

Arquivos em Java

Técnicas de Programação

Prof. Iális Cavalcante

Engenharia da Computação

2015.2



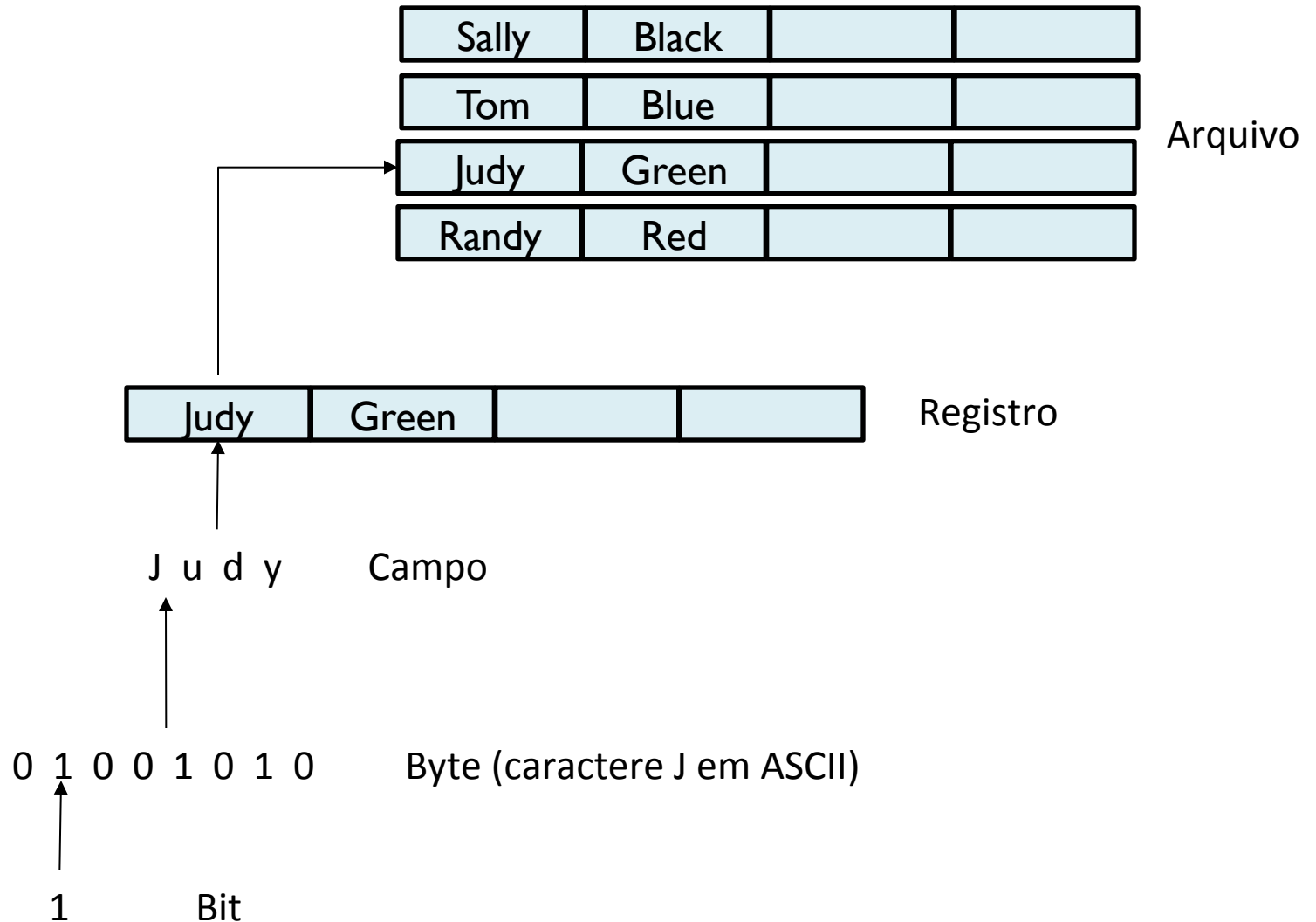
Arquivos

- Um arquivo é uma abstração utilizada para uniformizar a interação entre o ambiente de execução e os dispositivos externos.
- A interação de um programa com um dispositivo através de arquivos passa por três etapas:
 - abertura ou criação de um arquivo
 - transferência de dados
 - fechamento do arquivo

Arquivos: Binários x Textos

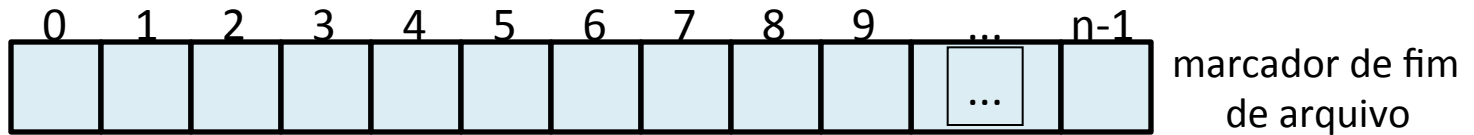
- O conteúdo de um Arquivo Binário é uma sequência de bytes (8 bits). Os arquivos binários podem conter imagens, sons, versões comprimidas de outros arquivos, etc...
- Um arquivo texto é normalmente formado por caracteres ASCII.

Hierarquia de Dados



Arquivos e fluxos

- Java vê cada arquivo como um fluxo seqüencial de bytes:

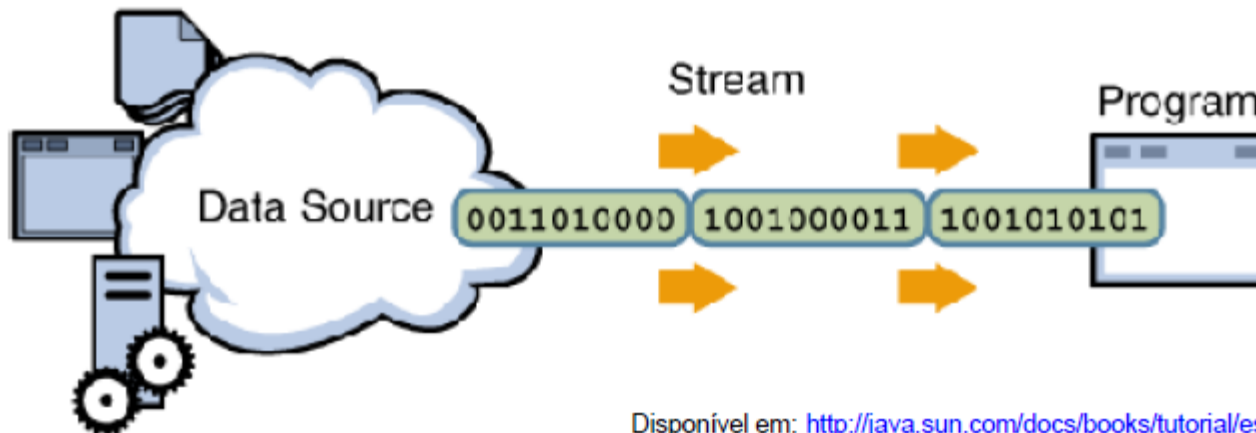


- Abre-se um arquivo através da criação de objetos e a associação de um fluxo de dados a este objeto.
- Processamento de arquivos: classes do pacote ***java.io***.

Arquivos e Fluxos

- Em Java utiliza-se: “Stream”
- Fluxo de dados (I/O), agrupados em bytes (8 bits) - `ByteStream`.
- É um objeto que “entrega” dados a um destino (dispositivos de saída: screen, file, etc) ou “recebe” dados de uma fonte (dispositivos de entrada: keyboard, file, etc.).
- Atua como um “buffer” entre a fonte de dados e o destino .

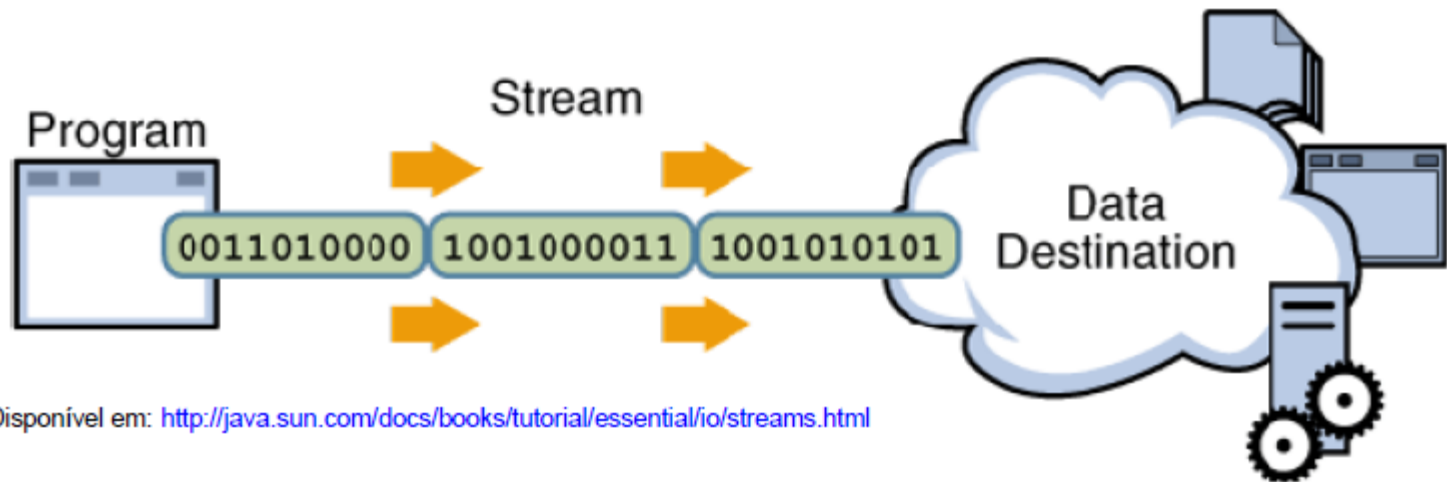
Input Stream



Disponível em: <http://java.sun.com/docs/books/tutorial/essential/io/streams.html>

Entrada de dados de uma fonte para o programa (aplicativo)

Output Stream



Disponível em: <http://java.sun.com/docs/books/tutorial/essential/io/streams.html>

Arquivos e fluxos

- Classes oriundas das funções de *System.in* e *System.out*;
- `FileInputStream`, `FileOutputStream` (bytes)
 - Entrada/saída de bytes de/para um arquivo;
- `FileReader`, `FileWriter` (caracteres)
 - Entrada/saída de caracteres de/para um arquivo;
- `File`
 - Descrição do arquivo

Arquivos e fluxos

- Criar um caminho usando:
 - `java.io.File`
- Ler um arquivo usando:
 - `java.io.FileReader`
 - `java.io.BufferedReader`
- Escrever em um arquivo usando:
 - `java.io.FileWrite`
 - `java.io.BufferedWriter`

Classe File

- Criando um caminho:
 - `File file=new File("/usr/tecnicas/arquivo.txt");`
 - `File noDiretorioAtual=new File("arquivo.txt");`
 - `File noDiretorioAnterior=new File("../arquivo.txt");`

Classe File

- Métodos importantes:
 - Descobrir para onde a instancia de **File** aponta:
 - `boolean exists();`
 - `boolean isDirectory();`
 - `boolean isFile();`
 - Checar a viabilidade de leitura e gravação:
 - `boolean canRead();`
 - `boolean canVWrite();`

Classe File

- Criar um ou mais diretórios:
 - `boolean mkdir();`
 - `boolean mkdirs();`
- Outros:
 - `void close();`
 - `boolean renameTo(File file);`
 - `long length();`
 - `long lastModified();`
 - `boolean delete();`
- Exemplo no **NetBeans**

Leitura de um Arquivo

- Classe FileReader
 - `File file = new File("arquivo.txt");`
 - `FileReader reader = new FileReader(file);`
 - `FileReader reader = new FileReader(new File("arquivo.txt"));`
 - `FileReader reader = new FileReader("arquivo.txt");`

Leitura de um Arquivo

- Classe `BufferedReader`
 - `File file = new File("arquivo.txt");`
 - `FileReader reader = new FileReader(file);`
 - `BufferedReader leitor = new BufferedReader(reader);`
 - `BufferedReader leitor = new BufferedReader(new FileReader(new File("arquivo.txt")));`

Leitura de um Arquivo

- Métodos importantes:
 - `String readLine();`
 - `void close();`

Exemplo de Leitura

- Lendo:

```
String linha = null;  
while((linha = leitor.readLine()) != null){  
    System.out.println(linha);  
}
```

Exemplo no **NetBeans**

Escrita em um Arquivo

- Classe FileWriter

- `File file = new File("arquivo.txt");`
- `FileWriter writer = new FileWriter(file);`
- `FileWriter writer = new FileWriter(new File("arquivo.txt"));`
- `FileWriter writer = new FileWriter("arquivo.txt");`
- `FileWriter writer = new FileWriter("arquivo.txt", true);`

Escrita em um Arquivo

- Classe `BufferedWriter`
 - `File file = new File("arquivo.txt");`
 - `FileWriter writer = new FileWriter(file);`
 - `BufferedWriter escritor = new BufferedWriter(writer);`
 - `BufferedWriter escritor = new BufferedWriter(new FileWriter(new File("arquivo.txt")));`

Escrita em um Arquivo

- Métodos importantes:
 - `void write(String str);`
 - `void newLine();`
 - `void close();`

Exemplo de Escrita

- Escrevendo:

```
for(int i=1; i<10; i++){  
    escritor.write("Escrevendo na linha "+i);  
    escritor.newLine();  
}
```

Exemplo no **NetBeans**

Exemplo I

```
import java.io.Serializable;
```

```
public class Computador implements  
    Serializable {
```

```
    String nome;
```

```
    String velocidade;
```

```
    String memoria;
```

```
    String revendedor;
```

```
}
```

Exemplo I

```
import java.io.FileOutputStream;  
import java.io.ObjectOutputStream;
```

```
public class GravaObjeto {  
  
    public static void main(String args[]) {  
        Computador c1 = new Computador();  
        c1.nome = "Pentium 4";  
        c1.velocidade = "1,5 GigaHertz";  
        c1.memoria = "256 MegaBytes";  
        c1.revendedor = "Novo Pc";  
        try {  
            FileOutputStream f = new FileOutputStream ("computador.arq");  
            ObjectOutputStream s = new ObjectOutputStream(f);  
            s.writeObject(c1);  
            s.flush();  
            System.out.println("Gravacao realizada com sucesso!");  
        }catch (Exception e){  
            System.out.println(e);  
        }  
    }  
}
```

Exemplo I

```
import java.io.FileInputStream;  
import java.io.ObjectInputStream;
```

```
public class LeObjeto {  
  
    public static void main(String args[]) {  
        try {  
            FileInputStream f = new FileInputStream("computador.arq");  
            ObjectInputStream s = new ObjectInputStream(f);  
            Computador x = (Computador)s.readObject();  
            if (x.nome=="Pentium 4")  
                System.out.println("OK");  
        }catch (Exception e) {  
            System.out.println(e);  
        }  
    }  
}
```


Exemplo 2

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

```
public class Arquivos {
```

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

```
        File file = new File("C:\\teste.txt");
```

```
        FileReader reader = new FileReader(file);
```

```
        BufferedReader leitor = new BufferedReader(reader);
```

```
        String linha;
```

```
        do {
```

```
            linha = leitor.readLine();
```

```
            if (linha != null)
```

```
                System.out.println(linha);
```

```
        } while (linha != null);
```

```
        leitor.close();
```

```
        reader.close();
```

```
    }
```

```
}
```

```
File selectedFile = fileChooser.getSelectedFile();
```

```
String path = selectedFile.getParent()+"\n"+selectedFile.getName();
```