Desenvolvimento de Software para Persistência

Persistência de Arquivos: texto, binário, CSV, propriedades.

Prof. Regis Pires Magalhães
regismagalhaes@ufc.br

UFC

Tipos de Arquivos

- Texto
  - Texto plano
  - Propriedades
  - CSV
  - XML
  - JSON
  - Código fonte
- Binário
  - Imagem
  - Vídeo
  - Áudio
  - Arquivo compactado
  - Código compilado: Executável / Bytecode.
  - PDF

Fluxos

- Fluxo de Entrada
  - InputStream – Ler bytes.
- Fluxo de Saída
  - OutputStream – Escrever bytes.
- Servem para operações sobre:
  - Arquivos.
  - Conexão remota via Socket.
  - Entrada e saída padrão (teclado e console).

FileInputStream

```
public class TestaEntrada {
  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("arquivo.txt");
    int b = is.read();
  }
}
```

Traduzindo de determinada codificação para unicode

```
public class TestaEntrada {
  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("arquivo.txt");
    InputStreamReader isr = new InputStreamReader(is);
    int c = isr.read();
  }
}
```

O construtor de InputStreamReader pode receber o encoding a ser utilizado como parâmetro, se desejado, tal como UTF-8 ou ISO-8859-1.

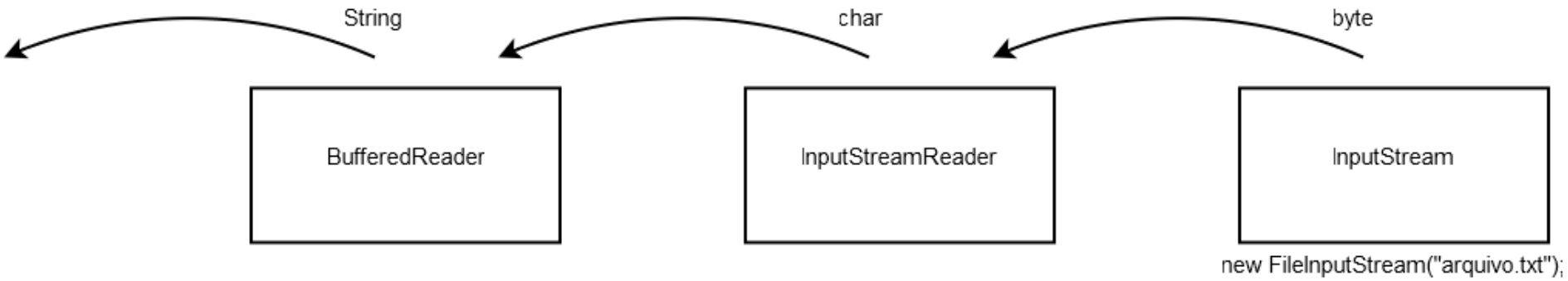
A classe abstrata Reader e descendentes manipulam chars.

Lendo uma linha inteira

```
public class TestaEntrada {
  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("arquivo.txt");
    InputStreamReader isr = new InputStreamReader(is);
    BufferedReader br = new BufferedReader(isr);
    String s = br.readLine();
  }
}
```

Lendo uma linha inteira

Leitura do Reader por pedaços, usando o Buffer, para evitar muitas chamadas ao SO.



Padrão de composição chamado Decorator Pattern.

Lendo o arquivo inteiro

```
public class TestaEntrada {
  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("arquivo.txt");
    InputStreamReader isr = new InputStreamReader(is);
    BufferedReader br = new BufferedReader(isr);
    String s = br.readLine(); // primeira linha

    while (s != null) {
      System.out.println(s);
      s = br.readLine();
    }

    br.close();
  }
}
```

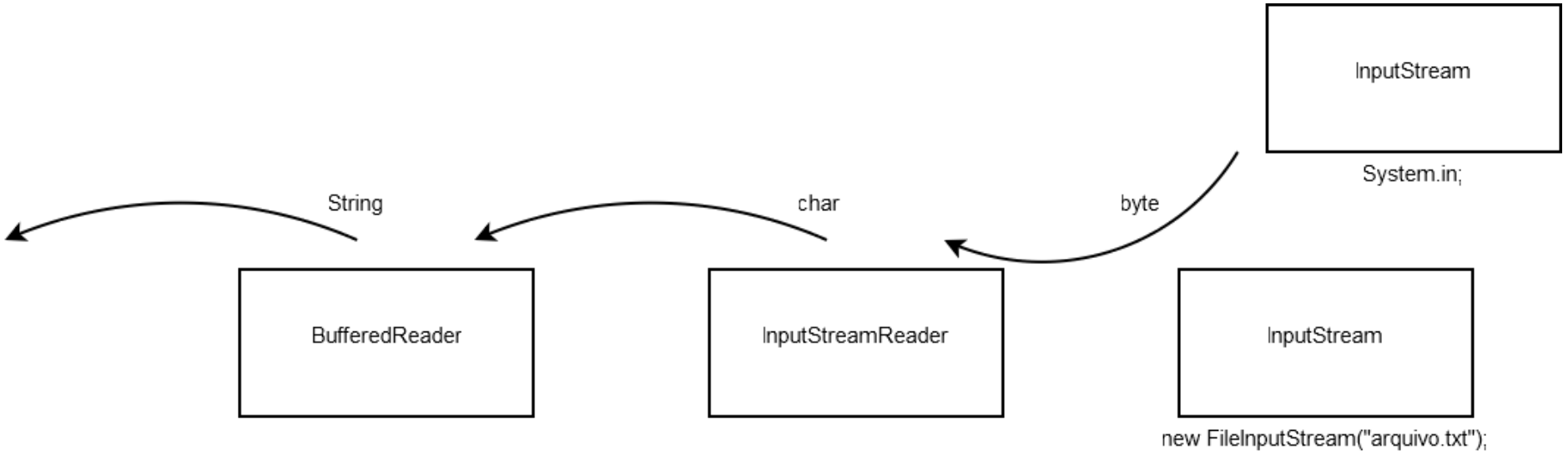
Lendo o arquivo inteiro

```
public class EntradaDeUmArquivo {
  public static void main(String[] args) throws IOException {
    InputStream is = new FileInputStream("arquivo.txt");

    Scanner entrada = new Scanner(is);
    while (entrada.hasNextLine()) {
      System.out.println(entrada.nextLine());
    }

    is.close();
  }
}
```

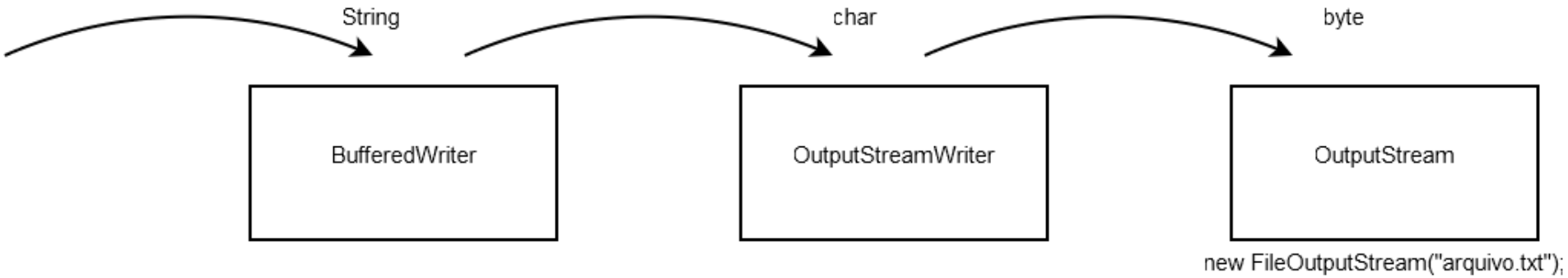
Lendo Strings do Teclado

```
public class TestaEntrada {
  public static void main(String[] args) throws IOException {
    InputStream is = System.in;
    InputStreamReader isr = new InputStreamReader(is);
    BufferedReader br = new BufferedReader(isr);
    String s = br.readLine();

    while (s != null) {
      System.out.println(s);
      s = br.readLine();
    }

  }
}
```

Lendo Strings do Teclado



Escrita em arquivo usando OutputStream



Escrita em arquivo usando OutputStream

```
public class TestaSaida {

  public static void main(String[] args) throws IOException {
    OutputStream os = new FileOutputStream("saida.txt");
    OutputStreamWriter osw = new OutputStreamWriter(os);
    BufferedWriter bw = new BufferedWriter(osw);

    bw.write("Java");
    bw.newLine();
    bw.close();
  }

}
```

FileOutputStream pode receber um booleano como segundo parâmetro, para indicar se você quer reescrever o arquivo ou manter o que já estava escrito (append).

Escrita em arquivo usando PrintStream

```
import java.io.FileNotFoundException;
import java.io.PrintStream;

public class Teste {

  public static void main(String[] args)
                                throws FileNotFoundException {

    PrintStream ps = new PrintStream("saida.txt");
    ps.println("Java");
    ps.close();
  }

}
```

Lendo Strings do Teclado e salvando em um arquivo

```
...
Scanner s = new Scanner(System.in);
PrintStream ps = new PrintStream("arquivo.txt");
while (s.hasNextLine()) {
  ps.println(s.nextLine());
}
...
```

Fechamento do arquivo no finally

No Java 7 a estrutura try-with-resources executará o close automático dos recursos declarados no try(), que implementam a interface java.lang.AutoCloseable (Readers, Writers, Streams).

```
try (BufferedReader br = new BufferedReader(new File("arquivo.txt"))) {
  // com exceção ou nao, o close() do br sera invocado
}
```

CSV - Comma-separated values

- O formato CSV é bastante simples e suportado por quase todas as planilhas eletrônicas e SGDB disponíveis no mercado.

CSV - Comma-separated values

Year	Make	Model	Description	Price
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""","",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

TSV - Tab-separated values

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa

```
Sepal length Sepal width Petal length Petal width Species
5.1 3.5 1.4 0.2 I. setosa
4.9 3.0 1.4 0.2 I. setosa
4.7 3.2 1.3 0.2 I. setosa
4.6 3.1 1.5 0.2 I. setosa
5.0 3.6 1.4 0.2 I. setosa
```

Planilhas

- Excel
- Google Planilha

Desenvolvimento de Software Low-Code / No-Code usando Planilhas:

- AppSheet do Google - https://www.appsheet.com/
- Bubble - https://bubble.io/
- Glide - https://www.glideapps.com/

Arquivos de propriedades

- Um mapa importante é a tradicional classe Properties, que mapeia strings e é muito utilizada para a configuração de aplicações.

- A Properties possui, também, métodos para ler e gravar o mapeamento com base em um arquivo texto, facilitando muito a sua persistência.

Arquivos de propriedades

```
Properties config = new Properties();
config.setProperty("database.login", "scott");
config.setProperty("database.password", "tiger");
config.setProperty("database.url","jdbc:mysql:/localhost/teste");

// muitas linhas depois...

String login = config.getProperty("database.login");
String password = config.getProperty("database.password");
String url = config.getProperty("database.url");
DriverManager.getConnection(url, login, password);
```

Não há necessidade de casting para String

Arquivos de propriedades - gravação

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Properties;

public class App {
  public static void main(String[] args) {
    Properties prop = new Properties();

    try {
      prop.setProperty("database", "localhost");
      prop.setProperty("dbuser", "mkyong");
      prop.setProperty("dbpassword", "password");

      prop.store(new FileOutputStream("config.properties"),
                 null);
    } catch (IOException ex) {
      ex.printStackTrace();
    }
  }
}
```

Arquivos de propriedades - leitura

```
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class App {
  public static void main(String[] args) {
    Properties prop = new Properties();

    try {
      prop.load(new FileInputStream("config.properties"));
      // get the property value and print it out
      System.out.println(prop.getProperty("database"));
      System.out.println(prop.getProperty("dbuser"));
      System.out.println(prop.getProperty("dbpassword"));

    } catch (IOException ex) {
      ex.printStackTrace();
    }
  }
}
```

Arquivos de propriedades - leitura

```
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class App {
  public static void main(String[] args) {
    Properties prop = new Properties();

    try {
      prop.load(App.class.getClassLoader()
                .getResourceAsStream("config.properties"));

      System.out.println(prop.getProperty("database"));
      System.out.println(prop.getProperty("dbuser"));
      System.out.println(prop.getProperty("dbpassword"));
    } catch (IOException ex) {
      ex.printStackTrace();
    }
  }
}
```

getResourceAsStream

Method	Parameter format	Lookup failure behavior	Usage example
`ClassLoader.getResourceAsStream()`	"/"-separated names; no leading "/" (all names are absolute)	Silent (returns `null`)	`this.getClass().getClassLoader().getResourceAsStream("some/pkg/resource.properties")`
`Class.getResourceAsStream()`	"/"-separated names; leading "/" indicates absolute names; all other names are relative to the class's package	Silent (returns `null`)	`this.getClass().getResourceAsStream("resource.properties")`
`ResourceBundle.getBundle()`	"."-separated names; all names are absolute; `.properties` suffix is implied	Throws unchecked `java.util.MissingResourceException`	`ResourceBundle.getBundle("some.pkg.resource")`

Manipulação de grandes arquivos

- `wc -l big_file.csv`
- `head –n 3 big_file.csv`
- `tail -n 3 big_file.csv`
- `cat big_file.csv | less`
- `cat big_file.csv | grep "52.2479 21.0191"`
- `sed "s/ /;/g" teste.txt`
- `awk '{ print $4 "," $5 }'  teste.txt`
- `awk -F "," '{ print $4 "," $5 }'  teste.csv`

Checksum / Hash de grandes arquivos

- Checksum
  - cksum teste*.*
- MD5
  - md5sum teste*.* > md5.txt
  - md5sum -c md5.txt
- SHA1
  - sha1sum teste*.* > sha1.txt
  - sha1sum -c sha1.txt

Armazenando vários arquivos em um só arquivo (arquivamento)

- tar
  - `tar cfv teste.tar teste*.*`
    - `c – create`
    - `f – file`
    - `v - verbose`

Compressão de arquivos

- Sem perda
  - zip, rar, 7z, gzip (um só arquivo), bzip2 (um só arquivo),...
    - `zip teste.zip teste*.*`
    - `gzip teste.txt`
    - `bzip2 teste.txt`
    - `tar c teste*.* | gzip > teste.tar.gz`
    - `tar cfvz teste.tar.gz teste*.*`
      - `z - gzip`
    - `tar cfvj teste.tar.bz teste*.*`
      - `j – bzip2`
- Com perda
  - jpg,mp3, mp4, ...

Encriptação de arquivos

- Encriptar:
  - `gpg -c teste.txt`
- Decriptar:
  - `gpg teste.txt.gpg`

http://www.techrepublic.com/article/how-to-easily-encryptdecrypt-a-file-in-linux-with-gpg/

http://irtfweb.ifa.hawaii.edu/~lockhart/gpg/

https://help.ubuntu.com/community/GnuPrivacyGuardHowto

Leitura de arquivo ZIP

```
package files;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.zip.ZipInputStream;

public class LeArquivoZip {

    public static void main(String[] args) throws Exception {
        InputStream is = new FileInputStream("/Users/regis/teste2.zip");
        ZipInputStream zis = new ZipInputStream(is);
        zis.getNextEntry();
        InputStreamReader isr = new InputStreamReader(zis);
        BufferedReader br = new BufferedReader(isr);
        String str;
        while ((str = br.readLine()) != null) {
            System.out.println(str);
        }
        br.close();
    }
}
```

Leitura de arquivo ZIP

```
package files;

import java.io.FileInputStream;
import java.io.InputStream;
import java.util.Scanner;
import java.util.zip.ZipInputStream;

public class LeArquivoZip2 {

    public static void main(String[] args) throws Exception {
        InputStream is = new FileInputStream("/Users/regis/teste2.zip");
        ZipInputStream zis = new ZipInputStream(is);
        zis.getNextEntry();
        Scanner sc = new Scanner(zis);
        while (sc.hasNextLine()) {
            System.out.println(sc.nextLine());
        }
        sc.close();
    }
}
```

Referências

- Apostila FJ11 da Caelum: Java e Orientação a Objetos.
  - Capítulo 15: Pacote java.io.
  - Capítulo 16: Collections framework.
- Java Properties File Examples
  - http://www.mkyong.com/java/java-properties-file-examples/
- Smartly load your properties
  - http://www.javaworld.com/javaworld/javaqa/2003-08/01-qa-0808-property.html

Obrigado!

Dúvidas, comentários, sugestões?

Regis Pires Magalhães
regispires@lia.ufc.br
@regispires

UFC