros, doubles, caracteres, etc As classes possuem a palavra Data (ex. DataInputStream)
- Objetos. As classes possuem a palavra Object (ex. ObjectInputStream)



# File – Visão Geral

- ▶ Representa arquivos e diretórios.
  - É apenas uma abstração: a existência de um objeto File não significa a existência de um arquivo ou diretório
  - Contém métodos para testar a existência de arquivos, para definir permissões, para apagar arquivos, criar diretórios, listar o conteúdo de diretórios, etc.



# File – Métodos

- ▶ `File (String pathname)` : construtor. Serve para definir o nome do arquivo ou diretório.
- ▶ `boolean createNewFile()` : cria fisicamente o arquivo em disco.
- ▶ `boolean exists()` : retorna verdadeiro se o arquivo ou diretório existem.





# File – Alguns Métodos

- ▶ `boolean isFile()` : retorna verdadeiro se for um arquivo
- ▶ `boolean isDirectory()` : retorna verdadeiro se for um diretório
- ▶ `String getName ()` : retorna o nome do arquivo
- ▶ `String getPath()` : retorna o caminho do arquivo



# File – Métodos

- ▶ `long lastModified()` : retorna um valor referente a data da última modificação, medido em milissegundos desde 1/01/1970
- ▶ `boolean delete()`: tenta apagar o diretório ou arquivo
- ▶ `long length()`: retorna o tamanho do arquivo em bytes



# File – Métodos

- ▶ `boolean mkdir()`: cria um diretório com o nome do arquivo
- ▶ `String[] list()`: retorna lista de arquivos contido no diretório



# E/S Caracteres Unicode

## ► Writer

- Classe abstrata para lidar com fluxos de caracteres de saída
- Subclasse: PrintWriter
- Principais Métodos:
  - void print( expr ): grava expr no arquivo
  - void println( expr): grava expr no arquivo e coloca um \n após
  - void flush(): esvazia o buffer
  - void close(): fecha o stream (essencial)



# E/S Caracteres Unicode

## ▶ Reader

- Classe abstrata para lidar com fluxos de caracteres de entrada
- Subclasse: Scanner
- Principais métodos:
  - `String next()`: lê um caractere (16 bits) por vez
  - `String NextLine()`: Lê uma linha do arquivo
  - `boolean hasNextLine()`: verifica se o arquivo possui novas linhas a serem lidas
  - `void close()`: fecha o stream



# E/S de Objetos

- ▶ Utiliza as classes
  - ObjectOutputStream
  - ObjectInputStream
- ▶ Objetos a serem gravados devem ser “serializáveis”
  - Serialização – transformação de uma estrutura de dados em um sequência de bytes
  - Todos os atributos também deverão ser serializáveis



# E/S de Objetos

- ▶ No caso do atributo ser uma referência para um outro objeto, o outro objeto também é enviado (ou lido) para o Stream

