Linguagem Java

Adenilso da Silva Simão

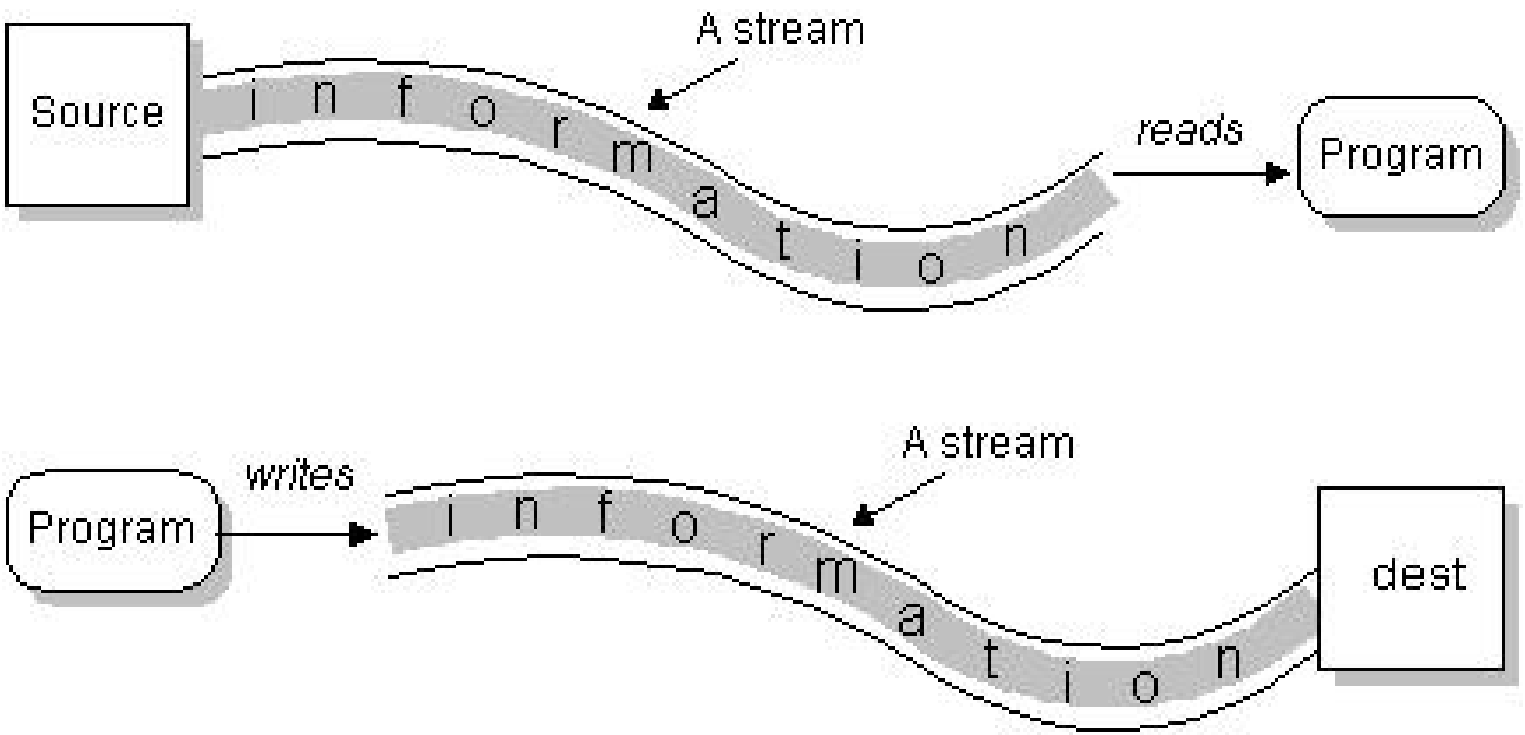
01/09/05

Introdução

- Onde conseguir informações sobre os pacotes:
- `http://java.sun.com/j2se/1.5.0/docs/api/`
- Tutorial
- `http://java.sun.com/docs/books/tutorial/essential/io/overview.html`

Introdução (II)

- Tanto a entrada quanto a saída são consideradas fluxos



Introdução (III)

- Para cada fonte de entrada, deve-se designar um fluxo
  - Teclado
  - Arquivo
  - Modem
  - Rede
- O mesmo vale para a saída

Introdução (IV)

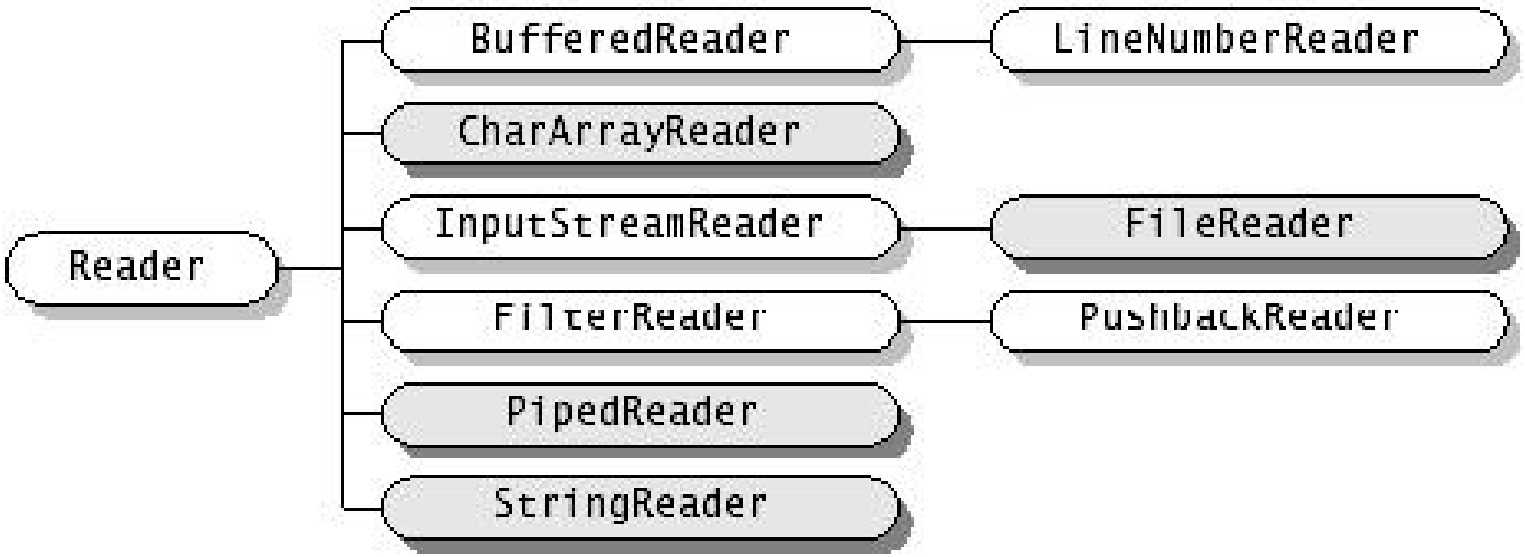
- Em java, existem duas categorias de fluxos
  - Fluxos de caracteres
    - Lida com caracteres unicode 16 bits
  - Fluxos de `bytes`
    - Lida com `bytes` de 8 *bits*

Fluxos de Caracteres

- A maioria dos aplicativos usam informações textuais
  - Fluxos de caracteres são mais adequados
- Unicode
  - Super tabela de caracteres
- Duas classes básicas
  - `Reader`
    - Ler informações
  - `Writer`
    - Escrever informações

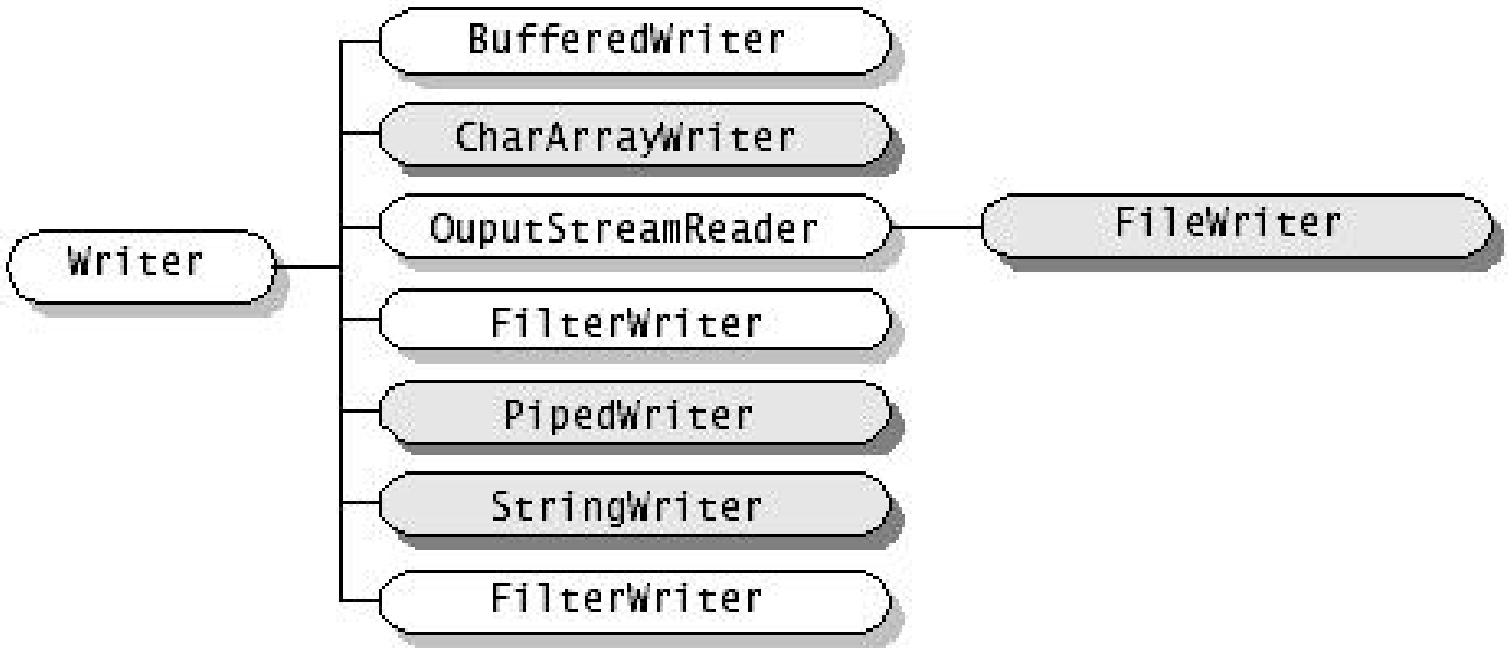
Readers

- Existem vários tipos de `Readers`
  - Alguns simplesmente usam a fonte de dados
    - Aparecem em cinza
  - Alguns fornecem algum processamento
    - Aparecem em branco
-



Writers

- De forma análoga, existem diversos tipos de `Writer`s
  - Observe que nome da `OutputStreamWriter` está trocado por engano



Exemplos

- Vamos criar uma classe que escreve em um arquivo

```
1  import java.io.*;
2
3  class UmaClasse {
4    public void f(String filename) {
5      FileWriter fw;
6      try {
7        fw = new FileWriter(filename);
8        fw.write("Nao tinha medo o tal Joao de Santo Cristo\n");
9        fw.write("Era o que todos diziam quando ele se perdeu\n");
10       fw.write("Deixou pra tras todo o marasmo da fazenda\n");
11       fw.close();
12     }
13     catch (IOException e) {
14       System.err.println("Arquivo nao encontrado!!");
15     }
16   }
17 }
```

Exemplos (II)

- Para usar o `println`?
  - Use um `PrintWriter`

```
1  import java.io.*;
2
3  class UmaClasse {
4    public void f(String filename) {
5      PrintWriter fw;
6      try {
7        fw = new PrintWriter(new FileWriter(filename));
8        fw.println(5);
9        fw.close();
10     }
11     catch (IOException e) {
12       System.err.println("Arquivo nao encontrado!!");
13     }
14   }
15 }
```

Exemplos (III)

- Para escrever na saída padrão?

```
1 import java.io.*;
2
3 class UmaClasse {
4   public void f() {
5       System.out.println("Nao tinha medo o tal Joao de Santo Cristo");
6       System.out.println("Era o que todos diziam quando ele se perdeu");
7       System.out.println("Deixou pra tras todo o marasmo da fazenda");
8   }
9 }
```

Exemplos (IV)

- Uma `f` mais genérica

```
1  import java.io.*;
2
3  class UmaClasse {
4    public void f(Writer fw) {
5      try {
6        fw.write("Nao tinha medo o tal Joao de Santo Cristo\n");
7        fw.write("Era o que todos diziam quando ele se perdeu\n");
8        fw.write("Deixou pra tras todo o marasmo da fazenda\n");
9        fw.flush();
10     }
11     catch (IOException e) {
12       System.err.println("Problemas ao escrever!!");
13     }
14   }
```

Exemplos (V)

- Usando `f` para escrever na saída padrão

```
1        OutputStreamWriter osw =
2            new OutputStreamWriter(System.out);
3        f(osw);
```

Exemplos (VI)

- Usando `f` para escrever em uma arquivo

```
1        FileWriter fw = new FileWriter("data.dat");
2        f(fw);
```

Exemplos (VII)

- Usando `f` para escrever em uma string

```
1        StringWriter sw = new StringWriter();
2        f(sw);
3        String str = sw.toString();
```

Exemplos (VIII)

- Como ler de um fluxo?

```
1  import java.io.*;
2
3  class UmaClasse {
4    void f() {
5      FileReader fr = null;
6      try {
7        fr = new FileReader("data.dat");
8        int r;
9        while((r = fr.read()) != -1) {
10         // Faz algo com o caracter!
11       }
12     }
13     catch (IOException e) {
14       // Tratar a excecao!
15     }
16   }
17 }
```

Exemplos (IX)

- Como ler uma linha inteira?
  - BufferedReader

```
1  import java.io.*;
2
3  class UmaClasse {
4    public static void main(String args[]) {
5      BufferedReader br = null;
6      try {
7        br = new BufferedReader(new FileReader("data.dat"));
8        String line;
9        while((line = br.readLine()) != null) {
10         System.out.println(line);
11       }
12     }
13     catch (IOException e) {
14       System.err.println(e);
15     }
16   }
17 }
```

Fluxo de `bytes`

- Conceitos similares se aplicam aos fluxos de `bytes`
  - `InputStream`
  - `OutputStream`

Cópia de Fluxos

- Defina um método que copia o conteúdo de um fluxo de caracteres para outro
  - Deve ser usado da seguinte forma

```
1    streamCopy(r, w);
```

  - Sendo que `r` é um `Reader` e `w` é um `Writer`

Lista de Arquivos

- Crie um tipo de fluxo de entrada que é construído com um `array` de *string*s
  - Cada *string* é o nome de um arquivo
  - Quando for solicitado, o fluxo deve ler os caracteres dos arquivos da lista, na ordem que aparecem

Lista de Arquivos (II)

- Exemplo de uso

```
1   String filenames[] = {"data1.dat", "data2.dat",
2     "data3.dat"};
3
4   MyFileListReader mflr = new MyFileListReader(filenames);
5   try {
6     int r;
7     while ((r = mflr.read()) != -1) {
8       // Faca alguma coisa
9     }
10  }
11  catch (IOException e) {
12    // Tratar a excecao!!
13  }
```