

Artigo

Invista em você! Saiba como a DevMedia pode ajudar sua carreira.



Introdução à Programação Orientada a Objetos em Java

Veja neste artigo a história da programação orientada a objetos, conheça quem começou a praticar essa técnica. Também saiba como aplicar esses conceitos na prática, usando a linguagem Java.



Anotar



Marcar como concluído

Artigos



Java



Introdução à Programação Orientada a Objetos em Java

Programação Orientada a Objetos



Nos desenvolvimentos de sistemas, existem alguns fatores importantes como: o entendimento do código, fácil manutenção, reaproveitamento entre outros. Para isso, a **Programação Orientada a Objetos também conhecida como POO**, tem a intenção de ajudar nesses fatores, dando tempo e agilidade no desenvolvimento de um sistema para o programador.

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

Relacionado: [Curso completo de Java](#)

Elementos

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

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

```
1 | public class Teste{  
2 | //ATRIBUTOS OU PROPRIEDADES  
3 | //MÉTODOS  
4 | }
```

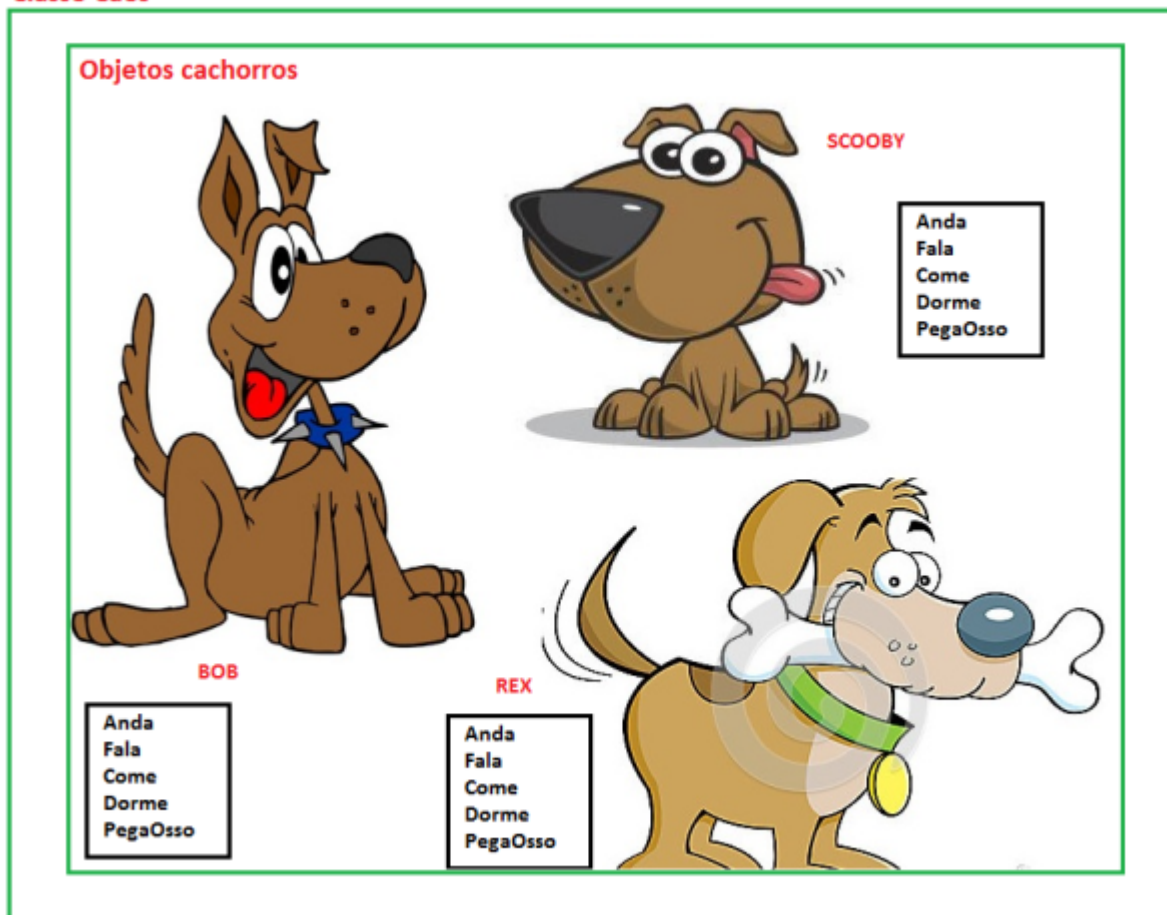
Na Listagem 2, são mostrados os componentes da classe, como métodos e atributos.

Listagem 2: Classe Caes

```
1 | public class Caes {  
2 |  
3 |     public String nome;  
    ...
```

```
7  
8     public void falar(){  
9         //MÉTODO FALAR  
10    }  
11  
12    public void andar(){  
13        //MÉTODO ANDAR  
14    }  
15  
16    public void comer(){  
17        //MÉTODO COMER  
18    }  
19  
20    public void dormir(){  
21        //MÉTODO DORMIR  
22    }  
23 }
```

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 2 mostra que todo objeto é algo que existe, uma coisa concreta, já a

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

```
1 public class Cachorro{  
2  
3     public String nome;  
4     public int peso;  
5     public String corOlhos;  
6     public int quantPatas;  
7 }
```

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

Listagem 4: Classe Testadora de Cachorro

```
1 public class TestaCaes {  
2  
3     public static void main(String[] args) {  
4         Cachorro cachorro1 = new Cachorro();  
5         cachorro1.nome = "Pluto";  
6         cachorro1.corOlhos = "azuis";  
7         cachorro1.peso = 53;  
8         cachorro1.quantPatas = 4;  
9     }
```

```
13     cachorro2.peso = 22;
14     cachorro2.quantPatas = 3;
15
16     Cachorro cachorro3 = new Cachorro();
17     cachorro3.nome = "Bob";
18     cachorro3.corOlhos = "marrom";
19     cachorro3.peso = 13;
20     cachorro3.quantPatas = 4;
21
22 }
23
24 }
```

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”.

```
1  class Cachorro{
2      int tamanho;
3      String nome;
4
5
6      void latir(){
7          if(tamanho > 60)
8              System.out.println("Wooof, Wooof!");
9          else if(tamanho > 14)
10             System.out.println("Ruff!, Ruff!");
11         else
12             System.out.println("Yip!, Yip!");
13     }
14 }
```

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

Listagem 6: Classe Testadora

```
1  public class Testa_Cachorro {
2
3      public static void main(String[] args) {
4
5          Cachorro bob = new Cachorro();
6          bob.tamanho = 70;
7          Cachorro rex = new Cachorro();
8          rex.tamanho = 8;
9          Cachorro scooby = new Cachorro();
10         scooby.tamanho = 35;
11
12         bob.latir();
13         rex.latir();
14         scooby.latir();
15
16     }
17 }
```


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

```
1  class ConstrutorProg{
2      private String nomeCurso;
3
4
5      public ConstrutorProg(String nome)
6      {
7          nomeCurso = nome;
8      }
9
10     public String getNome()
11     {
12         return "Nome do Curso retornado "+nomeCurso;
13     }
14
15 }
```

Listagem 8: Classe Testadora do Construtor

```
1  public class Construtor {
2
3      public static void main(String[] args) {
4
5          ConstrutorProg cp = new ConstrutorProg("DevMedia - Java");
```

```
8 |  
9 | }
```

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.

Espero que tenham gostado e fico por aqui. Até a próxima!

Links Úteis

■ Java 7:

Site com informações sobre o lançamento do Java 7

■ JavaFX:

Site para fazer download de aplicações JavaFX

■ JFXtras:

Site do projeto JFXtras

Saiba mais sobre Java ;)

■ O Que é JPA?:

Dominar a persistência de dados é uma necessidade indispensável aos programadores. Sem esse conhecimento nossas aplicações não terão a capacidade de armazenar e recuperar os dados por ela manipulados.

■ Preparando o ambiente para programar em Java:

Neste curso você aprenderá a preparar seu ambiente para programar em Java. Veremos aqui o que é necessário instalar e como proceder para desenvolver aplicações com essa linguagem.

■ Criando meu primeiro projeto no Java:

Neste curso você aprenderá a criar o seu primeiro programa com Java, e não, ele não será um simples “Hello, World!”. :) Para isso, vamos começar ensinando como instalar o Java e preparar o ambiente de desenvolvimento.

Tecnologias:

Java

POO



Anotar



Marcar como concluído



Confira outros conteúdos:



PARA QUEM QUER SER
PROGRAMADOR DE VERDADE.
VAGAS LIMITADAS

Em caso de dúvidas chame no whatsapp



Plano Anual

12x R\$ 59,90

12 MESES = R\$ 718,80

Formação FullStack completa ✓

+10mil exercícios gamificados ✓

+50 projetos reais ✓

Suporte online ✓

Matricule-se

MELHOR OPÇÃO

Plano Recorrente

R\$89,90/MÊS PRIMEIROS 3 MESES

R\$ 49,90/MÊS A PARTIR DO 4º MÊS

12 MESES = R\$ 718,80

Formação FullStack completa ✓

+10mil exercícios gamificados ✓

+50 projetos reais ✓

Suporte online ✓

Pra quem tem pouco limite no cartão ✓

Fidelidade de 12 meses ✓

Matricule-se

Perguntas Frequentes

Quem somos?



Como faço para começar a estudar?

Em quanto tempo de estudo vou me tornar um programador?

Sim, você pode se tornar um programador e não precisa ter diploma de curso superior!

O que eu irei aprender estudando pela DevMedia?

Principais diferenciais da DevMedia

Qual o investimento financeiro que preciso fazer para me tornar um programador?

Como funciona a forma de pagamento da DevMedia?

Nossos casos de sucesso

Leonardo Carlos



Eu sabia pouquíssimas coisas de programação antes de começar a estudar com vocês, fui me especializando em várias áreas e ferramentas que tinham na plataforma, e com essa bagagem **consegui um estágio logo no início do meu primeiro período na faculdade.**

Lucas Rodrigues



Estudo aqui na Dev desde o meio do ano passado! Nesse período a Dev me

lidero um time de desenvolvimento!

Minha meta é continuar estudando e praticando para ser um Full-Stack Dev!

Heráclito Júnior

Economizei 3 meses para assinar a plataforma e sendo sincero valeu muito a pena, pois a **plataforma é bem intuitiva e muuuuito didática a metodologia de ensino**. Sinto que estou EVOLUINDO a cada dia. Muito obrigado!

Julio Cablen

Nossa! Plataforma maravilhosa. To amando o curso de desenvolvimento front-end, tinha coisas que eu ainda não tinha visto. **A didática é do jeito que qualquer pessoa consegue aprender**. Sério, to apaixonado, adorando demais.

Joelberth Sena

Adquiri o curso de vocês e logo percebi que são os melhores do Brasil. É um passo a passo incrível. **Só não aprende quem não quer. Foi o melhor investimento da minha vida!**

Felipe Nunes

Foi um dos melhores investimentos que já fiz na vida e tenho aprendido bastante com a plataforma. Vocês estão fazendo parte da minha jornada nesse mundo da programação, **irei assinar meu contrato como programador graças a plataforma**.

Wanderson Oliveira

Comprei a assinatura tem uma semana, aprendi mais do que 4 meses estudando outros cursos. Exercícios práticos que não tem como não



José Lucas

Obrigado DevMedia, nunca presenciei **uma plataforma de ensino tão presente na vida acadêmica de seus alunos**, parabéns!

Eduardo Dorneles

Aprendi React na plataforma da DevMedia há cerca de 1 ano e meio... **Hoje estou há 1 ano empregado** trabalhando 100% com React!

Adauto Junior

Já fiz alguns cursos na área e **nenhum é tão bom quanto o de vocês**. Estou aprendendo muito, muito obrigado por existirem. Estão de parabéns... Espero um dia conseguir um emprego na área.

[Ver todos os casos de sucesso](#)

Por **Thiago**
Em 2012



[Guias](#)[Artigos](#)[Quem Somos](#)[Fale conosco](#)[Plano para Instituição de ensino](#)[Assinatura para empresas](#)[Assine agora](#)

Hospedagem web por Porta 80 Web Hosting.



70

