

POO - TDE

Armazenamento de dados em arquivos

Prof. Alcides Calsavara
PUCPR

Conceitos



1. Arquivo txt
2. Tabela de dados
3. Arquivo csv
4. Conversão de string para dados numéricos e boolean



Arquivo txt





Leitura de dados de um arquivo txt



FileReader

https://www.tutorialspoint.com/java/java_filereader_class.htm

```
class LeitorTXT {  
    public static void main(String[ ] args) {  
        try {  
            FileReader arquivo =  
                new FileReader("dados.txt");  
            BufferedReader buffer =  
                new BufferedReader(arquivo);  
            String str;  
            while ((str = buffer.readLine()) != null) {  
                System.out.println(str);  
            }  
        }  
        catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Crie um arquivo dados.txt e teste o programa acima.



Escrita de dados em um arquivo txt



FileWriter



https://www.tutorialspoint.com/java/java_filewriter_class.htm



Tabela de Dados



Tabela de Dados

Cotação do real:

| moeda | valor (R\$) |
|---------|-------------|
| dolar | 5.39 |
| euro | 6.38 |
| libra | 6.96 |
| bitcoin | 59638.56 |

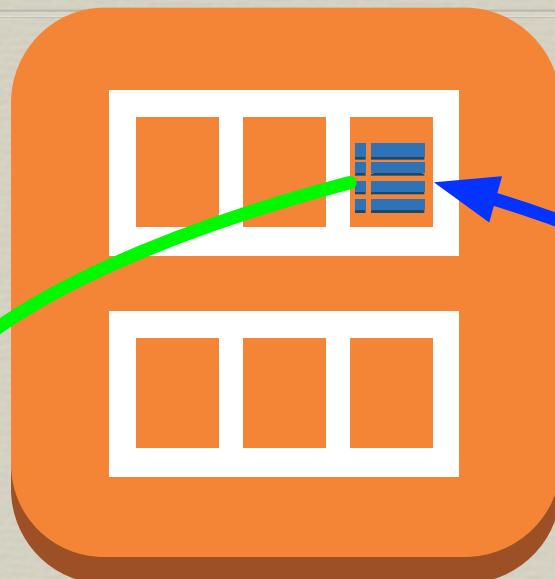
cabeçalho
(header)

linhas (registros)

colunas

MEMÓRIA

**TABELA DE DADOS
PRESENTE NA
MEMÓRIA É
GRAVADA EM
UM ARQUIVO**



**TABELA DE DADOS
É LIDA DE UM ARQUIVO
E ARMAZENADA NA
MEMÓRIA**

DISCO

**Locais de armazenamento
de uma tabela de dados**



Arquivo csv



comma-separated values

Arquivo csv



Arquivo csv no Excel (tabela de dados)

The screenshot shows a Microsoft Excel spreadsheet titled "cotacao.csv". The data consists of five rows of currency pairs and their values:

| | A | B | C | D | E | F |
|---|---------|----------|---|---|---|---|
| 1 | moeda | valor | | | | |
| 2 | dolar | 5.39 | | | | |
| 3 | euro | 6.38 | | | | |
| 4 | libra | 6.96 | | | | |
| 5 | bitcoin | 59638.56 | | | | |
| 6 | | | | | | |

Arquivo csv no editor de texto

The screenshot shows a text editor displaying the same CSV data as the Excel spreadsheet. The data is presented as a series of lines, where each line contains a currency name followed by its value, separated by a semicolon.

```
moeda;valor
dolar;5.39
euro;6.38
libra;6.96
bitcoin;59638.56
```

valores na mesma linha são separados por vírgula ou por ponto-e-vírgula

Arquivo CSV

Arquivo csv no Excel (tabela de dados)

The screenshot shows a Microsoft Excel spreadsheet titled 'cotacao.csv'. The data is presented in a table with columns labeled 'moeda' and 'valor'. The rows contain the following data:

| | A | B |
|---|---------|----------|
| 1 | moeda | valor |
| 2 | dolar | 5.39 |
| 3 | euro | 6.38 |
| 4 | libra | 6.96 |
| 5 | bitcoin | 59638.56 |
| 6 | | |

The Excel interface includes a ribbon bar with tabs like Home, Layout, Tables, Charts, and a formula bar at the top. The status bar at the bottom shows the file name 'cotacao.csv'.

Arquivo csv no editor de texto

The screenshot shows a text editor displaying the same CSV data as the Excel spreadsheet. The text is formatted as follows:

```
moeda;valor
dolar;5.39
euro;6.38
libra;6.96
bitcoin;59638.56
```

A yellow callout points from the word 'moeda' in the first line to the text 'dados na mesma linha são separados por vírgula ou por ponto-e-vírgula' below.

a primeira linha
do arquivo contém o
cabeçalho da tabela

dados na mesma linha são
separados por vírgula ou
por ponto-e-vírgula

Exemplo



```
nome,idade,email  
Gustavo,28,gustavo@dicasdeprogramcao.com.br  
Joao,35,joao@dicasdejava.com.br  
Maria,23,maria@dicasdeprogramacao.com.br  
Ana,25,ana@dicasdejava.com.br
```

Gere o correspondente arquivo com o nome **pessoas.csv**:

1. Usando um editor de texto
2. Usando uma ferramenta de planilha (por exemplo, Excel)

Leitura de um arquivo csv em tabela de strings usando BufferedReader

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

public class LeitorBuffer{
    public static void main(String[ ] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";

        List<List<String>> tabela = new ArrayList<>();
    }
}
```

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

public class LeitorBuffer{
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";

        List<List<String>> tabela = new ArrayList<>();

        try
        {
            FileReader arquivo = new FileReader(NOME_ARQUIVO);
            BufferedReader br = new BufferedReader(arquivo);

        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

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

public class LeitorBuffer {
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        List<List<String>> tabela = new ArrayList<>();
        try
        {
            FileReader arquivo = new FileReader(NOME_ARQUIVO);
            BufferedReader buffer = new BufferedReader(arquivo);
            String cabecalho = buffer.readLine();
            while (buffer.ready())
            {
                String linha = buffer.readLine();
                String[] tokens = linha.split(SEPARADOR);
                List<String> registro = Arrays.asList(tokens);
                tabela.add(registro);
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }

        for (List<String> registro: tabela)
        {
            for (String campo: registro)
                System.out.print(campo + " ");
            System.out.println();
        }
    }
}
```

Leitura de um arquivo csv em tabela de strings usando Scanner

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

public class LeitorScanner {
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        List<List<String>> tabela = new ArrayList<>();
        try
        {
            File arquivo = new File(NOME_ARQUIVO);
            Scanner scanner_arquivo = new Scanner(arquivo);
            String cabecalho = scanner_arquivo.nextLine();
            while (scanner_arquivo.hasNextLine())
            {
                String linha = scanner_arquivo.nextLine();
                Scanner scanner_linha = new Scanner(linha);
                scanner_linha.useDelimiter(SEPARADOR);
                List<String> registro = new ArrayList<>();
                while (scanner_linha.hasNext())
                {
                    String campo = scanner_linha.next();
                    registro.add(campo);
                }
                tabela.add(registro);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        for (List<String> registro: tabela)
        {
            for (String campo: registro)
                System.out.print(campo + " ");
            System.out.println();
        }
    }
}
```

Conversão de string para dados numéricos e boolean

Classes e métodos da Biblioteca de Java

1. Integer
 - ❖ parseInt
2. Float
 - ❖ parseFloat
3. Double
 - ❖ parseDouble
4. Boolean
 - ❖ parseBoolean

Exemplo de código



```
String texto_double = "3.49";
double d = Double.parseDouble(texto_double);
System.out.println(d);

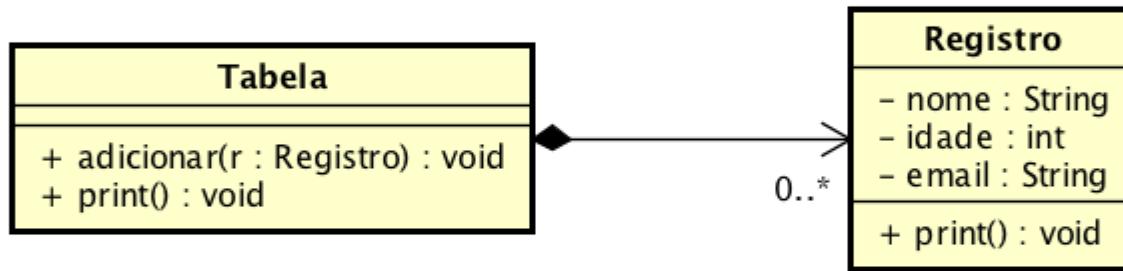
String texto_int = "3";
int k = Integer.parseInt(texto_int);
System.out.println(k);

String texto_float = "3.49";
float f = Float.parseFloat(texto_float);
System.out.println(f);

String texto_boolean = "true";
boolean b = Boolean.parseBoolean(texto_boolean);
System.out.println(b);
```

Leitura de um arquivo csv em tabela de objetos

pkg



```
public class Registro {  
    private String nome;  
    private int idade;  
    private String email;  
  
    public Registro(String nome, int idade, String email)  
    {  
        this.nome = nome;  
        this.idade = idade;  
        this.email = email;  
    }  
  
    public void print()  
    {  
        System.out.println(nome + " " + idade + " " + email);  
    }  
}
```

```
import java.util.*;  
  
public class Tabela {  
    private List<Registro> registros;  
    public Tabela()  
    {  
        registros = new ArrayList<Registro>();  
    }  
  
    public void adicionar(Registro r)  
    {  
        registros.add(r);  
    }  
  
    public void print()  
    {  
        for (Registro r: registros) r.print();  
    }  
}
```

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

public class LeitorPOO {

    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        Tabela tabela = new Tabela();
        try
        {
            File arquivo = new File(NOME_ARQUIVO);
            Scanner scanner_arquivo = new Scanner(arquivo);
            String cabecalho = scanner_arquivo.nextLine();
            while (scanner_arquivo.hasNextLine())
            {
                String linha = scanner_arquivo.nextLine();
                Scanner scanner_linha = new Scanner(linha);
                scanner_linha.useDelimiter(SEPARADOR);
                String nome = scanner_linha.next();
                int idade = scanner_linha.nextInt();
                String email = scanner_linha.next();
                Registro registro = new Registro(nome, idade, email);
                tabela.adicionar(registro);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        tabela.print();
    }
}
```