A Programação Orientada a Objetos foi criada por **Alan Kay**, autor da linguagem **Smalltalk**. Antes mesmo da criação Orientada a Objetos, já existiam algumas aplicações, neste caso da linguagem Simula 67, criada por **Ole Johan Dahl** e **Kristen Nygaard** em 1967.
Veja na Figura 1 a trajetória que a programação teve para se chegar ao uso da POO.

1950 – 1960 Era do Caos	1970 – 1980 Era da Estruturação	1990 até agora Era dos Objetos
Saltos, gotos, variáveis não estruturadas, variáveis espalhadas ao longo do programa	If-then-else Blocos Registros Laços-While	Objetos Mensagens Métodos Herança

Figura 1: Linha do tempo das técnicas de programação

Elementos

A Programação Orientada a Objetos é formada por alguns itens, dentre os quais destacamos: Classes, Objetos, Atributos, Métodos, Construtores, que irão ser mostrados nesse artigo.

Classes

As classes de programação são projetos de um objeto, aonde têm características e comportamentos, ou seja, permite armazenar propriedades e métodos dentro dela. Para construir uma classe é preciso utilizar o pilar da abstração. Uma classe geralmente representa um substantivo, por exemplo: uma pessoa, um lugar, algo que seja “abstrato”.

Características das classes

- Toda classe possui um nome;
- Possuem visibilidade, exemplo: public, private, protected;
- Possuem membros como: Características e Ações;
- Para criar uma classe basta declarar a visibilidade + digitar a palavra reservada class + NomeDaClasse + abrir e fechar chaves { }.

Listagem 1: Declaração de uma classe na linguagem Java

```
public class Teste{
//ATRIBUTOS OU PROPRIEDADES
//MÉTODOS
}
```

Exemplo :

```
class Programa {
public static void main(String[] args) {
Conta minhaConta;
minhaConta = new Conta();
minhaConta.dono = "JOAO";
minhaConta.saldo = 1000.0;
System.out.println("Saldo atual: " + minhaConta.saldo);
}
}
```

Listagem 2: Classe Caes

```
public class Caes {
        public String nome;
        public int peso;
        public String corOlhos;
        public int quantPatas;

        public void falar(){
                //MÉTODO FALAR
        }

        public void andar(){
                //MÉTODO ANDAR
        }

        public void comer(){
                //MÉTODO COMER
        }

        public void dormir(){
                //MÉTODO DORMIR
        }
}
```

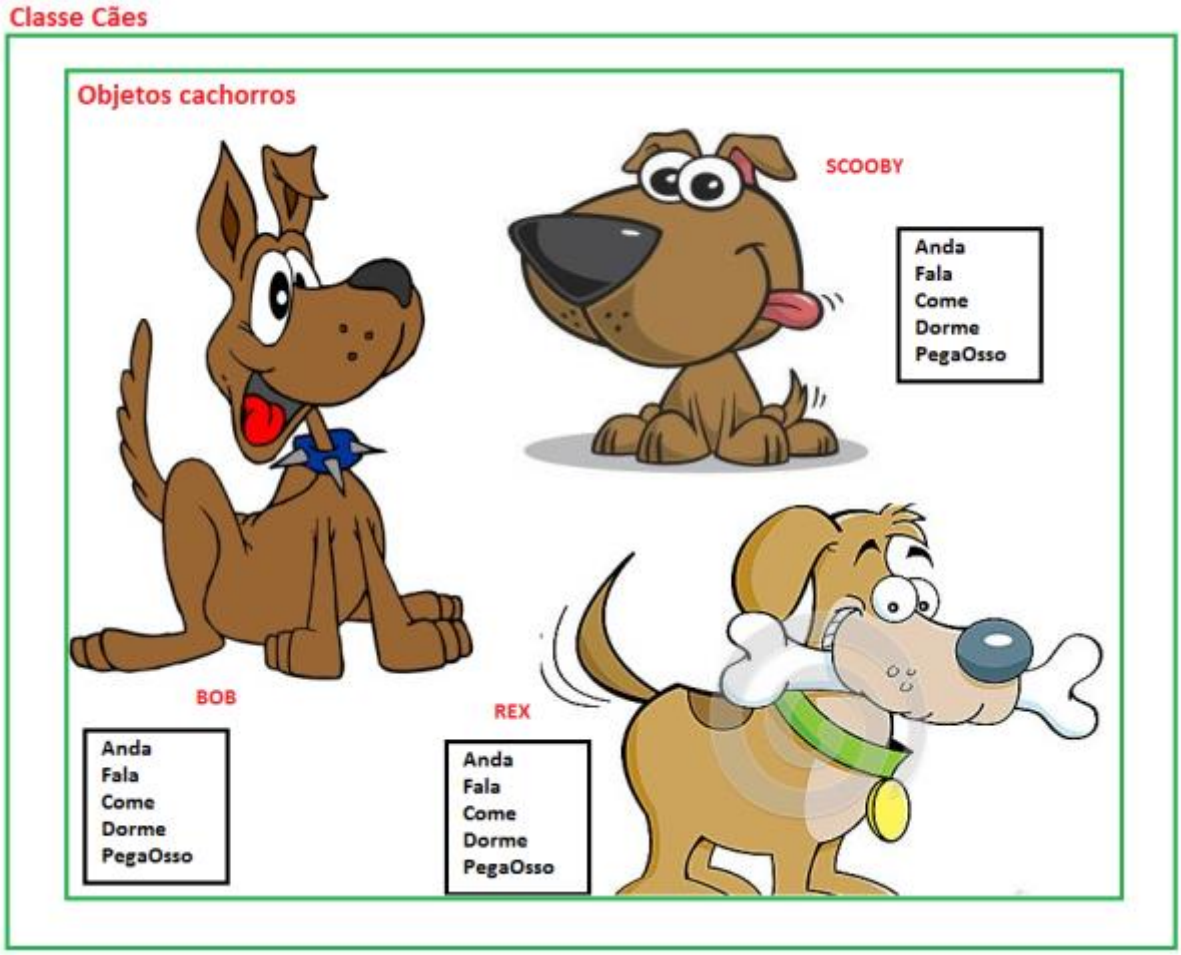

Figura 2: Demonstração da classe Cães
Na Listagem 2 e na Figura 2, mostra que a classe Cães de um modo genérico, tem os mesmos métodos independente de qualquer cachorro, sendo que a classe é sempre um molde/projeto para o objeto cachorro.

Objetos

Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos. Veja no exemplo da Figura 3.

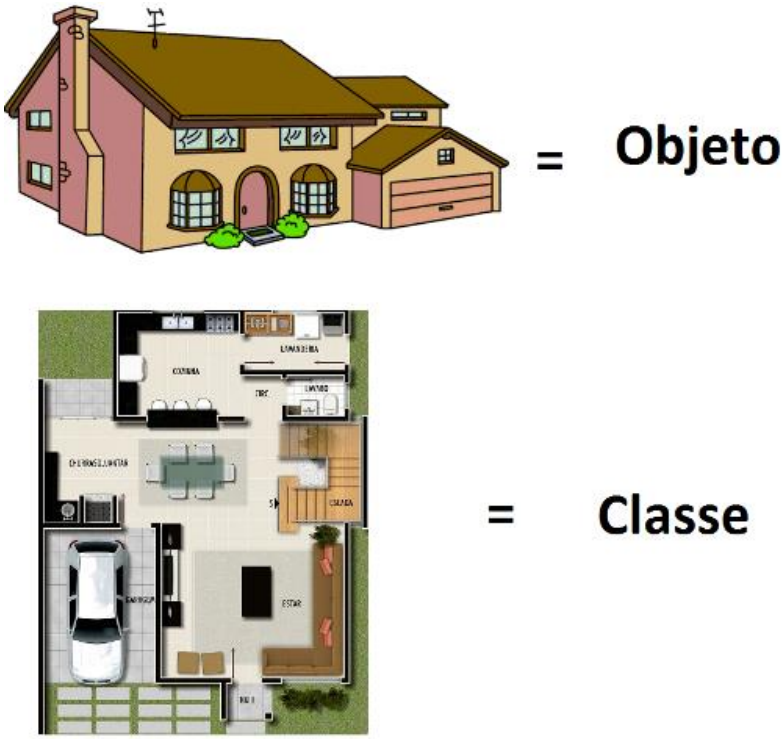

Figura 3: Diferença entre objeto e classe
A Figura 3 mostra que todo objeto é algo que existe, uma coisa concreta, já a classe é considerada como um modelo ou projeto de um objeto, sendo algo que não consegue tocar.

Atributos

Os atributos são as propriedades de um objeto, também são conhecidos como variáveis ou campos. Essas propriedades definem o estado de um objeto, fazendo com que esses valores possam sofrer alterações. A Listagem 3 mostra as características de um cachorro, mas os valores que são guardados nas variáveis são diferentes variando para cada cachorro.
Listagem 3: Classe Cachorro

```
public class Cachorro{

        public String nome;
        public int peso;
        public String corOlhos;
        public int quantPatas;
}
```

Na Listagem 4 é instanciada três vezes a classe “Cachorro”, mostrando que cada cachorro instanciado tem características diferentes.
Listagem 4: Classe Testadora de Cachorro

```
public class TestaCaes {

        public static void main(String[] args) {
                Cachorro cachorro1 = new Cachorro();
```

```
                cachorro1.nome = "Pluto";
                cachorro1.corOlhos = "azuis";
                cachorro1.peso = 53;
                cachorro1.quantPatas = 4;

                Cachorro cachorro2 = new Cachorro();
                cachorro2.nome = "Rex";
                cachorro2.corOlhos = "amarelo";
                cachorro2.peso = 22;
                cachorro2.quantPatas = 3;

                Cachorro cachorro3 = new Cachorro();
                cachorro3.nome = "Bob";
                cachorro3.corOlhos = "marrom";
                cachorro3.peso = 13;
                cachorro3.quantPatas = 4;

        }

}
```

Métodos

Os métodos são ações ou procedimentos, onde podem interagir e se comunicarem com outros objetos. A execução dessas ações se dá através de mensagens, tendo como função o envio de uma solicitação ao objeto para que seja efetuada a rotina desejada.
Como boas práticas, é indicado sempre usar o nome dos métodos declarados como verbos, para que quando for efetuada alguma manutenção seja de fácil entendimento. Veja algumas nomenclaturas de nomes de métodos:

- acaoVoltar
- voltar
- avançar
- correr
- resgatarValor
- pesquisarNomes

Na Listagem 5 são declaradas as características e o método, nota-se que tem uma condição de acordo com o valor informado na variável “tamanho”.
Listagem 5: Classe Cachorro com método

```
class Cachorro{
        int tamanho;
        String nome;


        void latir(){
                if(tamanho > 60)
```

```
                System.out.println("Wooof, Wooof!");
        else if(tamanho > 14)
                System.out.println("Ruff!, Ruff!");
        else
                System.out.println("Yip!, Yip!");
    }
}
```

A Listagem 6 mostra como uma variável pode mudar o estado de um objeto, comunicando-se com o método invocado.

Listagem 6: Classe Testadora

```
public class Testa_Cachorro {

    public static void main(String[] args) {

        Cachorro bob = new Cachorro();
        bob.tamanho = 70;
        Cachorro rex = new Cachorro();
        rex.tamanho = 8;
        Cachorro scooby = new Cachorro();
        scooby.tamanho = 35;

        bob.latir();
        rex.latir();
        scooby.latir();

    }
}
```

Construtores

O construtor de um objeto é um método especial, pois inicializa seus atributos toda vez que é instanciado (inicializado).

Toda vez que é digitada a palavra reservada **new**, o objeto solicita para a memória do sistema armazená-lo, onde chama o construtor da classe para inicializar o objeto. A identificação de um construtor em uma classe é sempre o mesmo nome da classe.

Na Listagem 7, o construtor recebe um parâmetro de uma String que será um argumento de entrada na classe testadora.

Listagem 7: Declaração de um construtor com parâmetro

```
class ConstrutorProg{
    private String nomeCurso;


    public ConstrutorProg(String nome)
    {
        nomeCurso = nome;
```

```
        }

        public String getNome()
        {
                return "Nome do Curso retornado "+nomeCurso;
        }

}
```

Listagem 8: Classe Testadora do Construtor

```
public class Construtor {

        public static void main(String[] args) {

                ConstrutorProg cp = new ConstrutorProg("DevMedia -
Java");
                System.out.println(cp.getNome());

        }
}
```

Na Listagem 8 é inicializada a classe “ConstrutorProg”, passando apenas um argumento no parâmetro que foi definido, se fosse apenas inicializado sem nenhum argumento estaria ocorrendo um erro de sintaxe pois já foi definido que no método construtor iria haver uma entrada de um parâmetro.

Sempre uma classe terá um construtor padrão, mesmo não sendo declarado o compilador irá fornecer um. Esse construtor não recebe argumentos e existe para possibilitar a criação de objetos para uma classe.