UFRPE

Entrada e Saída

Richarlyson A. D'Emery
site: https://sites.google.com/site/profricodemery/mpoo
grupo: http://groups.google.com/group/**mpoo_uast**
email grupo: **mpoo_uast**@googlegroups.com
contato: rico_demery@yahoo.com.br

Sumário

UFRPE

A Classe File

- A classe File é usada para representar o nome e um arquivo ou um diretório
  - `File f1 = new File("/"); // root`
  - `File f2 = new File("/", "etc/password");`
  - `File f3 = new File("config.sys");`
- É possível obter várias informações sobre o arquivo:
  - Se existe, se permite leitura, se permite escrita
  - Quando foi a última modificação, e qual o seu tamanho
- É possível criar e apagar arquivos
- O método list() da classe File retorna uma lista com nomes dos arquivos do diretório

Métodos da Classe File

UFRPE

- Métodos que retornam informações sobre o File:
  - String getName()
  - String getParent()
  - String getPath();
  - long lastModified()
  - long length();
- Métodos para Manipulação:
  - boolean delete()
  - boolean renameTo(File novoNome)

Métodos da Classe File

UFRPE

- Métodos específicos para diretórios:
  - boolean mkdir()
  - String [ ] list()
- Métodos de teste:
  - boolean exists()
  - boolean canRead()
  - boolean canWrite()
  - boolean isDirectory()
  - boolean isFile()

Entrada e Saída Padrão

UFRPE

- `System.out` permite escrever na saída padrão
  - **out** é uma instância da classe `PrintStream`
- `System.in` permite ler na saída padrão
  - **in** é uma instância da classe `InputStream`
- São atributos públicos da classe `System`

Saída Padrão

UFRPE

- Entendendo o `System.out.println`:
  - **System** é uma classe
  - **System** tem uma instância estática chamada **out**
  - **Out** é uma instância da classe **PrintStream**
  - A classe **PrintStream** tem um método **println()**

Stream – Fluxo de Dados

UFRPE

- Java suporta dois tipos de stream
  - de bytes e
  - de caracteres
- A entrada e saída de caracteres é feita por instâncias das classes **Reader** e **Writer**
- A entrada e saída de bytes é feita por instâncias das classes **InputStream** e **OutputStream**

Classes Básica para Entrada e Saída

	Bytes	**Caracteres**
Entrada	InputStream	Reader
Saída	OutputStream	Writer

Métodos da Classe InputStream

UFRPE

- Métodos para leitura de bytes:
  - `int read()`
  - `int read(byte [ ] buffer)`
  - `int read(byte [ ] buffer, int offset, int tam)`
- Outros métodos:
  - `void close()`
  - `int available()`
  - `void skip(long n)`

Métodos da Classe OutputStream

- Métodos para escrita de bytes:
  - `int write()`
  - `int write(byte [ ] buffer)`
  - `int write(byte [ ] buffer, int offset, int tam)`
- Outros métodos:
  - `void close()`
  - `void flush()`

Métodos da Classe Reader

UFRPE

- Métodos para leitura de caracteres:
  - `int read()`
  - `int read(char [ ] buffer)`
  - `int read(char [ ] buffer, int offset, int tam)`
- Outros métodos:
  - `void close()`
  - `boolean ready()`
  - `void skip(long n)`

Métodos da Classe Writer

- Métodos para escrita de caracteres:
  - `int write(int c)`
  - `int write(char [ ] buffer)`
  - `int write(char [ ] buffer, int offset, int tam)`
  - `int write(String s)`
  - `int write(String s, int offset, int tam)`
- Outros métodos:
  - `void close()`
  - `void flush()`

Como ler um Arquivo Texto – BufferedReader

UFRPE

- Exemplo1: Preenchendo uma String com um texto lido.

```
import java.io.BufferedReader;
import java.io.FileReader;

public class LerArquivoTexto{
   public static void main(String [ ] args){
         String s, s2 = new String();
         try {
           BufferedReader in = new BufferedReader( new FileReader("texto.txt"));
           while((s = in.readLine()) != null)
              s2 += s+ "\n";
           in.close();
         }
         catch (Exception e) {
           System.err.println (e.getMessage() + "\n");
           e.printStackTrace();
         }
         System.out.println(s2);
   }
}
```

Como ler um Arquivo Texto – InputStream e BufferedReader

UFRPE

- Exemplo2: Preenchendo uma String com um texto lido.

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;;

class LerArquivoTexto2 {
    public void ler() throws Exception{
         String linha;
         InputStream is = getClass().getResourceAsStream ("textonaClasse.txt");
        BufferedReader br = new BufferedReader (new InputStreamReader (is));
        while ((linha = br.readLine()) != null)
             System.out.println (linha);
         br.close(); // br.close já fecha "is" automaticamente
    }
    public static void main(String[] args){
         LerArquivoTexto2 lat = new LerArquivoTexto2();
          try{lat.ler();}
         catch(Exception e){
              System.err.println(e.getMessage() + "\n");
              e.printStackTrace();
          }
    }
}
```

Como escrever em um Arquivo Binário

UFRPE

- Exemplo3: Escrever dados.

```
import java.io.BufferedOutputStream;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
public class EscreveArquivo{
    public static void main(String [ ] args){
            String saida="";
            try{
                        DataOutputStream out = new DataOutputStream(
                        new BufferedOutputStream(new FileOutputStream("data3.txt")));

                        //Exemplo1
                        saida = "Um exemplo de dado do tipo String: \n";
                        out.writeBytes(saida);

                        //Exemplo2
                        //saida += 3.14; //concatenando para que o texto tenha um double
                        //int a=Integer.parseInt(saida) +1;
                        //out.writeBytes(saida);

                        //Exemplo3
                        //out.writeDouble(Double.parseDouble("3.14"));

                        out.close();
            }
            catch( Exception e){ }
    }
}
```

Como ler em um Arquivo Binário

UFRPE

- Exemplo4: Ler um valor.

```
import java.io.BufferedInputStream;
import java.io.DataInputStream;
import java.io.FileInputStream;

public class LerArquivoBinario{
   public static void main(String [ ] args){
        try{
                DataInputStream in = new DataInputStream(
new BufferedInputStream( new FileInputStream("data3.txt")
        ));
                System.out.println(in.readDouble());
        }catch(Exception e){}

   }
}
```

Como ler em um Arquivo Binário

UFRPE

- Exemplo4: Ler arquivo .bin.

```
import java.io.DataInputStream;
import java.io.FileInputStream;
public class LerArquivoBinario2{
     public static void main(String [ ] args){
            try{
                        DataInputStream input  = new DataInputStream(new FileInputStream(("teste.bin")));

                        long _dbl = input.readLong();
                        int _flt = input.readInt();
                        input.close();

        _flt = ( (_flt >>> 24 & 0x000000FF)
               | (_flt >>>  8 & 0x0000FF00)
               | (_flt <<   8 & 0x00FF0000)
               | (_flt <<  24 & 0xFF000000));
        _dbl = ( (_dbl >>> 56 & 0x00000000000000FFL)
               | (_dbl >>> 40 & 0x000000000000FF00L)
               | (_dbl >>> 24 & 0x0000000000FF0000L)
               | (_dbl >>>  8 & 0x00000000FF000000L)
               | (_dbl <<   8 & 0x000000FF00000000L)
               | (_dbl <<  24 & 0x0000FF0000000000L)
               | (_dbl <<  40 & 0x00FF000000000000L)
               | (_dbl <<  56 & 0xFF00000000000000L));

                        float flt = Float.intBitsToFloat(_flt);
                        double dbl = Double.longBitsToDouble(_dbl);
                        System.out.println (flt); // imprime 2.7182817
                        System.out.println (dbl); // imprime 3.141592653589793
            }
            catch (Exception e){
                        System.out.println(e.getMessage() + "\n");
                        e.printStackTrace();
            }
     }
}
```

Como ler a partir de uma URL - Exemplo


```
import java.io.*;
import java.net.*;

public class URLReader{
    public static void main(String [ ] args){
        try{
                URL url = new URL("http://www.google.com.br");
                BufferedReader in = new BufferedReader(new
    InputStreamReader(url.openStream()));
                String inputLine;
                while ((inputLine = in.readLine()) != null) {
                        System.out.println(inputLine);
                }
        }
        catch(Exception e){
                System.out.println(e.getMessage() + "\n");
                e.printStackTrace();
        }
    }
}
```

Exercício 1

- Crie uma aplicação Java que abra, leia, mostre e copie o conteúdo de qualquer arquivo texto para outro arquivo.
- Crie uma aplicação Java que liste o conteúdo de um diretório.

Exercício 2

UFRPE

- No exercício da calculadora:
  - A calculadora poderá acumular como histórico as operações
  - Colocar uma opção para imprimir os cálculos realizados
- Projeto Ata de Frequência
  - A partir de uma aplicação Java permite criar o arquivo contendo as presenças e faltas parciais e totais, descriminada por aula, dos alunos de uma disciplina.

UFRPE

FIM

Richarlyson D'Emery
rico_demery@yahoo.com.br