Linguagem de Programação Java & Web

JAVA – *Manipulação de Arquivos*

Profa. Joyce Santos

Entrada & Saída

- Tópicos
  - Classes de Manipulação de Arquivos;

Entrada & Saída

- Manipulação de Arquivos
  - **Classe File**
    - Esta classe é uma representação abstrata de um arquivo ou diretório.
    - A classe **File** oferece métodos comuns para:
      - verificar se o caminho existe na máquina local;
      - criar ou remover o arquivo/diretório com o nome especificado;
      - pegar o caminho completo do arquivo/diretório;
      - listar todos os arquivos do caminho do diretório;
      - verificar as propriedades do arquivo (data, readonly, etc);

Entrada & Saída

- Manipulação de Arquivos
  - **Classe File**
    - **File diretorio = new File(“D://Aluno”);**
    - **File arquivo= new File(“D://Aluno/nota.txt”);**

Entrada & Saída

- Manipulação de Arquivos
  - **Classe File – Alguns métodos**
    - **exists**
      - Verifica se o caminho informado existe
    - **isDirectory**
      - Verifica se o caminho informado é um diretório
    - **list**
      - No caso de um diretório, retorna como um vetor os arquivos do diretório
    - **createNewFile**
      - Cria um novo arquivo

Entrada & Saída

```
import java.io.*;
class Arquivo{
    public static void main(String args[]){
        try{
            File arq = new File("D:\\texto.txt");
            //arq representa o arquivo texto.txt
            File dir = new File("D:\\");
            //dir representa o diretório D:
            if( dir.exists() && dir.isDirectory() ) {
                String[] arquivos = dir.list(); //lista os arquivos

                for(int i=0;i < arquivos.length;i++){
                    System.out.println(arquivos[i]);
                }
            }
        }catch (Exception e) {
            System.out.print(e.toString());
        }
    }
```

Entrada & Saída

```
import java.io.*;
class CriandoArquivo{
    public static void main(String args[]){
        try{
            File arq = new File("D:\\criandoArquivo.txt");
            if(arq.exists()){
                System.out.println("O arquivo já existe. ");
            }else{
                arq.createNewFile();
            }
        }catch (Exception e) {
            System.out.print(e.toString());
        }
    }
```

Entrada & Saída

- Manipulação de Arquivos
  - **Classe File**
    - Para acessar os dados dos arquivos faremos uso dos **Streams** (fluxos) de entrada e saída de dados, das classes **Reader** e **Writer**.
  - **Reader e Writer**
    - Um exemplo de classes Readers e Writers são as classe **FileReader** e **FileWriter**.

```java
import java.io.*;
class AcessoReaderWriter{
    public static void main(String args[]){
        try{
            File arq = new File("D:\\readerwriter.txt");
            PrintWriter fw = new PrintWriter( arq );
            fw.println("Linha 1");
            fw.println("Linha 2");
            fw.close();

            FileReader fr = new FileReader( arq );
            BufferedReader buf = new BufferedReader(fr);
            while(buf.ready()){
                String linha = buf.readLine();
                System.out.println(linha);
            }

            fr.close();
            buf.close();
        }catch (Exception e) {
            System.out.print(e.toString());
        }
    }
}
```

Exercício

- Criar um programa que, graficamente, apresente o seguinte menu:
  - 1. Criar um arquivo
  - 1.1 Passar o nome do arquivo a ser criado
  - 2. Deletar um arquivo
  - 2.1 Passar o nome do arquivo a ser deletado
  - 3. Escrever um conteúdo em um arquivo
  - 3.1 Passar o nome do arquivo a ser modificado
  - 3.2 Passar o conteúdo que será escrito
  - 4. Ler o conteúdo de um arquivo
  - 4.1 Passar o nome do arquivo a ser lido

Entrada & Saída

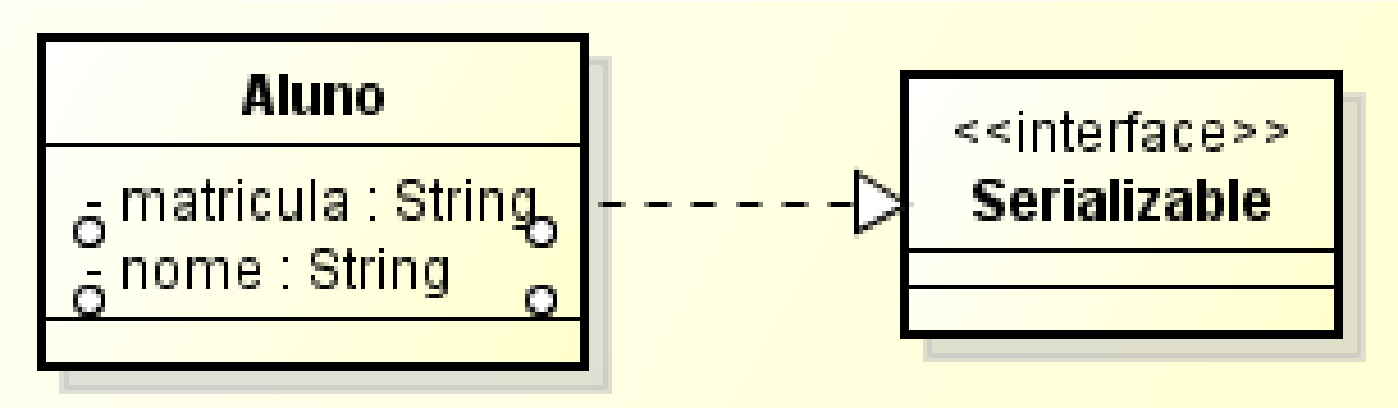
- **Serialização de Objetos**
  - **Objeto serializado**
    - Objeto será transformado em bytes, e poderá ser armazenado em disco ou transmitido via **Stream**.
  - **Stream**
    - É um objeto de transmissão de dados.
    - A transmissão do fluxo de dados serial se dá a partir de uma origem e de um destino.
    - Permite a manipulação de objetos serializados.

Entrada & Saída

- **Serialização de Objetos**
  - **Tipos de Stream**
    - **FileOutputStream**
      - Gravação em disco
    - **FileInputStream**
      - Leitura do disco
  - **Manipulação de Objetos Serializados**
    - **ObjectInputStream**
      - Insere objetos serializados no Stream
    - **ObjectOutputStream**
      - Recuoera objetos serializados no Stream

Entrada & Saída

- **Serialização de Objetos**
  - **Estudo de Caso**



Entrada & Saída

Serialização de Objetos

```
public class SavingObjectInDisk {
    public static void main(String[] args) {
        Aluno aluno = new Aluno("123456", "Fulano de Tal");
        try {
            //Gerando arquivo para armazenar objeto
            FileOutputStream file =
                    new FileOutputStream("C:\\ObjetosJava\\alunos.dat", true);
            //Classe responsavel por inserir o objeto
            ObjectOutputStream obj = new ObjectOutputStream(file);
            //gravando objeto no arquivo
            obj.writeObject(aluno);
            //limpando memória
            obj.close();
            file.close();
            System.out.println("Objeto gravado com sucesso!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Entrada & Saída

Serialização de Objetos

```
public class ReadingObjectFromDisk {
    public static void main(String[] args) {
        try {
            //Carregando arquivo
            FileInputStream file =
                    new FileInputStream("C:\\ObjetosJava\\clientes.dat");
            //Classe responsavel por recuperar o objeto
            ObjectInputStream obj = new ObjectInputStream(file);
            //Recuperando conteúdo do arquivo
            Usuario cliente = (Usuario) obj.readObject();
            System.out.println("Cliente:\n" + cliente);
            obj.close();
            file.close();
            System.out.println("Objeto gravado com sucesso!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Entrada & Saída

- **Serialização de Objetos**
  - **Pratique!**