

# Programação Orientada a Objetos (PO24CP)

## Aula #06

### Gravação de objetos em arquivo

Prof<sup>a</sup> Luciene de Oliveira Marin  
lucienemarin@utfpr.edu.br

## Gravação de objetos em arquivo

# Classe **Pessoa.java** (moodle)

```
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 imprimirDados(){
        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 enviados pela interface de rede, gravados em arquivos, etc.

# Classe **DemoPessoaGravar.java** (moodle)

```
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.ObjectOutputStream;  
  
public class Principal {  
  
    public static void main(String[] args) {  
        /*Criando um vetor que permite alocar ate  
        2 objetos da classe Pessoa */  
        Pessoa[] agenda = new Pessoa[2];  
  
        //Criando 2 objetos da classe Pessoa  
        agenda[0] = new Pessoa("joao", "123");  
        agenda[1] = new Pessoa("maria", "456");  
  
        /*Inserir aqui o trecho de codigo para  
        gravar em arquivo (proxima pagina)*/  
    }  
} //fim da classe
```

## Trecho de código para **gravar** objeto em arquivo:

```
File arquivo = new File("agenda.dat");
try {
    FileOutputStream fout = new FileOutputStream(arquivo);
    ObjectOutputStream oos = new ObjectOutputStream(fout);

    // gravando o vetor de pessoas no arquivo
    oos.writeObject(agenda);

    oos.flush();
    oos.close();
    fout.close();
} catch (Exception ex) {
    System.err.println("erro: " + ex.toString());
}
```

# Classe **DemoPessoaLer.java** (moodle)

Classe aplicativa para **ler** objeto do arquivo:

```
public class DemoPessoaLer{

    public static void main(String[] args) {
        File arquivo = new File("agenda.dat");
        try {
            FileInputStream fin = new FileInputStream(arquivo);
            ObjectInputStream oin = new ObjectInputStream(fin);
            /*Lendo os objetos de um arquivo e fazendo a
            coercao de tipos*/
            Pessoa[] agendaArq = (Pessoa[]) oin.readObject();
            oin.close();
            fin.close();
            //Uma forma de diferente do for para percorrer vetores
            for (Pessoa p : agendaArq) {
                p.imprimirDados();
            }
        } catch (Exception ex) {
            System.err.println("erro: " + ex.toString());
        }
    }
}
```

## Exercício: Agenda Simplória

- Utilizar os códigos da aula como base.

# Exercício - Agenda Simplória

Desenvolva uma aplicação com menu em modo texto que permita ao usuário:

- 1 Incluir pessoa
- 2 Listar pessoas cadastradas na agenda
- 3 Excluir pessoa (pelo nome)
- 4 Gravar a agenda em arquivo
- 5 Ler a agenda do arquivo
- 6 Sair do programa