

Arquivos

Computação Orientada a Objetos

Prof. Flávio Luiz Coutinho

EACH/USP

Arquivos

- Por que usar?
 - Armazenamento de dados a longo prazo.
 - Existem mesmo depois que os programas que os manipulam encerram sua execução.
 - Memória secundaria (seu estado é mantido mesmo depois que a máquina é desligada).
 - Dados **persistentes**.

Arquivos

- Organização:
 - Bit (0 ou 1).
 - Byte (8 bits).
 - Campo (agrupamento de bytes com significado).
 - Registro (grupo de campos relacionados).
 - Arquivo (grupo de registros relacionados).
 - Banco de dados (grupo de arquivos relacionados).
- Definição mais geral:
 - Dados arbitrários em formatos arbitrários (para muitos sistemas operacionais arquivos são simplesmente uma coleção de bytes).

Arquivos

- **Fluxo (stream):**
 - Programas Java enxergam arquivos como um **fluxo** sequencial de bytes.
 - Realizam processamento de arquivos usando classes do pacote **java.io** (inclui definições para diversas classes de fluxo).
- Arquivos binários:
 - Fluxo de bytes.
- Arquivos de texto:
 - Fluxo de caracteres.
- Maneira como os bytes são interpretados.

Arquivos

- Fluxos (streams) já são velhos conhecidos:
 - System.in (**InputStream**).
 - System.out (**PrintStream**).
 - System.err (**PrintStream**).
- Um fluxo de dados pode ter diversas origens:
 - Console (System.in, System.out, System.err).
 - Conexão de rede.
 - **Arquivos.**

Arquivos

- Procedimento geral:
 - Abrir fluxo.
 - Enquanto houver dados ler/escrever.
 - Fechar o fluxo.

Arquivos

- Classe **File**.
- Classes importantes (fluxo de bytes):
 - **InputStream** (abstrata).
 - **FileInputStream**.
 - **ObjectInputStream**.
- **OutputStream** (abstrata).
- **FileOutputStream**.
- **ObjectOutputStream**.

Arquivos

- Classes importantes (fluxo de bytes):
 - **FilterInputStream.**
 - **DataInputStream.**
 - **FilterOutputStream.**
 - **DataOutputStream.**
 - **PrintStream.**

Arquivos

- Classes importantes (fluxo de caracteres):
 - **Reader** (abstrata).
 - **InputStreamReader** (ponte entre InputStream e Reader).
 - **FileReader** (conveniência).
 - **BufferedReader** (bufferizado + leitura de linha).
- **Writer** (abstrata).
- **OutputStreamWriter** (ponte).
- **FileWriter** (conveniência).
- **PrintWriter**.

Arquivos

- Exemplos.