

# Programação Orientada a Objetos

Manipulação de Arquivos

Prof. Francisco de Assis S. Santos, Dr.

São José, 2014.



# Classe Pessoa.java

```
import java.io.Serializable;

public class Pessoa implements Serializable{
    private String nome;
    private String cpf;

public Pessoa(String nome, String cpf) {
        this.nome = nome;
        this.cpf = cpf;
    }

public void imprimir(){
        System.out.println("Nome: " + nome);
        System.out.println("CPF: " + cpf);
}
```

A interface java.io. Serializable permite que objetos sejam serializados, ou seja, permite que esses objetos sejam gravados em arquivos, transmitidos pela rede, etc.



### Classe Principal.java

```
public class Principal {
  /* Criando um vetor de tamanho 2 */
  private Pessoa[] agenda = new Pessoa[2];
  public void salvarEmDisco(){
  //veja o codigo na proxima lamina
  public void lerDoDisco(){
  //veja o codigo daqui 2 laminas
 public static void main(String[] args) {
  Principal p = new Principal();
  p.agenda[0] = new Pessoa("joao", "123");
  p.agenda[1] = new Pessoa("maria", "456");
  p.salvarEmDisco();
  p.lerDoDisco();
}//fim da classe
```



# Classe Principal.java

```
public void salvarEmDisco(){
 File arquivo = new File("agenda.dat");
     try {
       FileOutputStream fout = new FileOutputStream(arquivo);
       ObjectOutputStream oos = new ObjectOutputStream(fout);
       // gravando o vetor 'agenda' no arquivo chamado 'agenda.dat'
       oos.writeObject(this.agenda);
       oos.flush(); // limpando dados em buffer
       oos.close(); // fechando fluxo de saida
       fout.close(); // fechando arquivo
     } catch (Exception ex) {
       System.err.println("erro: " + ex.toString());
```



# Classe Principal.java

```
public void lerDoDisco(){
   File arquivo = new File("agenda.dat");
   try {
      FileInputStream fin = new FileInputStream(arquivo);
      ObjectInputStream oin = new ObjectInputStream(fin);
      // Lendo os objetos de um arquivo
      this.agenda = (Pessoa[]) oin.readObject();
      oin.close(); //fechando fluxo de entrada
      fin.close(); //fechando arquivo
      // Uma forma de diferente do for para percorrer vetores
      for (Pessoa p : this.agenda) {
        p.imprimir();
   } catch (Exception ex) {
      System.err.println("erro: " + ex.toString());
```



# Referências

Notas de aula do Prof. Emerson Ribeiro de Mello