

# PROGRAMAÇÃO ORIENTADA A OBJETOS

---

Arquivos, Streams e Serialização

Prof. Emanuel Barreiros



# Arquivos e Streams

- Java enxerga cada arquivo como um stream de bytes
- Todo sistema operacional fornece meios de sinalizar o fim de um arquivo
- Programas Java não precisam saber como o SO sinaliza esses arquivos, pois isso é gerenciado pela JVM

# Streams de bytes e de caracteres

- Streams de arquivo podem ser usados para obter e gravar dados tanto como bytes quanto como caracteres
- Streams de bytes manipulam dados em seu formato binário
- Streams de caracteres manipulam dados como uma sequencia de caracteres
  - Se o valor 5 fosse guardado em formato binário, ele seria gravado como 101, sua representação binária
  - Se o o valor '5' fosse guardado como um caractere, ele seria guardado como 00000000 00110101, a representação binária do número 53, o código unicode do caractere '5'

# O Pacote java.io

- Classes para processamento de arquivos estão nesse pacote
- Possui as classes
  - FileInputStream: leitura de arquivos binários
  - FileOutputStream: escrita de arquivos binários
  - FileReader: leitura de arquivos baseados em caracteres
  - FileWriter: gravação de arquivos baseados em caracteres

# A classe File

- Ela é usada para representar um arquivo ou um diretório
- Em um dos construtores da classe File, podemos especificar o caminho de um arquivo ou diretório
- O caminho pode ser absoluto ou relativo
- O caminho absoluto inicia na raiz de diretórios, enquanto que o relativo inicia onde o programa está localizado no disco

# Demonstração da classe File

- Demonstração no código.

# Acesso sequencial para escrita em arquivos

- Demonstração no código

# Acesso sequencial para leitura de arquivos

- Dados são guardados em arquivos para que possam ser lidos e processados em um momento futuro
- Precisamos ser capazes de ler esses arquivos
- Java fornece classes que conseguem fazer isso
- Vamos ver como funciona isso no código!



# Serialização de objetos

- As classes que vimos até agora conseguem guardar nossos registros de contas
- No entanto, nem sempre sabemos exatamente como os dados serão guardados (seu tamanho, por exemplo)
- Perdemos também os tipos dos dados
- O Java provê um mecanismo que consegue gravar dados de um objeto em um arquivo e recuperá-lo para recriar o objeto inteiramente na memória
- Isto chama-se serialização

# Serialização de objetos

- As classes `ObjectInputStream` e `ObjectOutputStream` são responsáveis pela serialização
- Basicamente, o que o Java faz, é salvar os objetos em um formato padrão, o que permite que eles possam ser lidos mais tarde e os objetos reconstruídos.
- Exemplo no código.