

# Programação Procedural e Programação Orientada a Objetos

## Programação procedural

Com a programação procedural devemos definir linha por linha o que nossa aplicação fará, devendo reescrever várias vezes o mesmo código para ações iguais/parecidas. Aplicações neste formato são de fato compostas por “começo”, “meio” e “fim”.

*\*Acredito que as funções não estão disponíveis na linguagem de programação Java.\**

**Funções:** As funções são pequenos, podendo ser grandes também, conjuntos de códigos que nos possibilitam fazer o uso do “reaproveitamento de código”, não sendo necessário reescrever o mesmo código 10 vezes para a mesma ação. Com o uso de funções podemos criar aplicações que “pulam de linha em linha” sem seguir uma ordem exata, eliminando o “começo”, “meio” e “fim” anteriormente informado.

## Programação orientada a objetos

Com a programação orientada a objetos podemos simular quaisquer coisas do mundo real dentro dos códigos.

**Classes:** O uso de classes nos permite “pegar” elementos do mundo real e “codifica-los” em nossas aplicações, não somente isso, mas também nos permite reutilizar o mesmo código e incorporá-lo a outras classes, gerando assim uma “herança” entre as classes.

**Métodos:** Os métodos são funções, porém são funções de classes. Cada classe possui seus métodos, podendo executar ações somente referente ao objeto em questão.

Exemplos de classes:

Começaremos criando uma classe que representa os celulares:

```

classe Celular {
    marca
    modelo
    armazenamento

    Celular(marca, modelo, armazenamento) {
        this.marca = marca
        this.modelo = modelo
        this.armazenamento = armazenamento
    }

    ligar() {
        imprima("O celular está ligando")
    }
}

```

A classe “Celular” é composta por três variáveis, que são “marca”, “modelo” e “armazenamento”, o método construtor “Celular” e um método próprio “ligar”. Precisamos definir o método construtor pois é nele que passaremos as informações quando formos usar a classe.

Usando a classe criada:

```

Celular celular1 = Celular("Iphone", "Iphone 15", "1tb") // Aqui definimos um novo celular.
Celular celular2 = Celular("Samsung", "Galaxy S24", "250gb") // Outro celular

```

Conseguimos criar dois celulares usando somente a classe Celular.

Agora vamos criar uma classe chamada “Pessoa”, a qual possuirá as variáveis “nome”, para o nome da pessoa, e “celular”, para o celular da pessoa, e por fim o método “ligarCelular”:

```

classe Pessoa {
    nome
    celular

    Pessoa(nome, celular) {
        this.nome = nome
        this.celular = celular
    }

    ligarCelular() {
        imprima(nome + "está ligando o celular " + celular.modelo)
        celular.ligar()
    }
}

```

Vamos fazer o uso da classe Pessoa:

```
Pessoa caique = Pessoa("Caique", celular1)
Pessoa eduardo = Pessoa("Eduardo", celular2)

caique.ligarCelular()
eduardo.ligarCelular()
```

Neste pequeno trecho de código definimos as pessoas junto aos celulares que elas possuem. Foi possível passar as variáveis "celular1" e celular2" que foram construídas usando a classe Celular para a classe Pessoa, podendo usar os métodos nelas contidas.

Este será o retorno no terminal:

```
Caique está ligando o celular Iphone 15
O celular está ligando

Eduardo está ligando o celular Galaxy S24
O celular está ligando
```

Ao fazer uso de classes e métodos podemos simplificar a programação aproveitando a alta reutilização de código.