Estruturas de Programação

AGSI Sistemas
Agosto / 2021

Anteriormente ...

Como vimos na apresentação anterior:

- Visão geral do ambiente WEB;
- Estrutura do Ambiente WEB;
- Apache Tomcat;
- Java;
- Linguagens;
- Microsserviços.

Índice

Introdução

Com a necessidade de entrega de softwares de forma rápida e com alta qualidade, os paradigmas de programação estruturada precisaram mudar, para conseguir se adequar com as entregas.

A alta complexidade da estrutura dos negócios também começou a fazer com que a programação estruturada não fosse mais utilizada, já que a programação e a manutenção ficaram mais complexas.

Programação Orientada a Objetos

A Programação Orientada a Objetos (POO) é um padrão de desenvolvimento de softwares largamente utilizado em muitas linguagens de programação.

Nesse processo de programação, são criadas coleções de objetos com estrutura e comportamentos próprios. Tais objetos interagem entre si e executam as ações solicitadas.

O objetivo da POO é aproximar o mundo real do mundo virtual e promover a unificação de dados e processos, gerando o reuso de códigos.

Princípios da POO

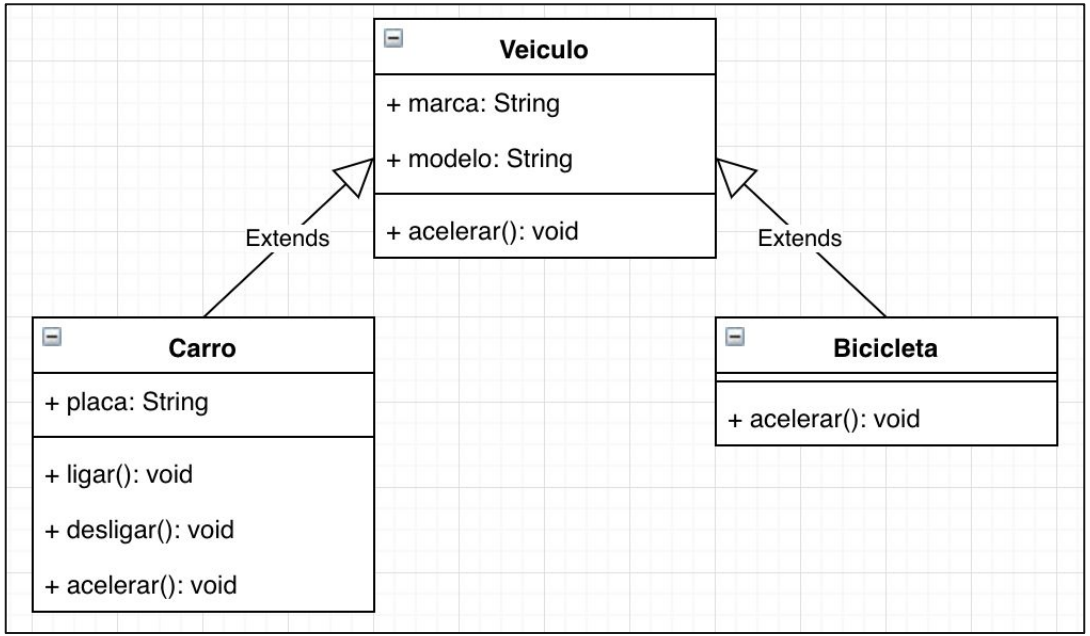
- **Encapsulamento:** Permitir que um objeto seja funcional, porém sem exibir o como ele faz para funcionar. Dessa forma ele se mantém invisível para o mundo externo.
- **Herança:** Permite o reaproveitamento de código, pois permite que um objeto filho tenha características e atributos que o objeto mãe tem.
- **Polimorfismo:** Capacidade de um método ser utilizado por diferentes objetos.
- **Abstração:** Representar um objeto de forma abstrata, que será implementado em classes que deverão implementar essa interface (interface de comunicação).

Vantagens da POO

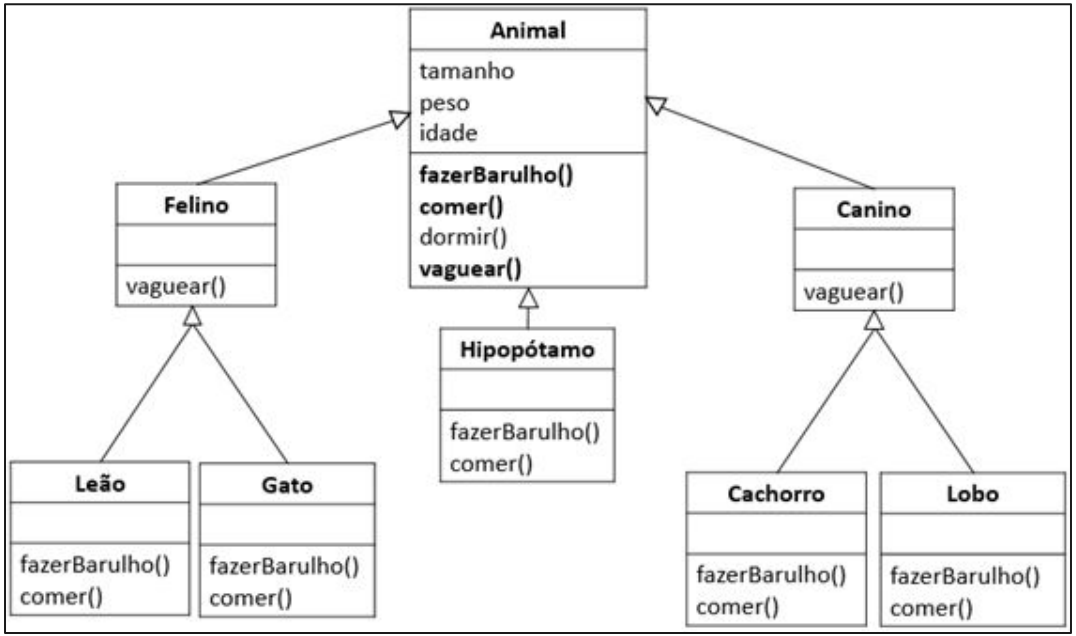
- **Confiável:** Independência entre os objetos dentro do sistema.
- **Oportuno:** Aumento da efetividade da programação, múltiplas pessoas atuando em diferentes partes.
- **Ajustável:** Permite que manutenções em certas partes possam ser reaproveitadas para outras modalidades do sistema.
- **Extensível:** Reúso de software para diferentes funcionalidades dentro do sistema.
- **Reutilizável:** Uma pequena parte de software pode ser reaproveitada para diferentes partes do sistema se tiverem a mesma funcionalidade (Aluno e Professor herdam de pessoa).
- **Natural:** Proximidade com o mundo real.

Vamos ver na prática ...

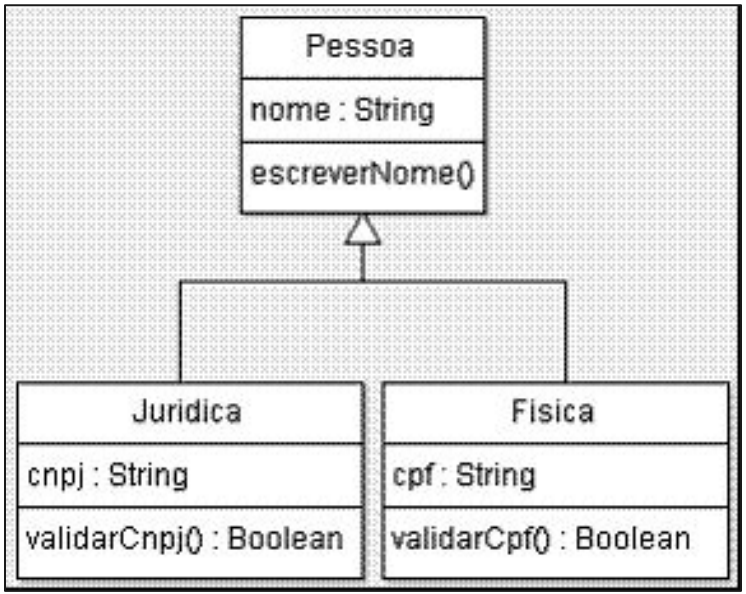
Exemplo 01


Veiculo
+ marca: String
+ modelo: String
+ acelerar(): void
Extends
Extends
Carro
+ placa: String
+ ligar(): void
+ desligar(): void
+ acelerar(): void
Bicicleta
+ acelerar(): void

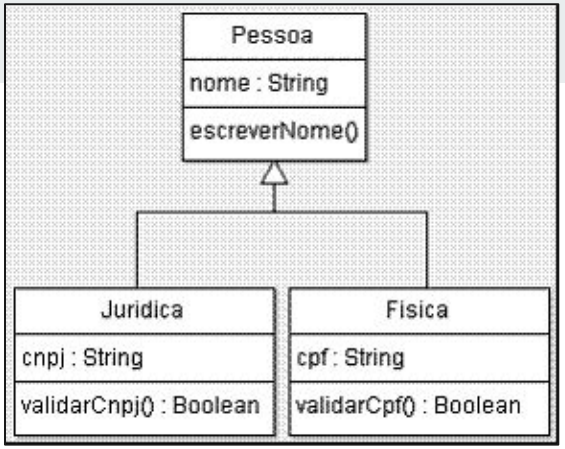
Exemplo 02


Animal
tamanho
peso
idade
fazerBarulho()
comer()
dormir()
vaguear()
Felino
vaguear()
Canino
vaguear()
Hipopótamo
fazerBarulho()
comer()
Leão
fazerBarulho()
comer()
Gato
fazerBarulho()
comer()
Cachorro
fazerBarulho()
comer()
Lobo
fazerBarulho()
comer()

Exemplo 03


Pessoa
nome : String
escreverNome()
Juridica
cnpj : String
validarCnpj() : Boolean
Fisica
cpf : String
validarCpf() : Boolean

Estrutura do Exemplo 03



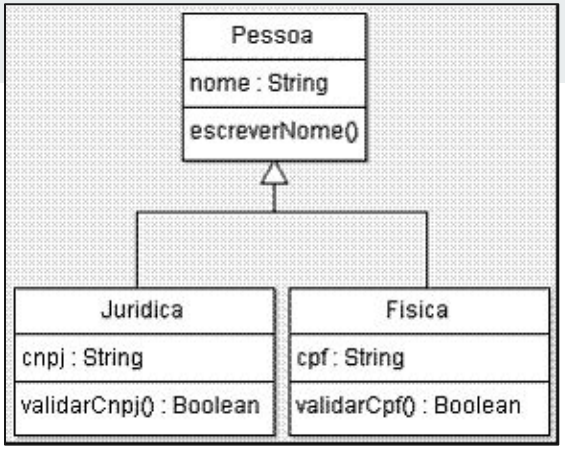

```java
// A classe é chamada de pessoa
public class Pessoa{
    // A pessoa tem um atributo/característica nome
    private String nome;

    // A pessoa tem um construtor
    public Pessoa(String nome){
        this.nome = nome; // A pessoa ganha um nome
    }

    // Ações/métodos que a pessoa consegue fazer
    public void escreverNome(){
        System.out.println("O meu nome é " + this.nome + ".");
    }

}
```

Estrutura do Exemplo 03

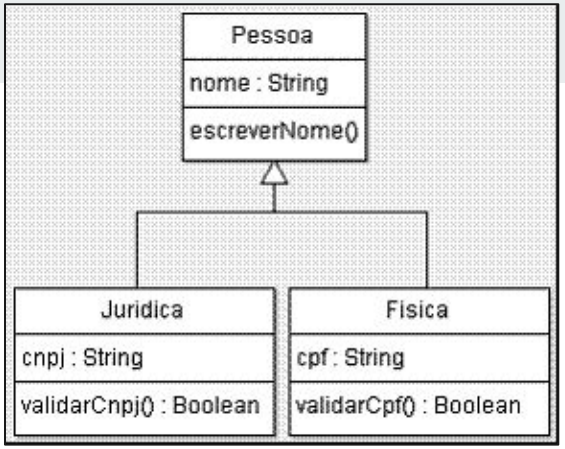



```java
public class Juridica extends Pessoa{
    private String cnpj;

    public Juridica(String nome, String cnpj){
        super(nome)
        this.cnpj = cnpj;
    }

    public Boolean validarCnpj(String cnpj){
        if(cnpj == this.cnpj){
            System.out.println("O CNPJ é válido!");
        }
    }
}
```

Estrutura do Exemplo 03

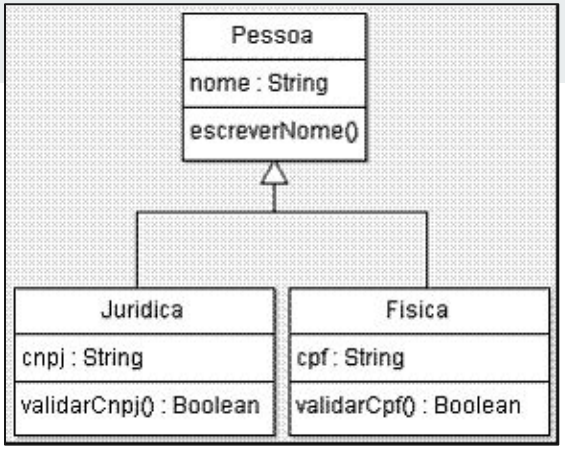



```java
public class Fisica extends Pessoa{
    private String cpf;

    public Fisica(String nome, String cpf){
        super(nome)
        this.cpf = cpf;
    }

    public Boolean validarCpf(String cpf){
        if(cpf == this.cpf){
            System.out.println("O CPF é válido!");
        }
    }
}
```

Estrutura do Exemplo 03




```java
public class Principal{
    public static void main(String args[]){
        // Falo que vão existir essas pessoas
        Pessoa pessoa01, pessoa02;
        Fisica pessoaFisica01, pessoaFisica02;
        Juridica pessoaJuridica01, pessoaJuridica02;

        // Falo quem são as pessoas
        pessoa01 = new Pessoa("Bob da Silva Sauro");
        pessoa01.escreverNome();
        pessoa02 = new Pessoa("Dino da Silva Sauro");
        pessoa02.escreverNome();

        // Falo quem serão as pessoas físicas
        pessoaFisica01 = new Fisica("Baby da Silva Sauro","111.222.333-44");
        pessoaFisica01.escreverNome();
        pessoaFisica01.validarCpf("111.222.333-44");
        pessoaFisica02 = new Fisica("Marlene da Silva Sauro","555.666.777-88");

        // Falo quem serão as pessoas jurídicas
        pessoaJuridica01 = new Juridica("Baby da Silva Sauro S.A.", "11.222.333/4444-55");
        pessoaJuridica01.escreverNome();
        pessoaJuridica01.validarCnpj("11.222.333/4444-55");
        pessoaJuridica02 = new Juridica("Era da Pedra S.A.", "99.888.777/6666-55");
    }
}
```

Alguns Cuidados

"Nove pessoas não podem fazer um bebê em um mês."

Fred Brooks

"Antes do software poder ser reutilizável ele primeiro tem de ser utilizável."

Ralph Johnson

"A arte de programar consiste em organizar e dominar a complexidade."

Edsger W. Dijkstra

"Há duas formas de construir um projeto de software: Uma maneira de fazer isso deve ser tão simples que, obviamente, não deixem deficiências, e a outra forma é a de torná-lo tão complicado que não percebam as evidentes deficiências. O primeiro método é muito mais difícil."
CAR Hoare

Conclusão

O uso de P.O.O. é essencial hoje em dia. Desenvolver projetos de software usando uma linguagem que não possua essas funções, dificulta muito o processo de desenvolvimento e de manutenções futuras. Porém, em primeiro lugar, deve-se entender a proposta para somente então se projetar da forma correta.

Próximos Passos ...

- Linux, Linux, Linux ... (agora vai ...)
- Aguarde, tem mais ...

Obrigado