



Interativa

Unidade I

PROGRAMAÇÃO ORIENTADA A OBJETOS I

Prof. Cassiano Gunji



Paradigmas de desenvolvimento de *software*

Os paradigmas de desenvolvimento de *software* envolvem:

- análise de sistemas;
- linguagens de programação.

Os paradigmas de desenvolvimento são:

- paradigma não estruturado;
- paradigma estruturado;
- paradigma orientado a objetos.



Classes e objetos



Receita
de bolo



Bolo
1



Bolo
2



Bolo
3

Unified Modeling Language (UML) – Linguagem de Modelagem Unificada

Diagramas comportamentais

- Diagrama de casos de uso
- Diagrama de atividades
- Diagrama de máquina de estados
- Etc.

Diagramas estruturais

- Diagrama de classes
- Diagrama de objetos
- Diagrama de pacotes
- Etc.

Diagramas de interação

- Diagrama de sequência
- Diagrama de comunicação
- Diagrama de tempos
- Etc.



Diagrama de classes

NomeDaClasse
+ atributo1: int + atributo2: double + atributo3: string
+ metodo1 (): void + metodo2 (): int + metodo3 (parametro:string): void + metodo4 (parametro:double): string

Abstraindo classes

Luminária

- + Tensão
- + NúmeroDeLâmpadas
- + Ligar()
- + Desligar()

Automóvel

- + Fabricante
- + Modelo
- + Ano
- + Placas
- + Acelerar()
- + Buzinar()
- + AbrirPorta()
- + FecharPorta()

Interatividade

Qual a mudança proposta pelo paradigma orientado a objetos que o torna diferente do paradigma estruturado?

- a) A prática do reúso (reaproveitamento) de código, diminuindo o retrabalho.**
- b) Uso de desvios incondicionais, que permitem a elaboração de código mais fácil de ser entendido, tornando sua manutenção mais simples.**
- c) O tratamento simultâneo de dados e comportamentos que modificam estes dados.**
- d) Uso de herança entre classes.**
- e) Uso de polimorfismo.**



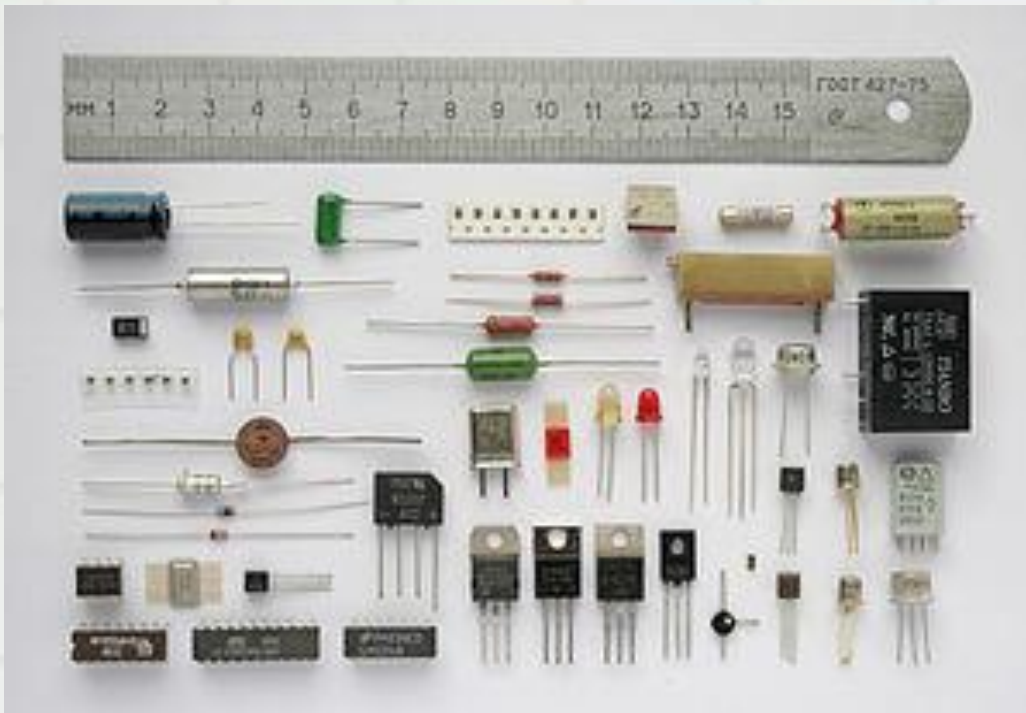
Resposta

Qual a mudança proposta pelo paradigma orientado a objetos que o torna diferente do paradigma estruturado?

- a) A prática do reúso (reaproveitamento) de código, diminuindo o retrabalho.
- b) Uso de desvios incondicionais, que permitem a elaboração de código mais fácil de ser entendido, tornando sua manutenção mais simples.
- c) O tratamento simultâneo de dados e comportamentos que modificam estes dados.
- d) Uso de herança entre classes.
- e) Uso de polimorfismo.



Encapsulamento



Fonte: Arquivo pessoal

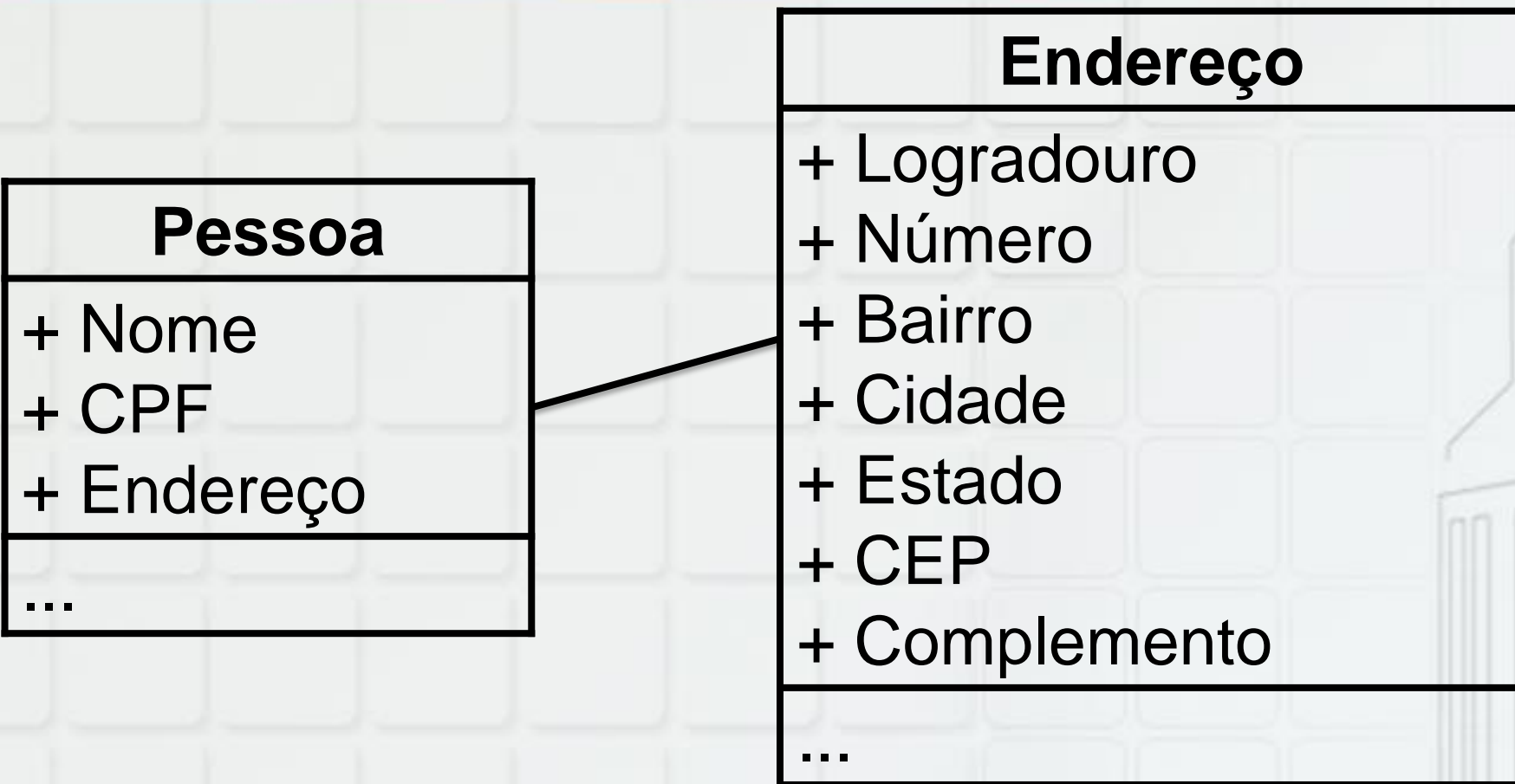
Encapsulamento

Se uma classe abstrai um determinado conceito:

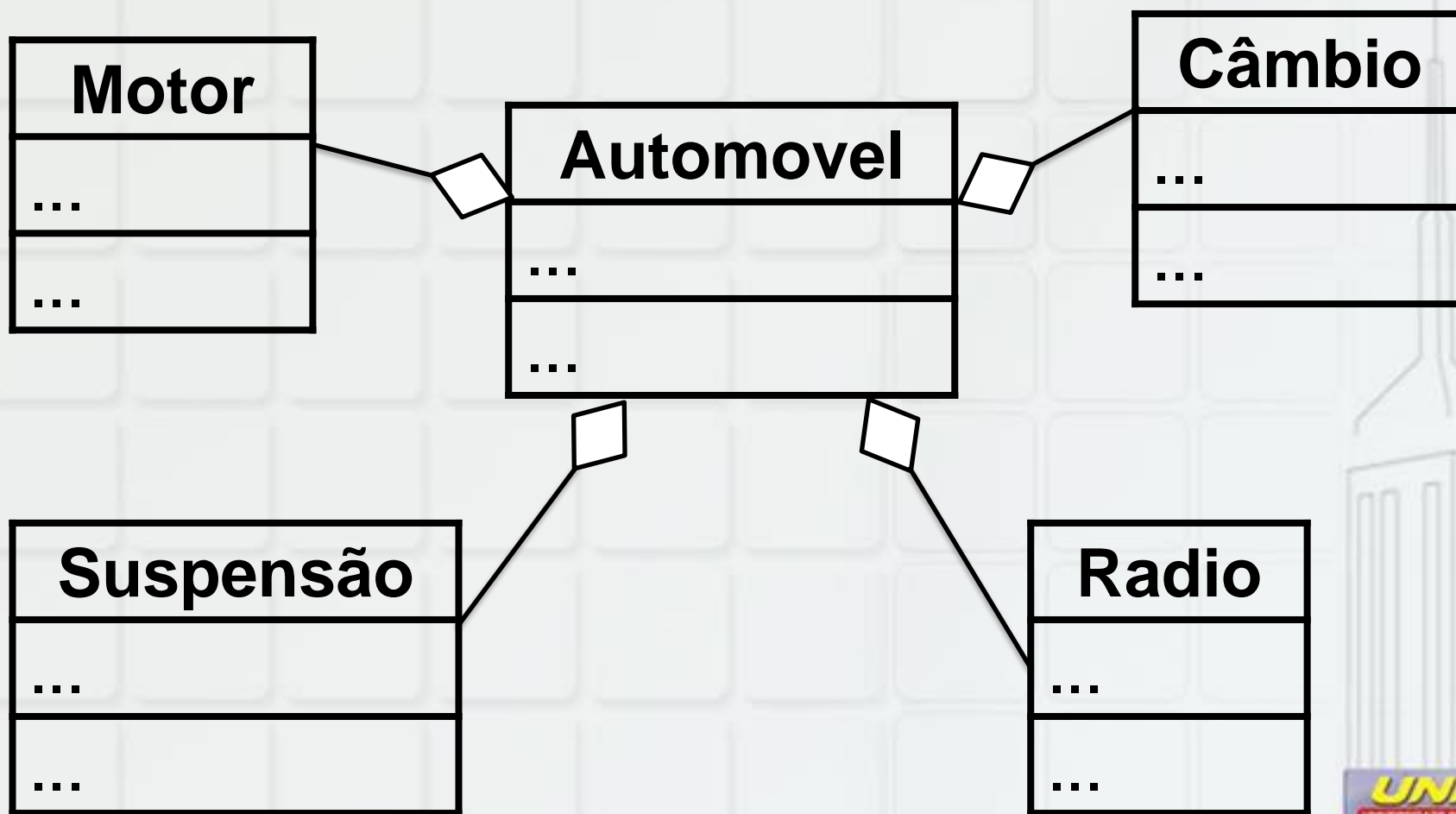
- **a classe só deve conter atributos e métodos que dizem respeito a esse conceito;**
- **se um método ou atributo diz respeito a este conceito, ele deve estar nesta classe.**



Associação



Agregação



Interatividade

Se estivermos modelando uma classe para representar um automóvel de passeio, qual dos seguintes atributos não seria apropriado?

- a) Placa.
- b) NúmeroDeChassis.
- c) NomeDoProprietário.
- d) Quilometragem.
- e) CnhDoProprietário.



Resposta

Se estivermos modelando uma classe para representar um automóvel de passeio, qual dos seguintes atributos não seria apropriado?

- a) Placa.
- b) NúmeroDeChassis.
- c) NomeDoProprietário.
- d) Quilometragem.
- e) **CnhDoProprietário.**



Herança

Cliente

- + Nome
- + CPF
- + Telefone
- + Ativo

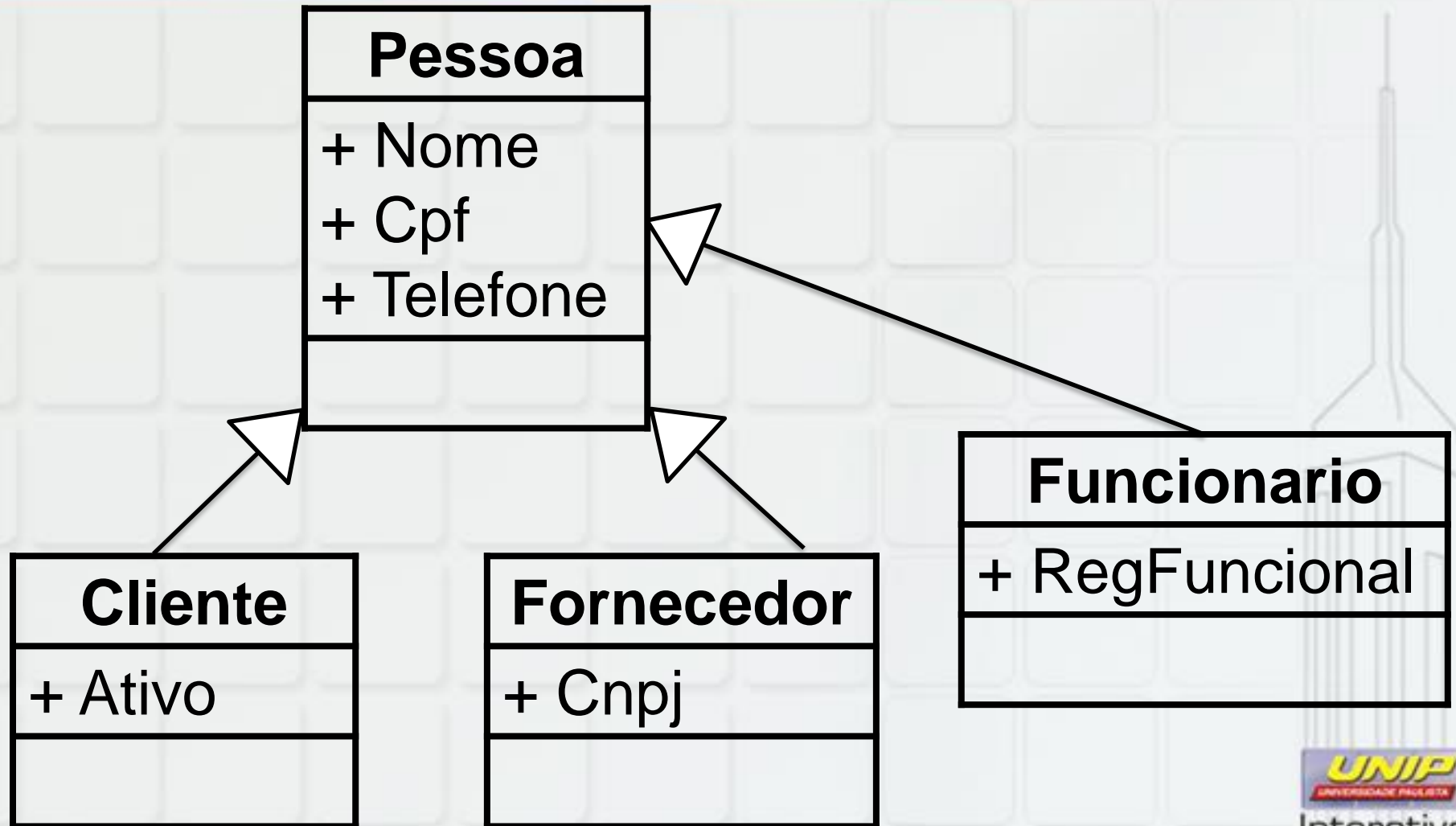
Fornecedor

- + Nome
- + CPF
- + Telefone
- + CNPJ

Funcionário

- + Nome
- + CPF
- + Telefone
- + RegFuncional

Herança



Classe e objetos

Motor
+ Volume
+ Cilindros
+ Acelerar()
+ Ligar()

Classe e objetos

Econo: Motor

- + Volume = 1.0
- + Cilindros = 4
- + Acelerar()
- + Ligar()

Power: Motor

- + Volume = 1.8
- + Cilindros = 4
- + Acelerar()
- + Ligar()

Elite: Motor

- + Volume = 2.4
- + Cilindros = 6
- + Acelerar()
- + Ligar()

Interatividade

Qual das alternativas abaixo não se aplica ao conceito de herança em orientação a objetos?

- a) O uso de herança torna o programa mais eficiente no uso de memória e processamento.**
- b) O uso de herança incentiva a prática do reúso.**
- c) O uso de herança torna o código mais fácil de ser entendido.**
- d) O uso de herança reduz o custo de manutenção do código.**
- e) O uso de herança só é possível em linguagens orientadas a objetos.**

Resposta

Qual das alternativas abaixo não se aplica ao conceito de herança em orientação a objetos?

- a) O uso de herança torna o programa mais eficiente no uso de memória e processamento.
- b) O uso de herança incentiva a prática do reúso.
- c) O uso de herança torna o código mais fácil de ser entendido.
- d) O uso de herança reduz o custo de manutenção do código.
- e) O uso de herança só é possível em linguagens orientadas a objetos.

Polimorfismo

O termo polimorfismo tem origem na composição de duas palavras gregas:

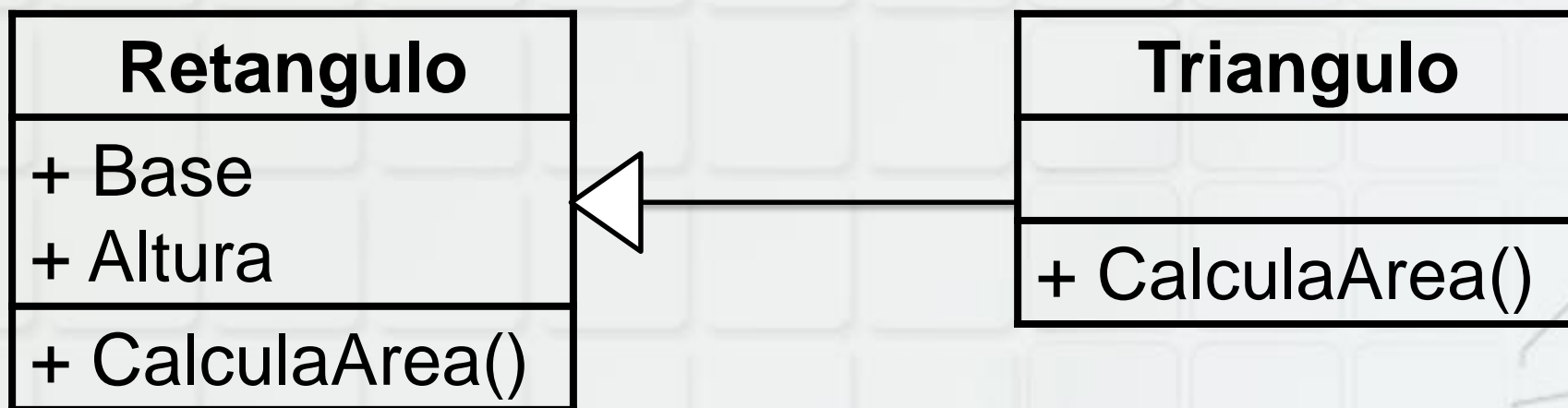
- *Poli* = muitas.
- *Morphos* = formas.



Polimorfismo – sobrecarga

Motor
+ Volume + Cilindros
+ Acelerar (Álcool) + Acelerar (Gasolina) + Acelerar (Álcool, Gasolina) + Ligar ()

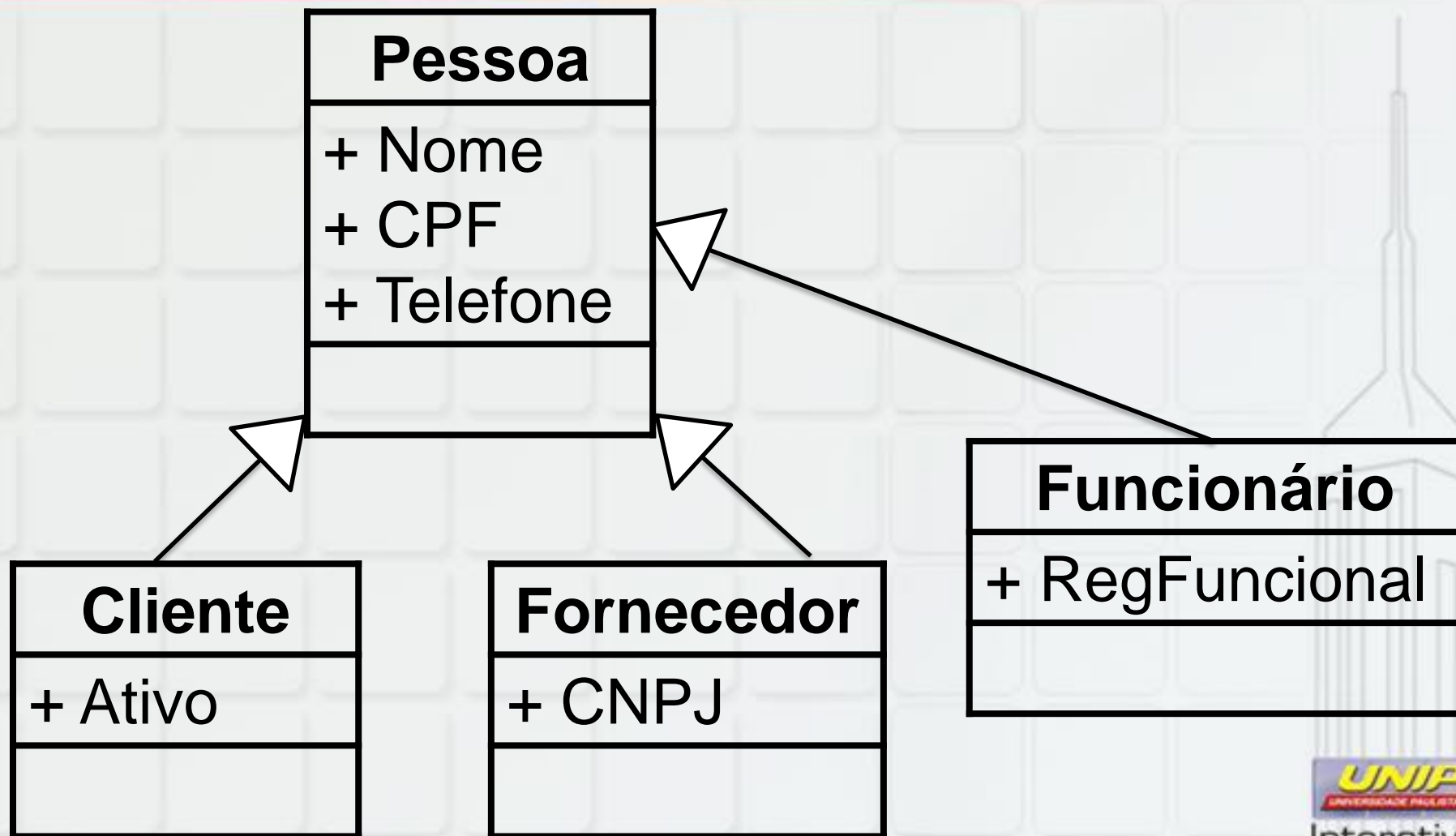
Polimorfismo – sobrescrita



$\text{Área} = \text{Base} * \text{Altura}$

$\text{Área} = (\text{Base} * \text{Altura}) / 2$

Polimorfismo dinâmico ou polimorfismo de classes



Polimorfismo dinâmico ou polimorfismo de classes

Loja
+ EntradaServico (Fornecedor) + EntradaServico (Funcionario) + EntradaPrincipal (Pessoa)

- + EntradaServico (Fornecedor)
- + EntradaServico (Funcionario)
- + EntradaPrincipal (Pessoa)

Interatividade

Qual das seguintes afirmações é falsa?

- a) Objeto é uma instância de uma classe.**
- b) Objetos reúnem dados e comportamentos relacionados com um único conceito.**
- c) Classes não podem se relacionar com outras classes, afinal, cada classe abstrai um único conceito.**
- d) O uso de polimorfismo costuma tornar o código mais fácil de ser entendido.**
- e) O uso de polimorfismo pode reduzir a quantidade de linhas do código.**



Resposta

Qual das seguintes afirmações é falsa?

- a) Objeto é uma instância de uma classe.
- b) Objetos reúnem dados e comportamentos relacionados com um único conceito.
- c) **Classes não podem se relacionar com outras classes, afinal, cada classe abstrai um único conceito.**
- d) O uso de polimorfismo costuma tornar o código mais fácil de ser entendido.
- e) O uso de polimorfismo pode reduzir a quantidade de linhas do código.



ATÉ A PRÓXIMA!

