

SERIALVERSIONUID

SERIALVERSIONUID

- Definição
 - O termo objeto serializado significa que o objeto será transformado em bytes e poderá ser armazenado em disco ou transmitido por um stream.
 - Stream é um objeto de transmissão de dados, onde o fluxo de dados serial é feito por meio de uma origem e de um destino.
 - Em Java, dois tipos de stream:
 - FileOutputStream
 - FileInputStream

SERIALVERSIONUID

- `FileOutputStream`
 - É um fluxo de arquivo que permite a gravação em disco.
- `FileInputStream`
 - É um fluxo de arquivo que permite a leitura de um arquivo em disco.

SERIALVERSIONUID

- Manipulando objetos serializados
 - ObjectInputStream
 - Permite inserir objetos serializados.
 - ObjectOutputStream
 - Permite recuperar objetos serializados.
- Diversas aplicações que utilizam o recurso de serialização.
 - streams para JME
 - Qualquer aplicação que trabalhe com fluxo de I/O.
 - Qualquer tipo de trabalho que envolver persistência ou tramitação de dados.

SERIALVERSIONUID

- Exemplo:

```
import java.io.Serializable;

public class Cliente implements Serializable{
    private String nome; private char sexo; private String cpf;

    public Cliente(String nome, char sexo, String cpf){
        super();
        this.nome = nome;
        this.sexo = sexo;
        this.cpf = cpf;
    }

    //getters and setters

    public String toString(){
        return this.nome + " / " + "Sexo:" + this.sexo + "\n" + "Cpf: "
        + this.cpf;
    }
}
```

SERIALVERSIONUID

- Exemplo (continuação):

```
package br.com.artigos.serial;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

public class ExemploStream {
    public static void main(String[] args) {
        // Cria o objeto serializado
        Cliente cliente = new Cliente("Rodrigo", 'M', "00000111111");
        try { //Gera o arquivo para armazenar o objeto
            FileOutputStream arquivoGrav =
                new FileOutputStream("/Users/henrique/Desktop/saida.dat"); //Classe
responsavel por inserir os objetos
            ObjectOutputStream objGravar = new ObjectOutputStream(arquivoGrav);
//Grava o objeto cliente no arquivo
            objGravar.writeObject(cliente);
            objGravar.flush();
            objGravar.close();
            arquivoGrav.flush();
```

```
            arquivoGrav.close();
            System.out.println("Objeto gravado com sucesso!");
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        System.out.println("Recuperando objeto: ");
        try {
            //Carrega o arquivo
            FileInputStream arquivoLeitura = new FileInputStream("/saida.dat");
            // Classe responsavel por recuperar os objetos do arquivo
            ObjectInputStream objLeitura =
                new ObjectInputStream(arquivoLeitura);
            System.out.println(objLeitura.readObject());
            objLeitura.close();
            arquivoLeitura.close();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

REFERÊNCIAS

- DEITEL, P.J. Java - Como Programar. Porto Alegre: Bookman, 2001.
- NIEMEYER, Patrick. Aprendendo java 2 SDK. Rio de Janeiro: Campus, 2000.
- MORGAN, Michael. Java 2 para Programadores Profissionais. Rio de Janeiro: Ciência Moderna, 2000.
- HORSTMANN, Cay, S. e CORNELL, Gary. Core Java 2. São Paulo: Makron Books, 2001 v.1. e v.2.