SENAI SÃO PAULO

SENAI — SP

Prof. Vedilson Prado

AULA 03

PILARES POO - ABSTRAÇÃO E ENCAPSULAMENTO

 Prof. Vedilson

SENAI

Programação Orientada a Objetos

Polimorfismo

Herança

Abstração

Encapsulamento

SENAI

ABSTRAÇÃO

ENCAPSULAMENTO

HERANÇA

POLIMORFISMO

ABSTRAÇÃO

Conceito: A abstração é o processo de **destacar apenas as características essenciais** de um objeto no mundo real e representá-las no sistema. Ou seja, nós não nos preocupamos com os detalhes internos, apenas com o que realmente importa para o contexto.

👉 Mostrar apenas o que é importante, escondendo detalhes internos.

SENAI

ABSTRAÇÃO

ABSTRAÇÃO

- nome | string
- hp | int
- chacra | int
- categoria | string

- jutsu(nomeJutsu)

SENAI

ABSTRAÇÃO

CLASSE

Instanciação

É quando você cria um espaço na memória da sua aplicação utilizando uma base.

OBJETO

SENAI

ABSTRAÇÃO

CLASSE NINJA
nome
hp
chacra
categoria

OBJETO
Naruto
100
50
Genin

OBJETO
Jiraya
300
200
Sannin Lendário

ABSTRAÇÃO

```
class Ninja {
    String nome;
    String aldeia;
    int chakra;

    void usarJutsu(String jutsu) {
        System.out.println(nome + " usou " + jutsu + "!");
    }
}
```

ABSTRAÇÃO

```
public class Main {
    public static void main(String[] args) {
        Ninja naruto = new Ninja(); // instanciando
        naruto.nome = "Naruto Uzumaki";
        naruto.aldeia = "Folha";
        naruto.chakra = 100;

        naruto.usarJutsu("Rasengan");
    }
}
```

CONSTRUTOR

Conceito: Um construtor é um método especial dentro de uma classe que é chamado automaticamente quando um objeto dessa classe é criado. Ele serve para inicializar os atributos do objeto, ou seja, atribuir valores iniciais às suas propriedades.

ABSTRAÇÃO

```
class Ninja {
    String nome;
    String aldeia;
    int chakra;

        public Ninja(String nome, String aldeia, int chakra) {
        this.nome = nome;
        this.aldeia = aldeia;
        this.chakra = chakra;
    }
}
```

ABSTRAÇÃO

```
public class Main {
    public static void main(String[] args) {
        // Instanciando com construtor
        Ninja naruto = new Ninja("Naruto Uzumaki", "Folha",
100);

        naruto.usarJutsu("Rasengan");
    }
}
```

SENAI

ABSTRAÇÃO

ENCAPSULAMENTO

HERANÇA

POLIMORFISMO

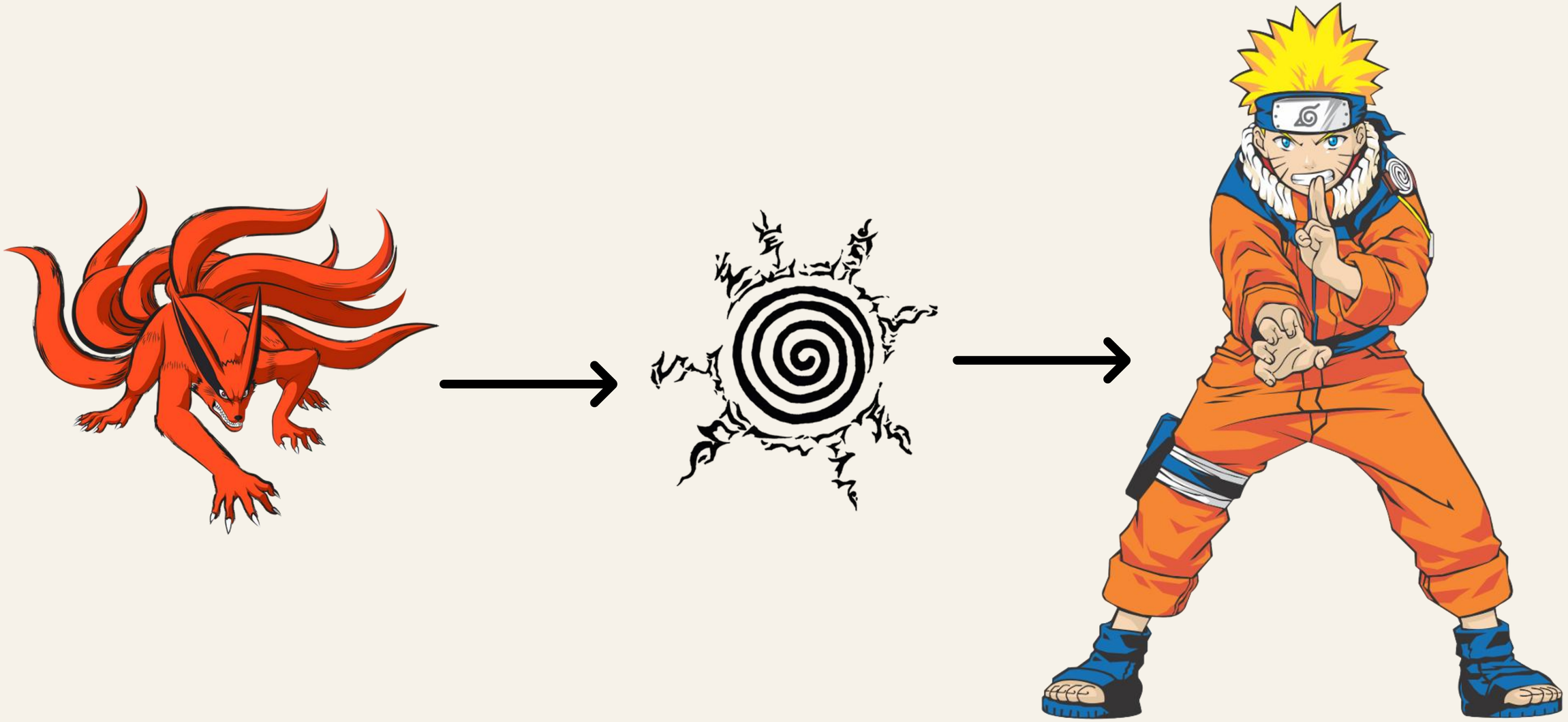
ENCAPSULAMENTO

Conceito: O encapsulamento é o pilar que trata da **proteção dos dados internos de um objeto**, permitindo que sejam acessados apenas de forma controlada. Isso significa que os atributos não ficam totalmente expostos: usamos métodos (getters e setters) para gerenciar como eles podem ser consultados ou alterados.

👉 Proteger os dados internos da classe, permitindo acesso controlado.

SENAI

ABSTRAÇÃO



ABSTRAÇÃO

```
class Ninja {
                String nome;
                String aldeia;
                int chakra;
    private int chakraKurama 100;

    void usarJutsu(String jutsu) {
        System.out.println(nome + " usou " + jutsu + "!");
    }
}
```

ABSTRAÇÃO

```
    public void usarKurama() {
        if (chakra > 50) {
            System.out.println("Naruto usa o poder da
Kurama!");
            chakraKurama -= 50;
        } else {
            System.out.println("Chakra insuficiente!");
        }
    }
}
```

ABSTRAÇÃO

```
public class Main {
    public static void main(String[] args) {
        // Instanciando com construtor
        Ninja naruto = new Ninja("Naruto Uzumaki", "Folha",
100);

        naruto.usarJutsu("Rasengan");
        naruto.usarKurama();
        naruto.usarKurama();
    }
}
```

GETTERS E SETTERS

Conceito: Getters e setters são métodos especiais em Java que permitem controlar o acesso aos atributos de uma classe. Eles são fundamentais para o conceito de encapsulamento, que visa proteger os dados internos de uma classe e garantir a integridade dos objetos.

O que fazem os getters e setters?

- Getter (get): Retorna o valor atual de um atributo privado. É como se fosse um "pegar" o valor.
- Setter (set): Atribui um novo valor a um atributo privado. É como se fosse um "definir" um novo valor.

ABSTRAÇÃO

```
class Ninja {
    // Atributos privados (encapsulamento)
    private String nome;
    private String aldeia;
    private int chakra;

    // Construtor
    public Ninja(String nome, String aldeia, int chakra) {
        this.nome = nome;
        this.aldeia = aldeia;
        this.chakra = chakra;
    }
```

ABSTRAÇÃO

```
// Getters e Setters
    public String getNome() {
        return nome;
    }

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

    public String getAldeia() {
        return aldeia;
    }

    public void setAldeia(String aldeia) {
        this.aldeia = aldeia;
    }

    public int getChakra() {
        return chakra;
    }

    public void setChakra(int chakra) {
        this.chakra = chakra;
    }
```

ABSTRAÇÃO

```
    // Método da classe
    public void usarJutsu(String jutsu) {
        if (chakra > 0) {
            System.out.println(nome + " usou " + jutsu + "!");
            chakra -= 10; // cada jutsu gasta chakra
        } else {
            System.out.println(nome + " está sem chakra!");
        }
    }
}
```

ABSTRAÇÃO

```
public class Main {
    public static void main(String[] args) {
        // Instanciação com construtor
        Ninja naruto = new Ninja("Naruto Uzumaki", "Folha", 100);

        // Acessando informações com getters
        System.out.println("Ninja: " + naruto.getNome());
        System.out.println("Aldeia: " + naruto.getAldeia());
        System.out.println("Chakra inicial: " + naruto.getChakra());

        // Usando método
        naruto.usarJutsu("Rasengan");
        System.out.println("Chakra restante: " + naruto.getChakra());
    }
}
```


SENAI
DEPARTAMENTO REGIONAL
DE SÃO PAULO
www.sp.senai.br