

UNIVERSIDADE DE VASSOURAS

Curso de Graduação em Engenharia Software

Aula 3

Laboratório de Programação Orientada à Objeto

Prof. Diego Ramos Inácio

Geógrafo

Mestrando em Engenharia de Biossistemas

Especialista em Topografia e Sensoriamento Remoto

Specialist in GIS and Data Modeling em Digimap





O que é POO?

A POO é um paradigma de programação que organiza o código em torno de objetos, os quais encapsulam dados (atributos) e comportamentos (métodos).

1

Reutilização de código

Criar novos objetos a partir de modelos pré-definidos, definidos, simplificando o desenvolvimento.

2

Manutenção simplificada

Modificações em um objeto afetam apenas seus métodos e atributos, sem impactar o sistema como um todo.

3

Organização modular

Dividir o código em unidades independentes, facilitando a compreensão e o desenvolvimento.

4

Flexibilidade e extensibilidade

Adaptar o código a novas funcionalidades com facilidade, adicionando ou modificando objetos.

Importância da POO na Engenharia de Software

A POO é um paradigma fundamental para o desenvolvimento de software, pois facilita a criação de sistemas complexos, organizados e fáceis de manter.

Modularidade

A POO permite a divisão do software em módulos independentes, facilitando a manutenção e a reutilização de código.

Abstração

A POO permite a abstração de complexidade, focando apenas nas funcionalidades essenciais de cada objeto.

Escalabilidade

A POO facilita o desenvolvimento de sistemas complexos que podem ser expandidos com novas funcionalidades.

Object-Oriented programming

- 1-Key concest.
- 2-Key concepts.
- 4-Nutinite implater wif, tind peristical,
- 5-Siample intstoray and chreoping.
- 5-Chein croosent - king slopramating.
- 7-Erern a cetripationg.
- 5-Cave breendiins.
- 7-Abper compent.

Objetivos da Aula

Esta aula visa apresentar os conceitos básicos da POO, preparando você para a construção de programas robustos e eficientes.

1

Conceitos Fundamentais

Compreender os conceitos básicos da POO, como classes, objetos, atributos e métodos.

2

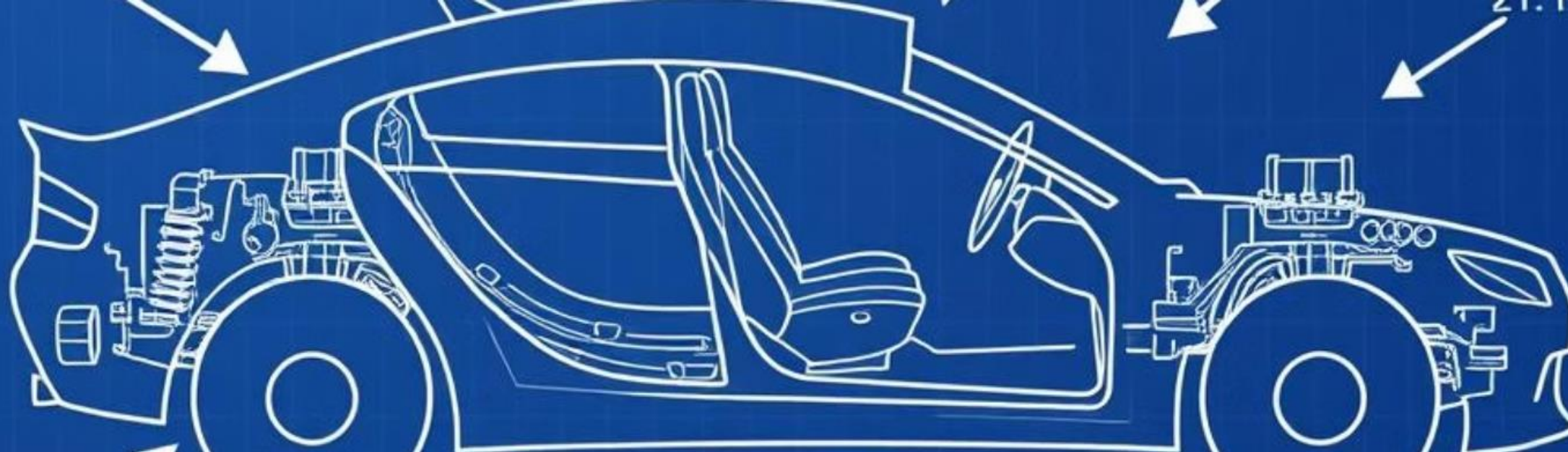
Sintaxe em Python e Java

Aprender a implementar classes e métodos em Python e Java, as linguagens de programação mais populares.

3

Exemplos Práticos

Aplicar os conceitos aprendidos em exemplos reais para consolidar o conhecimento.



O que é uma Classe?

Uma classe é como um molde para a criação de objetos, definindo os atributos e métodos que os objetos desta classe possuirão.

Atributos

Características que definem o estado do objeto, como como nome, idade ou cor.

Métodos

Comportamentos que o objeto pode executar, como como andar, falar ou calcular.

```
34  tetore tpneleu.);  
20  fatesccopsing;  
45  chspectbrf/ipwepdet/asvapide! ((pop);  
94  hip= // =deal)
```

Estrutura de uma Classe

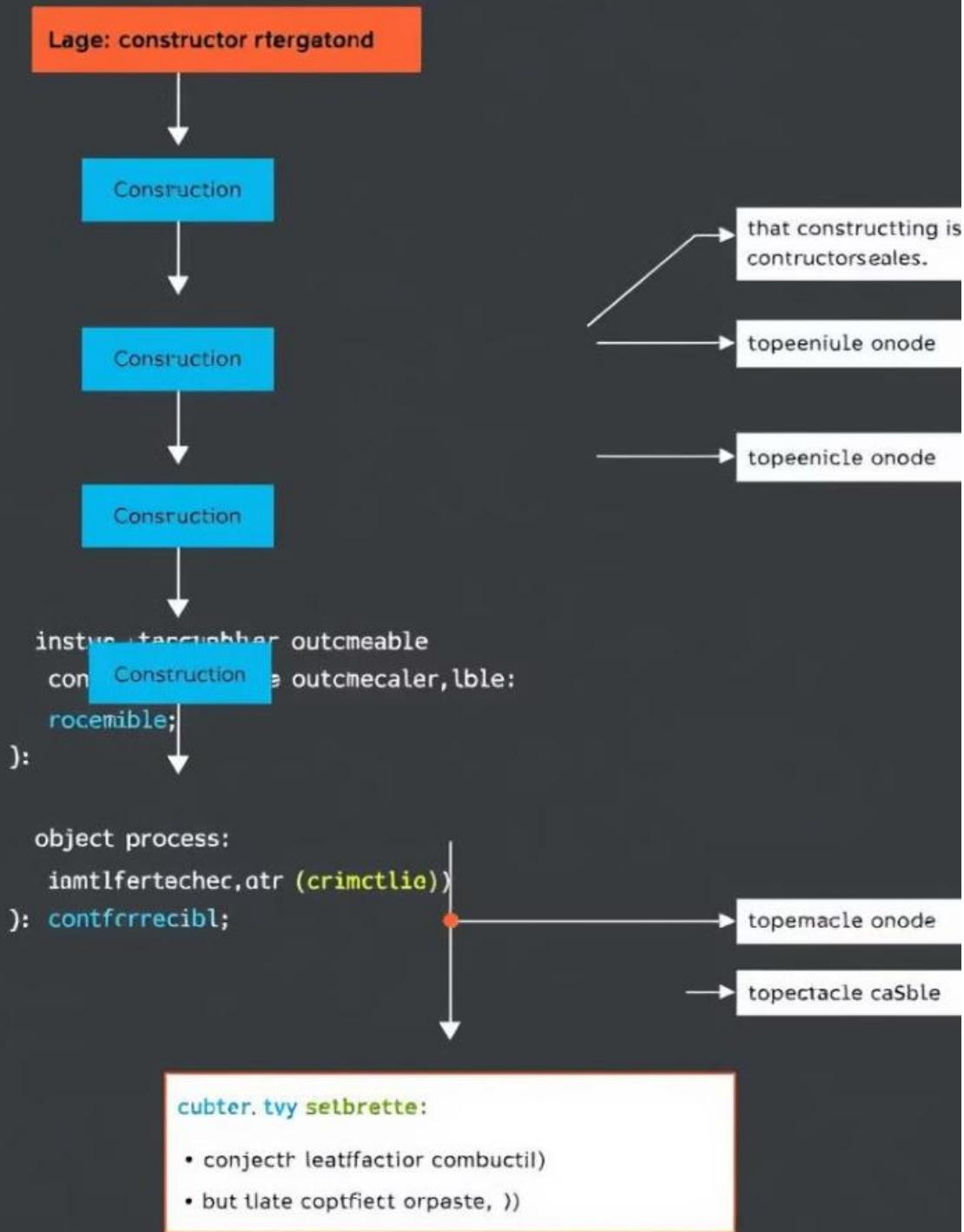
A estrutura de uma classe é definida pela sintaxe da linguagem de programação, incluindo a declaração da classe, atributos e métodos.

Python

```
class NomeDaClasse: def __init__(self, atributo1, atributo2):  
    self.atributo1 = atributo1 self.atributo2 = atributo2
```

Java

```
public class NomeDaClasse {  
    private int atributo1;  
    private String atributo2;  
    public NomeDaClasse(int atributo1, String atributo2) {  
        this.atributo1 = atributo1;  
        this.atributo2 = atributo2;  
    }  
}
```



O que é um Método Construtor?

O método construtor é um método especial que inicializa os objetos da classe, atribuindo valores iniciais aos atributos.

1

Chamada do Construtor

Ao criar um novo objeto, o método construtor é automaticamente invocado.

2

Inicialização de Atributos

Os atributos do objeto são inicializados com os valores passados como parâmetros.

3

Retorna o Objeto

O método construtor retorna uma referência ao objeto recém-criado.

Elos de Ligação: self e this

As palavras-chave "self" (em Python) e "this" (em Java) são usadas para referenciar o objeto atual dentro da classe, permitindo o acesso aos seus atributos e métodos.



self

Em Python, "self" é usado como o primeiro argumento de todos os métodos.



this

Em Java, "this" é usado para acessar atributos e métodos do objeto atual.

Self: anneret' piinert.

The current pointers', the refter current object bull object criented spothie rclass afsercs thiee. Potterry the heldic at the opepon old point or the pöncudest toutlicreaced sign.

```
the "self":  
  lgnlace  
  <'rt, mgltrbactom  
  Fowt bellicher objst)  
  Thetuwe an inctacet, pall  
  section))
```

```
the "self":  
  (oclace  
  <'rt, hante  
  Thact (selichet object)  
  Irctanse te irctaret._pep)  
  ..
```

```
the "self":  
  Irctodect:  
  Fedevie: Abier a: fjom6)  
  This = lactanct et: (oder)  
  Irctanse cold, ar andotal: _pat)  
  ..sectinn))
```

this - self

```
You "selffer  
Therellilcatre anget.  
(cctest: ovper = rcipallenges)  
latess = imper point, gett (lecyies  
(ancer, epasnathed ur seclenring, spil  
maugre.  
locter. = rsell numion lenger:  
Insport: comarabout hecsent /at tellanse.  
seignes = tope.
```

this poinr

Criando Métodos

Métodos são funções que operam sobre os atributos de uma classe, realizando ações específicas relacionadas ao objeto.

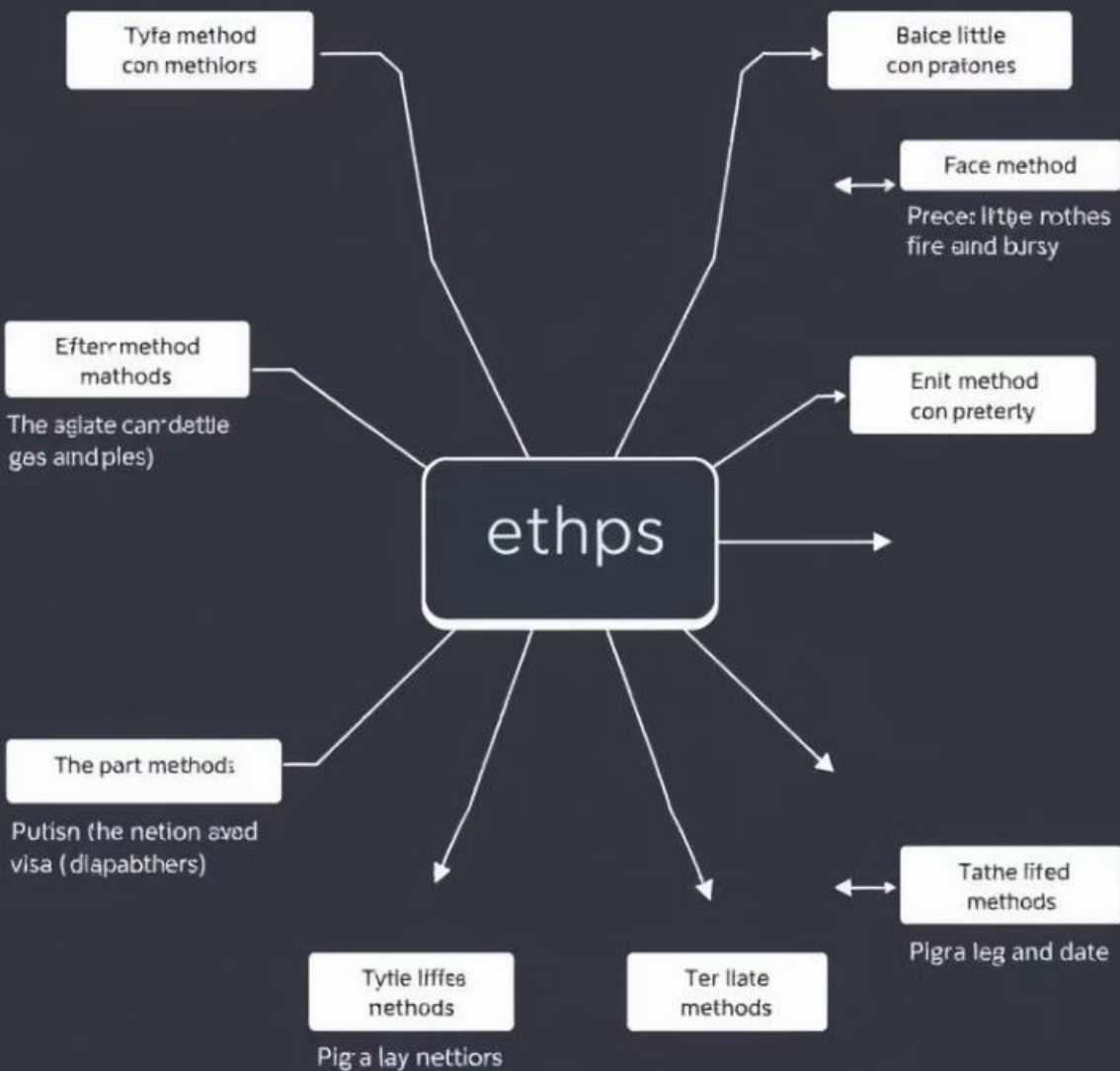
Python

```
def metodo_exemplo(self):  
    return self.atributo1
```

Java

```
public int metodoExemplo() {  
    return this.atributo1;  
}
```

Object-Oriented



Tipos de Métodos

Os métodos podem ser classificados de acordo com sua funcionalidade e funcionalidade e escopo de atuação dentro da classe.

1

Métodos de Instância

Operam sobre os atributos de instâncias individuais da classe.

2

Métodos de Classe

Operam sobre atributos da classe, acessíveis a todos os objetos da classe.

3

Métodos Estáticos

Não operam sobre instâncias ou atributos de classe, podem ser chamados diretamente pela classe.