

```
package Model;
```

```
import java.io.Serializable;
```

```
/*001*/
```

```
public class PessoaFisica extends Pessoa implements Serializable {
```

```
    /* 002 */
```

```
    private String cpf;
```

```
    private int idade;
```

```
    /* 003 */
```

```
    public PessoaFisica() {
```

```
        /* 003.1 */
```

```
        super();
```

```
    }
```

```
    /* 004 */
```

```
    public PessoaFisica(int id, String nome, int idade, String cpf) {
```

```
        super(id, nome);/* 004.1 */
```

```
        this.cpf = cpf;
```

```
        this.idade = idade;
```

```
    }
```

```
    /* 005 */
```

```
public void exibir() {  
    super.exibir();/* 005.1 */  
    System.out.println("Digite seu CPF: " + cpf);  
    System.out.println("Digite sua Idade: " + idade);  
}
```

```
/* 006 */  
public int getIdade() {  
    return idade;  
}
```

```
/* 006 */  
public void setIdade(int idade) {  
    this.idade = idade;  
}
```

```
/* 006 */  
public String getCpf() {  
    return cpf;  
}
```

```
/* 006 */  
public void setCpf(String cpf) {  
    this.cpf = cpf;  
}
```

/*

* 001° Informação importante, eu estou extendendo o arquivo 'Pessoa',
* então caso eu não use o 'extends' e não aponte a arquivo (pensa como
* um modulo do Python) não vai funcionar, prestar bastante atenção
* pra não ficar quebrando cabeça a toa.

*

* 002° Aqui seria o campo de entrada, onde declaro as propriedades do
* ID e nome, se vão ou não ser String, int, long ou mesmo double.

*

* 003° Constructor inicial

* 003.1° Observação importante, o 'SUPER' seria uma superclasse e estou
* puxando ela através do extends 'Pessoa' para o Constructor Composto,
* que seria 'ID e Nome'

*

* 004° Constructor composto, observe que o 'super' esta puxando a
* informações do arquivo 'Pessoa' que seriam o 'ID e Nome', espero que
* estas informações me ajude com o decorer do curso.

*

* 005° Como vai ser apresentado pra o Usuario.

* 005.1° lembrar que devo replicar estas mesmas informações nos demais
* arquivos .java

*

* 006° Getter: metodo que retorna seu valor.

* 006° Setter: define ou atualiza.

$\ast/$

}