

POO - PROGRAMAÇÃO ORIENTADA A OBJETOS

Trabalhando com Arquivos

Rodrigo R Silva

Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense
Campus Bagé

Nesta Aula Veremos...

- 1 Introdução
- 2 Arquivos .txt
- 3 Arquivos .xls
- 4 Arquivos .json

Introdução

Conceito

- A leitura e escrita de arquivos por um programa pode se muito útil:
 - 1 Seja para servir como arquivo de configurações;
 - 2 Evitar a necessidade de modificação frequente do código fonte do próprio programa para cada cenário;
 - 3 Simplesmente para servir para entrada e saída de dados.

Dependências

```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>5.0.0</version>
</dependency>

<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.9</version>
</dependency>
```

Arquivos .txt

Gravar Arquivo

```
public static void initGravaArquivo() throws IOException {
    File arquivo = new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo.txt");
    //File arquivo = new File("/home/rodrigo/eclipse-workspace/projeto-poo-arquivos/src/main/resources/arquivo.csv");
    if(!arquivo.exists()) {
        arquivo.createNewFile();
    }
    FileWriter escrever_arquivo = new FileWriter(arquivo);
    //escrever_arquivo.write("TEXT0");
    try {
        ClasseDAO classeDao = new ClasseDAO();
        List<BeanAlunoFone> list = classeDao.listarAlunoFone(5L);

        for(BeanAlunoFone b : list) {
            escrever_arquivo.write(b.getNome()+";"+b.getEmail()+";"+b.getNumero()+";"+b.getTipo()+"\n");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    escrever_arquivo.flush();//Persistir
    escrever_arquivo.close();//Fechar
}
```

Ler Arquivo

```
public static void initLerArquivo() throws FileNotFoundException {
    FileInputStream entrada = new FileInputStream(
        new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo.txt"));

    //FileInputStream entrada = new FileInputStream(
    //    new File("/home/rodrigo/eclipse-workspace/projeto-poo-arquivos/src/main/resources/arquivo.csv"));
    Scanner lerArquivo = new Scanner(entrada, "UTF-8");
    //Ler arquivo
    /*while (lerArquivo.hasNext()) {
        String linha = lerArquivo.nextLine();

        if(linha != null && !linha.isEmpty()) {
            System.out.println(linha);
        }
    }*/
    //Ler arquivo e add em uma lista de objetos
    List<BeanAlunoFone> list = new ArrayList<BeanAlunoFone>();
    while (lerArquivo.hasNext()) {
        String linha = lerArquivo.nextLine();

        if(linha != null && !linha.isEmpty()) {
            String[] dados = linha.split("\\;");

            BeanAlunoFone beanAlunoFone = new BeanAlunoFone();
            beanAlunoFone.setNome(dados[0]);
            beanAlunoFone.setEmail(dados[1]);
            beanAlunoFone.setNumero(dados[2]);
            beanAlunoFone.setTipo(dados[3]);

            list.add(beanAlunoFone);
        }
    }
    for(BeanAlunoFone b : list) {
        System.out.println(b.toString());
    }
}
```


Arquivos .xls

Gravar Arquivo

```
public static void initGravaArquivoPOI() throws Throwable {
    File arquivo = new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo_planilha.xls");
    if(!arquivo.exists()) {
        arquivo.createNewFile();
    }

    ClasseDAO classeDao = new ClasseDAO();
    List<BeanAlunoFone> list = classeDao.listarAlunoFone(5L);

    HSSFWorkbook hssfWorkbook = new HSSFWorkbook();//Vai ser usado para escrever a planilha
    HSSFSheet linhasInformacoes = hssfWorkbook.createSheet("Planilha de Informações");//Criar planilha
    //Controle linha
    int numeroLinha = 0;
    for(BeanAlunoFone b : list) {
        Row linha = linhasInformacoes.createRow(numeroLinha++);//Criando a linha

        int celula = 0;

        Cell celNome = linha.createCell(celula++);
        celNome.setCellValue(b.getNome());

        Cell celEmail = linha.createCell(celula++);
        celEmail.setCellValue(b.getEmail());

        Cell celNumero = linha.createCell(celula++);
        celNumero.setCellValue(b.getNumero());

        Cell celTipo = linha.createCell(celula++);
        celTipo.setCellValue(b.getTipo());

    }//Termina de montar a planilha

    FileOutputStream registro = new FileOutputStream(arquivo);
    hssfWorkbook.write(registro);
    registro.flush();
    registro.close();
}
```

Ler Arquivo

```

public static void initLerArquivoPOI() throws Exception{
    FileInputStream entrada = new FileInputStream(
        new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo_planilha.xls"));

    HSSFWorkbook hssfWorkbook = new HSSFWorkbook(entrada); //Prepara a entrada do arquivo excel para ler
    HSSFSheet planilha = hssfWorkbook.getSheetAt(0); //Pega a 1ª planilha

    Iterator<Row> linhaIterator = planilha.iterator();

    List<BeanAlunoFone> list = new ArrayList<BeanAlunoFone>();

    while (linhaIterator.hasNext()) { //Enquanto tiver linha no arquivo excel
        Row linha = linhaIterator.next(); //Dados da pessoa na linha
        Iterator<Cell> celula = linha.iterator();

        BeanAlunoFone beanAlunoFone = new BeanAlunoFone();

        while(celula.hasNext()) { //Enquanto tiver celulas
            Cell cell = celula.next();
            switch (cell.getColumnIndex()) {
                case 0: beanAlunoFone.setNome(cell.getStringCellValue());
                    break;
                case 1: beanAlunoFone.setEmail(cell.getStringCellValue());
                    break;
                case 2: beanAlunoFone.setNumero(cell.getStringCellValue());
                    break;
                case 3: beanAlunoFone.setTipo(cell.getStringCellValue());
                    break;
            }
        } //Fim das celulas da linha
        list.add(beanAlunoFone);
    }
    entrada.close(); //Fecha o arquivo

    for(BeanAlunoFone b : list) {
        System.out.println(b.toString());
    }
}

```

Editar Arquivo

```
public static void initEditarArquivoPOI() throws Exception {  
  
    File arquivo = new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo_planilha.xls");  
  
    FileInputStream entrada = new FileInputStream(arquivo);  
  
    HSSFWorkbook hssfWorkbook = new HSSFWorkbook(entrada); //Prepara a entrada do arquivo excel para ler  
    HSSFSheet planilha = hssfWorkbook.getSheetAt(0); //Pega a 1ª planilha  
  
    Iterator<Row> linhaIterator = planilha.iterator();  
  
    while (linhaIterator.hasNext()) { //Enquanto tiver linha no arquivo excel  
        Row linha = linhaIterator.next(); //Dados da pessoa na linha  
  
        int numeroCelulas = linha.getPhysicalNumberOfCells();  
  
        Cell cell = linha.createCell(numeroCelulas);  
        cell.setCellValue("5.487,20");  
        //Fim das celulas da linha  
    }  
  
    entrada.close(); //Fecha o arquivo  
  
    FileOutputStream saida = new FileOutputStream(arquivo);  
    hssfWorkbook.write(saida);  
    saida.flush();  
    saida.close();  
}
```

Editar Arquivo

```
public static void initEditarArquivoPOI2() throws Exception {  
  
    File arquivo = new File("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/arquivo_planilha.xls");  
  
    FileInputStream entrada = new FileInputStream(arquivo);  
  
    HSSFWorkbook hssfWorkbook = new HSSFWorkbook(entrada); //Prepara a entrada do arquivo excel para ler  
    HSSFSheet planilha = hssfWorkbook.getSheetAt(0); //Pega a 1ª planilha  
  
    Iterator<Row> linhaIterator = planilha.iterator();  
  
    while (linhaIterator.hasNext()) { //Enquanto tiver linha no arquivo excel  
        Row linha = linhaIterator.next(); //Dados da pessoa na linha  
  
        String valorCelula = linha.getCell(0).getStringCellValue();  
        valorCelula = "Novo Nome";  
        linha.getCell(0).setCellValue(valorCelula);  
        //Fim das celulas da linha  
    }  
  
    entrada.close(); //Fecha o arquivo  
  
    FileOutputStream saida = new FileOutputStream(arquivo);  
    hssfWorkbook.write(saida);  
    saida.flush();  
    saida.close();  
}
```

Arquivos .json

Gravar Arquivo

```
public static void initGravarJSON() throws IOException {

    Pessoa p = new Pessoa();
    p.setNome("Rodrigo");
    p.setCpf("999999999999");
    p.setLogin("admin");
    p.setSenha("minhaSenha");

    Pessoa p2 = new Pessoa();
    p2.setNome("Nai");
    p2.setCpf("888888888888");
    p2.setLogin("meuLogin");
    p2.setSenha("admin");

    List<Pessoa> list = new ArrayList<Pessoa>();
    list.add(p);
    list.add(p2);

    Gson gson = new GsonBuilder().setPrettyPrinting().create();
    String jsonUser = gson.toJson(list);

    System.out.println(jsonUser);

    FileWriter fileWriter = new FileWriter("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/"
        + "arquivo_json.json");

    //Se ocorrer erro de encond
    //OutputStreamWriter gravar = new OutputStreamWriter(new FileOutputStream(arquivo), "UTF-8");

    fileWriter.write(jsonUser);
    fileWriter.flush();
    fileWriter.close();
}
```

Ler Arquivo

```
public static void initLerJSON() throws IOException {  
    //-----LER ARQUIVO-----  
    FileReader fileReader = new FileReader("/home/rodrigo/git/repository14/projeto-poo-arquivos/src/main/resources/"  
        + "arquivo_json.json");  
  
    JSONArray jsonArray = (JSONArray) JsonParser.parseReader(fileReader);  
  
    List<Pessoa> listPessoas = new ArrayList<Pessoa>();  
  
    for(JsonElement jsonElement : jsonArray) {  
        Pessoa pessoa = new Gson().fromJson(jsonElement, Pessoa.class);  
        listPessoas.add(pessoa);  
    }  
    System.out.println(listPessoas);  
}
```


OBRIGADO!