

# JAVA KAGE

OS 4 JITSUS SAGRADOS DO CODIGO NINJA



BIANCA CUNHA

# JUTSU JAVA

## O Caminho Do Código Ninja

Todo ninja precisa dominar suas técnicas básicas antes das grandes missões — e na programação Java, isso significa conhecer os 4 Mandamentos da Programação Ninja. Neste capítulo, você vai aprender os fundamentos da Programação Orientada a Objetos: classe e objeto, encapsulamento, herança e polimorfismo. Esses conceitos são como jutsus secretos que tornam seu código mais organizado, flexível e poderoso. Prepare-se para desbloquear o verdadeiro poder do código ninja!



# 01

---

## CLASSE E OBJETO

# O MOLDE E O CLONE



Uma classe é o molde. Um objeto é a coisa feita a partir desse molde. Como um jutsu de clonagem: a classe é a técnica, o objeto é o clone.

Classes.java

```
public class Ninja {
    String nome;
    String vila;

    void apresentar() {
        System.out.println("Sou o ninja " + nome + " da Vila " + vila);
    }
}

public class Missao {
    public static void main(String[] args) {
        Ninja naruto = new Ninja();
        naruto.nome = "Naruto";
        naruto.vila = "Folha";
        naruto.apresentar();
    }
}
```



02

---

# ENCAPSULAMENTO



# PROTEGENDO SEU CHAKRA



Encapsulamento é esconder os dados e controlar o acesso com getters e setters. Como um pergaminho secreto: só abre com permissão.

```
public void setNome(String nome) {
    this.nome = nome;
}

public String getNome() {
    return this.nome;
}
}

public class Treino {
    public static void main(String[] args) {
        Jutsu jutsu = new Jutsu();
        jutsu.setNome("Rasengan");
        System.out.println("Jutsu aprendido: " + jutsu.getNome());
    }
}
```



03

---

HERANÇA

# PASSANDO O LEGADO NINJA



Com herança, uma classe pode herdar os atributos e métodos de outra. Como o Sharingan sendo passado pelos Uchiha.

```
Herança.java

public class Ninja {
    String nome;

    void lutar() {
        System.out.println(nome + " está lutando!");
    }
}

public class NinjaEspecial extends Ninja {
    void usarJutsu() {
        System.out.println(nome + " usou um jutsu especial!");
    }
}

public class Batalha {
    public static void main(String[] args) {
        NinjaEspecial kakashi = new NinjaEspecial();
        kakashi.nome = "Kakashi";
        kakashi.lutar();
        kakashi.usarJutsu();
    }
}
```





# 04

---

## POLIMORFISMO



# UM JUTSU, VARIAS FORMAS



Polimorfismo é quando métodos se comportam de forma diferente, mesmo com o mesmo nome. Como um jutsu que muda de forma conforme o usuário.

```
Polimorfismo.java

public class Ninja {
    void atacar() {
        System.out.println("Ataque básico!");
    }
}

public class NinjaFogo extends Ninja {
    void atacar() {
        System.out.println("Ataque com jutsu de fogo!");
    }
}

public class Treinamento {
    public static void main(String[] args) {
        Ninja n1 = new Ninja();
        Ninja n2 = new NinjaFogo(); // polimorfismo

        n1.atacar();
        n2.atacar(); // executa a versão da subclasse
    }
}
```



# AGRADECIMENTOS

---



# CODIGO DE GRATIDAO DO NINJA



Parabéns, jovem shinobi do código!

Se você chegou até aqui, é porque enfrentou bugs, compilou com coragem e dominou os mandamentos da Programação Ninja com determinação digna de um Hokage.

Este conteúdo foi forjado com o auxílio de uma inteligência artificial (IA) — como um clone das sombras digital — e revisado por um humano com olhos atentos, mas sem uma validação profunda.

Ou seja, este eBook é um treinamento didático, criado para ajudar na jornada de aprendizado, e pode conter falhas ou jutsus mal lançados (vulgo erros gerados pela IA).

**NUNCA SUBESTIME O PODER DE UM PROGRAMADOR NINJA.**

