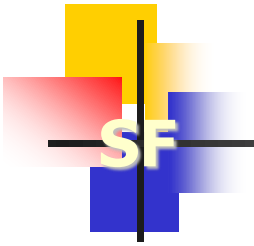


Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Programação Orientada a Objetos

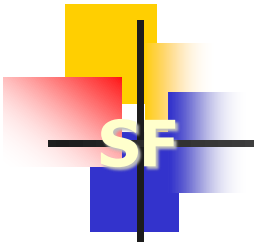
Semana 10
Manipulação de Arquivos

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Reflexão

"Seu trabalho vai preencher boa parte da sua vida e a única maneira de ser verdadeiramente satisfeito é fazer o que acredita ser um ótimo trabalho. E a única maneira de fazer um ótimo trabalho é amar o que você faz."

Steve Jobs



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Arquivos

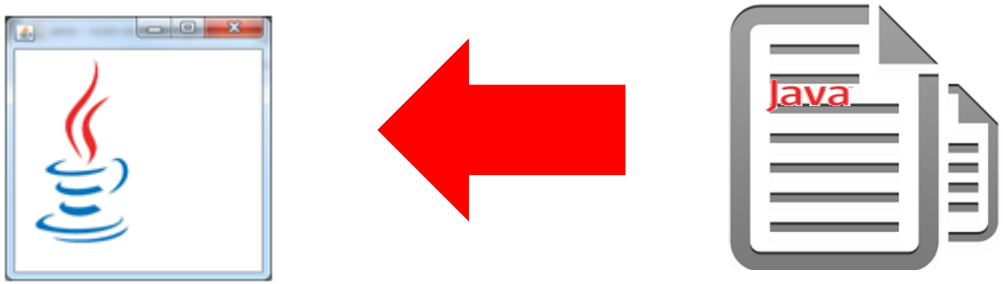
Uma solução para recuperar dados obtidos em uma execução anterior do programa é armazenar esses dados em disco na forma de arquivos.

Os arquivos permitem o armazenamento de dados por um período de tempo indeterminado, em uma memória não-volátil, de forma independente do programa que o manipula.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

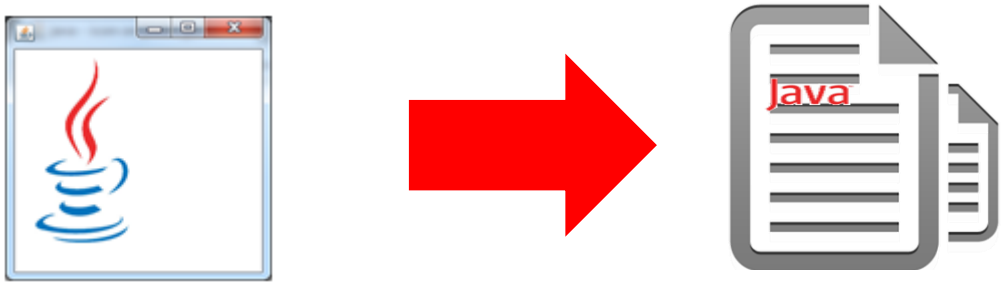
Arquivos

Leitura: a aplicação lê dados do arquivo e armazena em uma variável.



Programa Java **Arquivo**

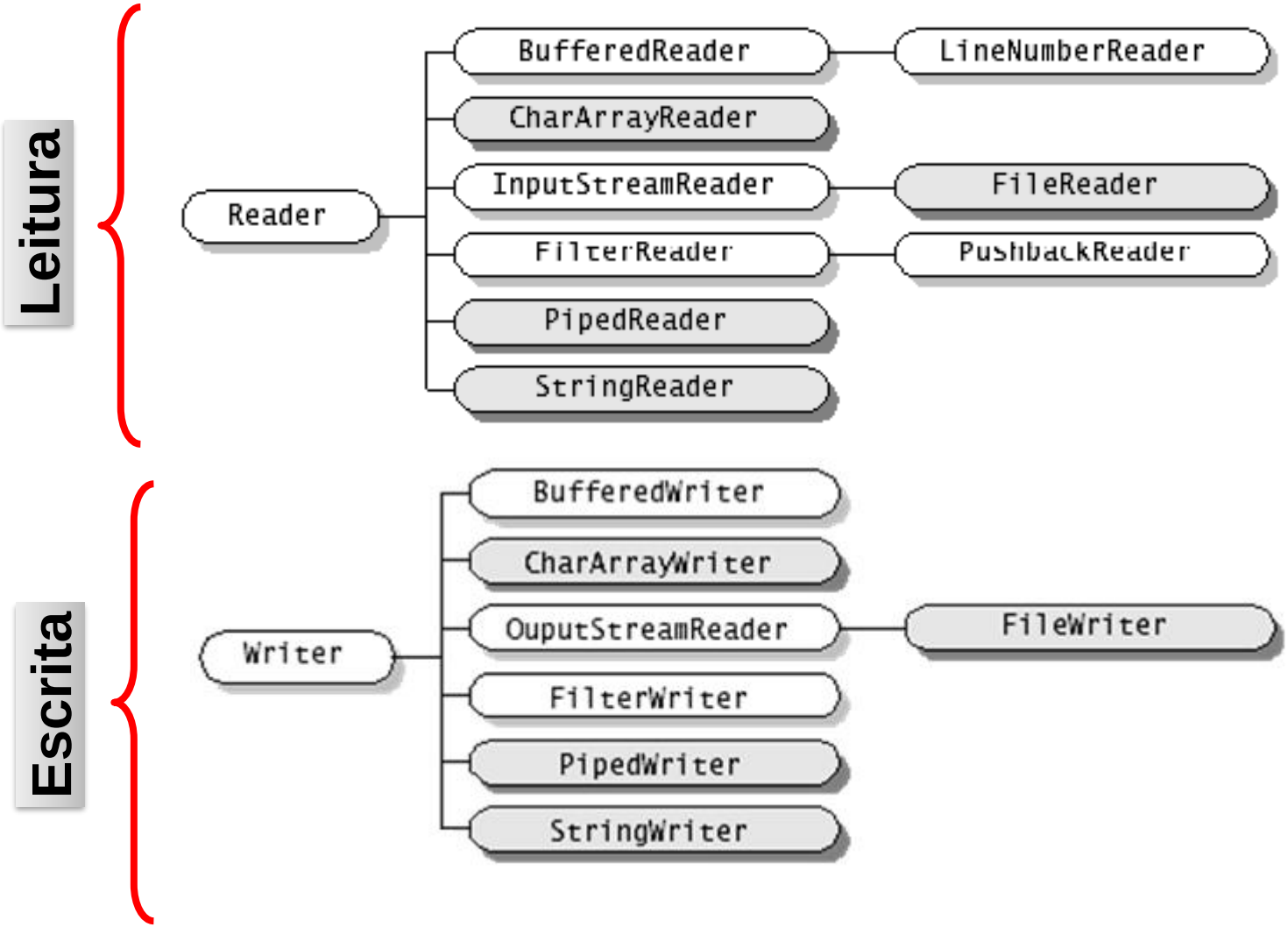
Escrita: a aplicação escreve o valor de uma variável no arquivo.



Programa Java **Arquivo**

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Arquivos



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Tipos de Arquivos

Arquivos de texto: são compostos por uma série de caracteres ASCII agrupados em uma ou mais linhas. São compreendidos pelos seres humanos.

Arquivos binários: composto por uma série de bytes representados por caracteres não compreendidos pelo ser humano. Ex.: imagens, vídeo, áudio, etc.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Manipulação de Arquivos em Java

Pacote java.io possui as classes para a manipulação de arquivos.

Sintaxe:

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

Essas classes são divididas em duas hierarquias de acordo com o tipo de arquivos que manipulam:

`FileInputStream/FileOutputStream` - arquivos binários

`FileReader/FileWriter` - arquivos de texto

Os arquivos e diretórios podem ser representados através da classe File.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe File

`public String getParent();` retorna o diretório pai

`public list();` retorna lista de arquivos contidos no diretório

`public boolean isFile();` retorna se é um arquivo

`public boolean isDirectory();` retorna se é um diretório

`public boolean delete();` tenta apagar o diretório ou arquivo

`public long length();` retorna o tamanho do arquivo em bytes

`public boolean mkdir();` cria um diretório com o nome do arquivo

`public String getAbsolutePath();` retorna o caminho absoluto

`public String getPath();` retorna o caminho

`public String getName();` retorna o último nome da seqüência de nomes do nome do caminho

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe FileReader

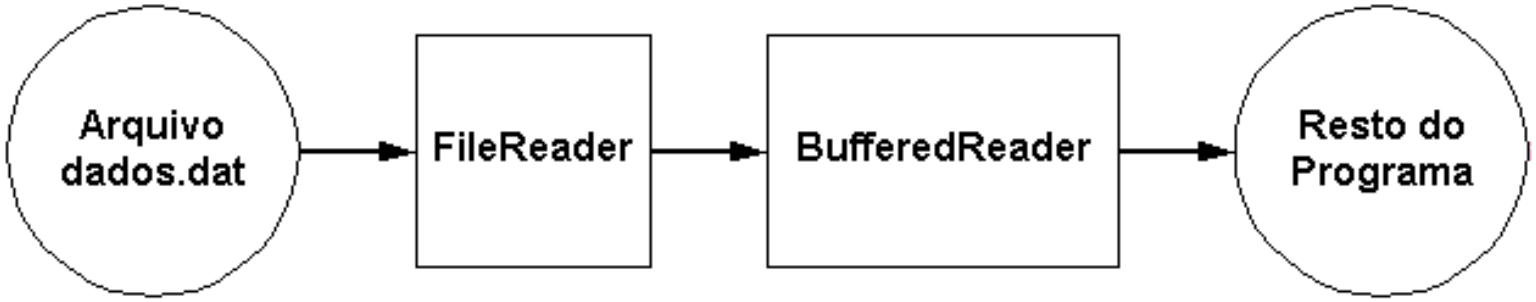
Utilizada para escrita em arquivos de texto

Sintaxe:

```
public FileReader(String name)
public FileReader(File file)
```

Usa o método read() para ler um caracter por vez

Para agilizar a leitura é usada a classe BufferedReader



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe FileWriter

Utilizada para escrita em arquivos de texto

Sintaxe:

```
public FileWriter(String name)
public   FileWriter(String   name,   boolean
append)
public FileWriter(File file)
public FileWriter(File file, boolean append)
```

Usa o método write() para escrever um caracter por vez

Para agilizar a escrita é utilizada a classe BufferedWriter

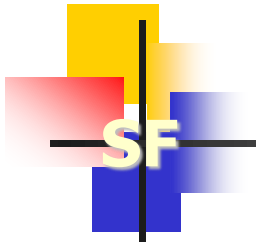
Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe FileOutputStream

Utilizada para escrita em arquivos binários

Sintaxe:

```
public FileOutputStream(String name)
public   FileOutputStream(String   name,   boolean
append)
public FileOutputStream(File file)
public FileOutputStream(File file, boolean append)
```



Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe FileInputStream

Utilizada para leitura de arquivos binários

Sintaxe:

```
public FileInputStream(String name)
public FileInputStream(File file)
```

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe BufferedRead

Utilizada para leitura de arquivos de texto.
Caracteres são armazenados em buffers após serem lidos.

Sintaxe:

```
FileReader fr = new FileReader (filename);
BufferedReader in = new BufferedReader (fr);
```

"`String readLine()`" //Retorna a próxima linha de texto do arquivo;

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Classe BufferedWriter

Utilizada para escrita de arquivos de texto.

Sintaxe:

```
FileWriter fw = new FileWriter(filename);

BufferedWriter bw = new BufferedWriter( fw );

bw.write( "Texto a ser escrito no txt" ); //escreve o conteúdo no arquivo
bw.newLine(); //quebra de linha
bw.close(); //fecha os recursos
fw.close(); //fecha os recursos
```

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

1. Crie uma aplicação contendo apenas uma classe com um main que leia os dados da entrada padrão. Para isso, você vai precisar de um BufferedReader que leia do System.in

2. Considere dois arquivos: arq1 e arq2, ambos arquivos de texto. Crie classes em Java que implementem as seguintes funcionalidades:

    a) copia o conteúdo do arq1 para arq2

    b) copia o conteúdo do arq1 para arq2 ao final do arq1 (append)

    c) imprime na tela todo o conteúdo do arquivo arq1

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

3. Crie uma aplicação que represente uma agenda de contatos, onde o usuário pode armazenar o nome, e-mail, telefone, endereço e data de aniversário em um arquivo de texto, o usuário pode armazenar quantos contatos forem necessários.

4. Crie uma aplicação contendo uma interface gráfica para a agenda criada no exercício 3, onde o usuário pode escolher inserir novo contato, apagar um contato existente e localizar algum contato cadastrado.

5. Crie uma aplicação que gere uma versão “criptografada” de um arquivo texto trocando cada caractere de código ASCII j pelo caractere de código ASCII j+k, onde k é um valor inteiro especificado pelo usuário. Fique atento para não gerar códigos ASCII fora da faixa permitida para os caracteres (valores entre 32 a 126).

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

6. Uma loja possui 4 filiais, cada uma com um código de 1 a 4. Um arquivo contendo todas as vendas feitas durante o dia nas quatro filiais é gerado a partir de uma planilha, sendo que cada linha do arquivo contém o número da filial e o valor de uma venda efetuada, separados por vírgula.

Ex.:
1,189.90
1,45.70
3,29.90
4,55.00

No exemplo acima, a filial 1 fez duas vendas, a primeira totalizando R$ 189,90 e a segunda R$ 45,70. A filial 3 fez uma venda de R$ 29,90 e a 4 uma de R$ 55,00. Crie uma aplicação que leia este arquivo e informe, ao final, o total e o valor médio das vendas de cada filial.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

7. Crie uma aplicação que leia um arquivo texto chamado “entrada.txt” e imprima, em outro arquivo texto, denominado “saida.txt”, o total de letras, vogais, consoantes, espaços em branco, palavras e o total de linhas encontradas no primeiro arquivo.

8. Crie uma aplicação que receba um nome de arquivo e uma sequencia de palavras como argumentos na linha de comando, informe se o arquivo existe ou não, caso não exista, crie-o e, por fim, escreva neste arquivo a sequencia de palavras passadas como argumentos.

9. Crie uma aplicação que leia dois arquivos texto (alunos.txt e medias.txt) contendo, respectivamente, o nome do aluno e a média das notas (um valor entre 0 e 10) de um grupo de 10 alunos. Em cada arquivo, antes do conteúdo existe um número inteiro sequencial (de 1 a 10) que corresponde ao código do aluno. Dessa forma, o arquivo alunos.txt terá em sua primeira linha um código e nas outras 10 linhas seguintes os nomes dos alunos. O mesmo ocorre para o arquivo medias.txt. Após a leitura dos arquivos, o programa deverá apresentar o resultado final “APROVADO” ou “REPROVADO” para cada aluno.

Prof. Dr. Sérgio Furgeri: sergio.furgeri@fatec.sp.gov.br

Exercícios

10. Crie uma aplicação que implemente uma interface gráfica que contenha uma tela inicial com as opções:

    a) Criar arquivo → quando selecionada solicita o nome do arquivo e cria o arquivo dentro do diretório c:\meusarquivos, caso o arquivo já exista, mostrar mensagem.

    b) Escrever arquivo → Quando selecionada escreve um texto digitado pelo usuário no arquivo criado na questão a, caso ele não exista, mostra mensagem.

    c) Ler arquivo → quando selecionada, realiza a leitura do arquivo escrito na questão b, caso ele não exista, mostra mensagem.

    d) Consultar arquivo → Faz uma busca de uma palavra fornecida pelo usuário no arquivo criado, retornando se a mesma existe ou não.