



Aula 1

O que é **Orientação a objetos**?

- Criado na década de 70
- Impulsionado pela linguagem java nos anos 90

Como esse paradigma funciona?

Para cada necessidade importante, teremos **objetos** que interagem entre si e são compostos por **comportamentos** (Métodos) e por **estados** (Atributos).

Vantagens do paradigma

- O Paradigma trás melhor representação do mundo real
- Um software orientado a objetos é oportuno - Várias partes podem ser desenvolvidas em paralelo
- Um software orientado a objetos é manutenível/ Reutilizável
- Uma pequena modificação beneficia todas as partes que usarem o objeto



Pilares da Orientação a Objetos

- Abstração
- Encapsulamento
- Herança
- Polimorfismo

O que é uma
classe?



O que é um
objeto?



Classes e Objetos

Classe



Objetos



Exemplos

```
public class Contato {  
    String nome;  
    String telefone;  
    String cpf;  
    String email;  
}
```

Exemplos

```
public class Pessoa {  
    String nome;  
    int idade;  
    double peso;  
    double altura;  
}
```

Exemplos

```
public class ContaCorrente {  
    String agencia;  
    String numeroDaConta;  
    double saldo;  
    double limiteDeCredito;  
}
```

Métodos Construtores

Também chamado de Constructors, eles são responsáveis por criar um objeto em memória, ou seja, instanciar a classe que foi definida.

Exemplos

```
public Contato(String nome, String telefone, String email, int idade){  
    this.nome = nome;  
    this.telefone = telefone;  
    this.email = email;  
    this.idade = idade;  
}
```



Métodos

(Comportamentos)

Um método é um bloco de instruções que realizam uma tarefa específica. Um método precisa estar dentro de uma classe e pode ser reaproveitado.

Métodos

(Comportamentos)

```
public void mostrarDados() {  
    System.out.println("O nome é:" +nome);  
    System.out.println("O telefone é:" +telefone);  
    System.out.println("O e-mail é: " +email);  
    System.out.println("A idade é: " +idade);  
}
```




Métodos

```
modificadorDeAcesso tipoDeRetorno nomeDoMetodo (Parametros){  
  
    Instruções  
    retorno  
  
}
```

