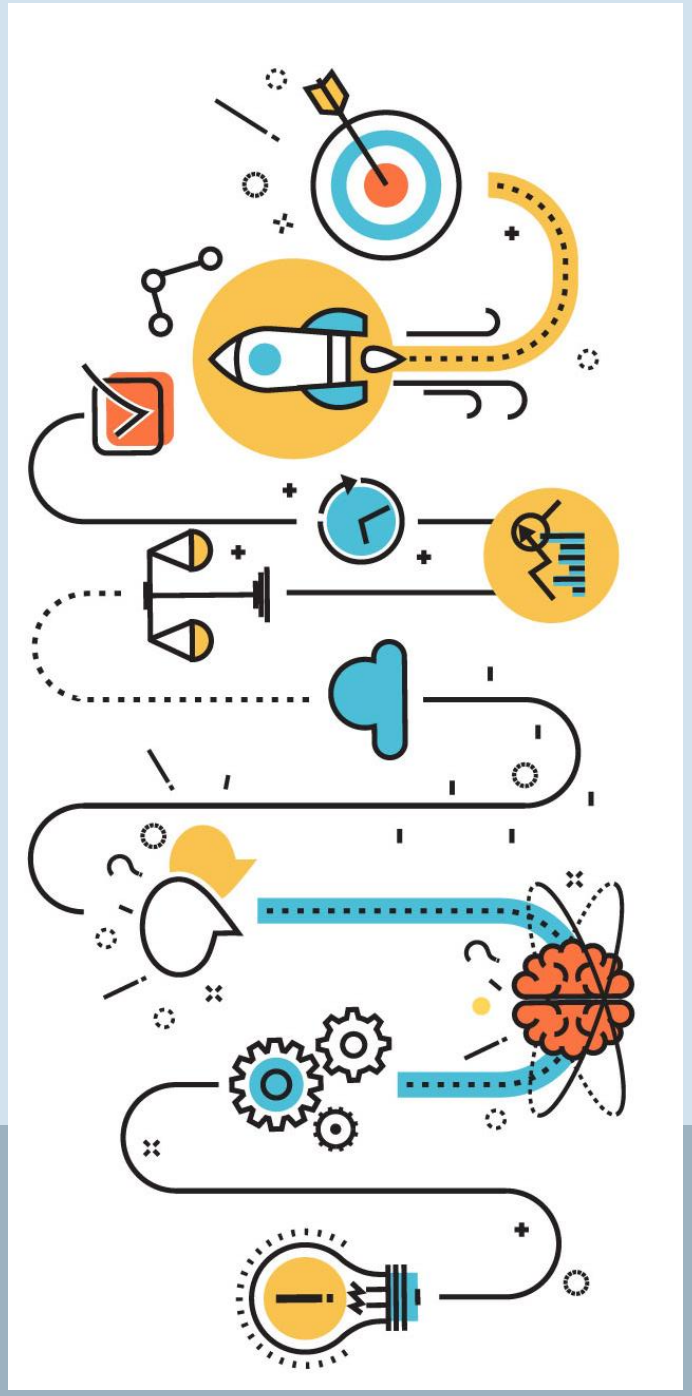


Leitura e Gravação de Arquivos de Textos

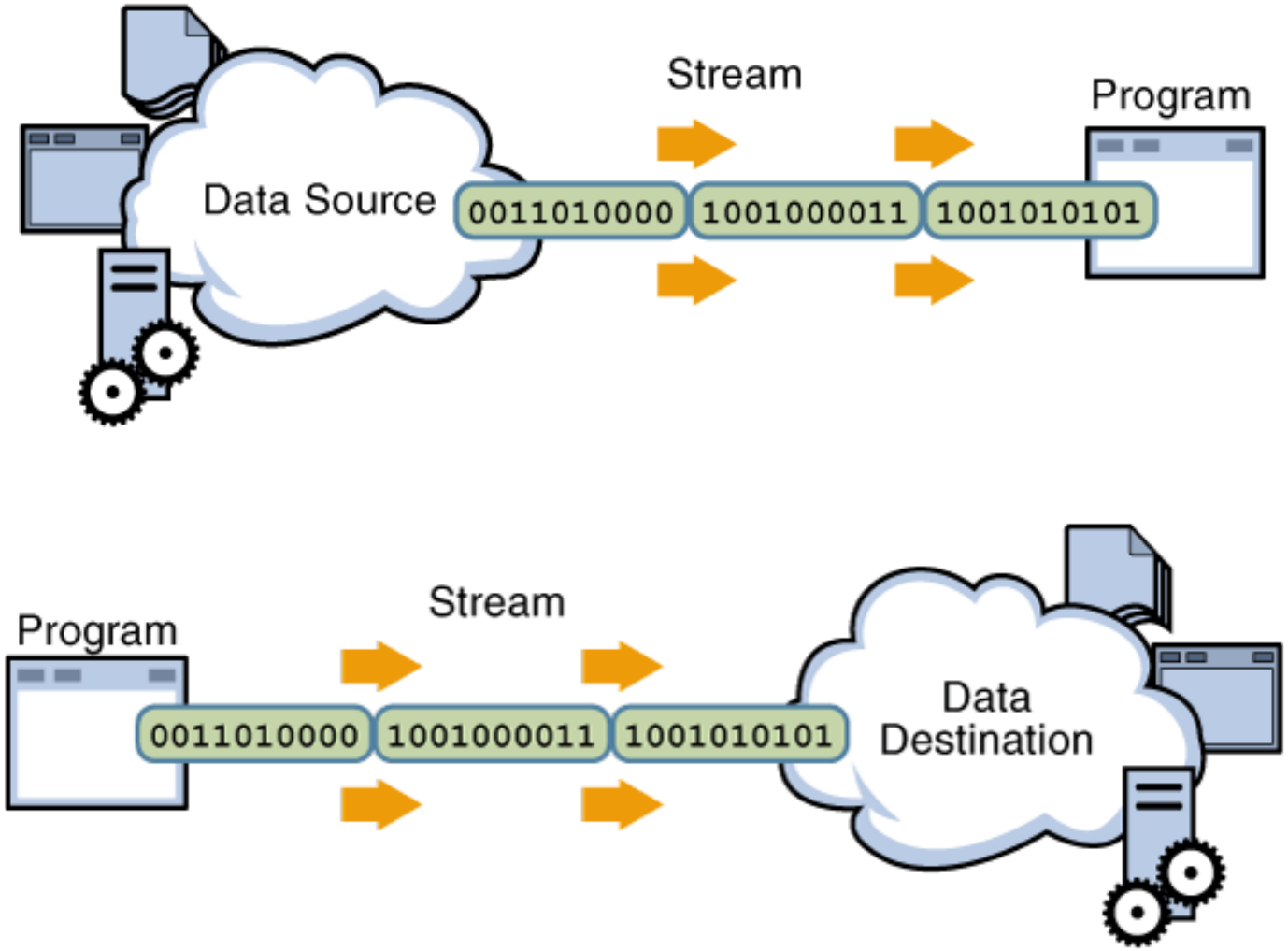
Prof. Rhafael Freitas da Costa, M. Sc.

Desenvolvimento de Software

Introdução

- Arquivos de texto (cenário mais comum)
- Existem outras formas de leitura (arquivos binários, por exemplo)
- Hoje vamos focar nos arquivos de texto
- Manipulação de arquivos

Leitura e gravação em baixo nível



String e seus métodos (vetor de caracteres)

- String é uma “classe” amplamente utilizada na linguagem Java para manipular textos que são armazenados na memória do computador como uma vetor de caracteres;
- .length( ), .charAt( ), .subString( ).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	r	o	g	r	a	m	a	ç	ã	o		J	a	v	a

Criando diretórios e arquivos

- Para a criação de diretórios e arquivos em Java, podemos utilizar a classe java.io.File.

- Para criar um diretório devemos primeiro criar uma instância da classe File para criar um objeto File. Em seu construtor passamos o local e o nome do diretório que será criado.

Criando Diretorio

```
import java.io.File;

// Criando o objeto File

File diretorio = new File("C:\\TutorialArquivos");

// Criando o diretorio

boolean statusDir = file.mkdir();

System.out.print(statusDir);
```

Criando Arquivos

```
import java.io.File;

// Criando o objeto File

File diretorio = new File(diretório, “teste.txt”);

// Criando o arquivo

boolean statusArq = arquivo.createNewFile();

System.out.print(statusArq);
```

Métodos da Classe File

```
// Verificando se o diretório existe
if (dir.exists()) {
   System.out.println("Diretório existe!");
   // verificando se o arquivos existe
   if (arq.exists()) {
      System.out.println("Arquivo existe!");
   }
}
```

Métodos da Classe File

```
// Criando o objeto do tipo File

File files = new File("C:\\");

//Percorrendo a lista de arquivos

for (File file : files.listFiles()) {

  System.out.println(file);

}
```

Métodos da Classe File

```
// Criando o diretorio
File dir = new File("C:\\TutorialArquivos");
// Criando os arquivos
File arq = new File(dir, "arq_01.txt");
File arq2 = new File(dir, "arq_02.txt");
// Alterando o nome do arquivo
boolean statusRename = arq.renameTo(arq2);
// Imprimindo no console
System.out.println("Renomedo: " + statusRename + " -> New name: " + arq);}
```

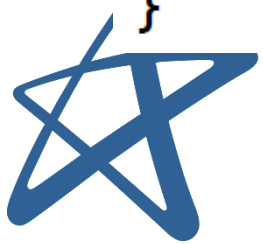
Métodos da Classe File

```
// Excluindo o diretório
System.out.println( dir3.delete() );

// Excluindo o arquivo
System.out.println( arq3.delete() );
```

Escrevendo

Como gravar arquivos de dados? (Opção 1)

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class Programa {
    public static void main(String[ ] args) throws IOException {
        FileWriter arquivo = new FileWriter("C:\\_ws\\numeros.txt");
        PrintWriter gravador = new PrintWriter(arquivo);
        for (int i = 1; i <= 10; i++) {
            System.out.println("Gravando número: " + i);
            gravador.println("Número: " + i);
        }
    }
}
```



Como gravar arquivos de dados? (Opção 2)

```
import java.io.IOException;
import java.util.Formatter;

public class Programa {
    public static void main(String[ ] args) throws IOException {
        Formatter gravador = new Formatter("C:\\_ws\\numeros.txt");
        for (int i = 1; i <= 10; i++) {
            System.out.println("Gravando número: " + i);
            gravador.format("Número: " + i);
        }
        gravador.close();
    }
}
```

Criando Arquivos

```
File dir = new File("C:\\TutorialArquivos");
File arq = new File(dir, "User.txt");
arq.createNewFile();
// Esse construtor aceita dois tipos de parâmetros, File ou String.
// O parâmetro true indica que reescrevemos no arquivo sem apagar o que já existe.
// O false apagaria o conteúdo do arquivo e escreveria o novo conteúdo.
FileWriter fileWriter = new FileWriter(arq, false);
//Agora vamos usar a classe PrintWriter para escrever fisicamente no arquivo.
//Precisamos passar o objeto FileReader em seu construtor
PrintWriter printWriter = new PrintWriter(fileWriter);
printWriter.println(user.getId());
printWriter.println(user.getNome());
printWriter.println(user.getObservacao());
//o método flush libera a escrita no arquivo
printWriter.flush();
//No final precisamos fechar o arquivo
printWriter.close();
```

Exercício

- Escreva um programa Java que crie um arquivo chamado "numeros.txt" e escreva nele os números de 1 a 10, um por linha.

Exercício

- Escreva um programa Java que crie um arquivo chamado "informacoes.txt" e escreva nele algumas informações sobre o Livro: ISBN, Titulo, Autor, Gênero e Número de Paginas.
- Esses dados devem ser informados pelo usuário;
- Usuário poderá cadastrar mais de um Livro.

Lendo

Manipulação de arquivos

- Para ler um arquivo, vamos precisar de duas classes auxiliares fornecidas pelo Java:
- FileReader: faz a leitura de arquivos de texto, gerando um "fluxo" de caracteres
- BufferedReader: retira do fluxo de caracteres as informações relevantes que nos interessam, armazenando-as em um "buffer" (espaço de memória)

String e seus métodos (vetor de caracteres)

```
File dir = new File("C:\\TutorialArquivos");
File arq = new File(dir, "User.txt");
// Indicamos o arquivo que será lido
FileReader fileReader = new FileReader(arq);
// Criamos o objeto bufferReader que nos
// oferece o método de leitura readLine()
BufferedReader bufferedReader = new BufferedReader(fileReader);
// String que irá receber cada linha do arquivo
String linha = "";
// Fazemos um loop linha a linha no arquivo,
// enquanto ele seja diferente de null.
// O método readLine() devolve a linha na
// posicao do loop para a variavel linha.
while ( ( linha = bufferedReader.readLine() ) != null) {
   //Aqui imprimimos a linha
   System.out.println(linha);
}
//liberamos o fluxo dos objetos ou fechamos o arquivo
fileReader.close();
bufferedReader.close();
```

Como ler arquivos de texto?

```java
import java.util.Scanner;

public class Programa {
    public static void main(String[ ] args) throws FileNotFoundException {
      File arquivo = new File("C:\\_ws\\arquivo.txt");
      Scanner leitor = new Scanner(arquivo);

      while (leitor.hasNext()) {
       System.out.println(leitor.nextLine());
     }
    }
}
```

Como ler textos da web?

```
import java.util.Scanner;

public class Programa {
    public static void main(String[ ] args) throws IOException {
      URL url = new URL("http://.../arquivo.txt");
      Scanner leitor = new Scanner(url.openStream());

      while (leitor.hasNext()) {
        System.out.println(leitor.nextLine());
      }
      leitor.close();
    }
}
```

Exercícios

Métodos

- Refatore o exercício 1 para permitir que o usuário armazene os dados inseridos em um arquivo. Além disso, adicione a funcionalidade para que o usuário possa ler os dados do arquivo e calcular a nota final.

OBRIGADO!

Desenvolvimento de Software

Professor Rhafael Freitas
da Costa, M. Sc.