

Programação Web II

Fundamentos da Programação Orientada a Objetos

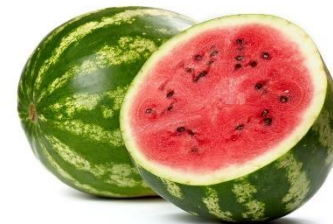


Curso Superior de Tecnologia em Sistemas para Internet

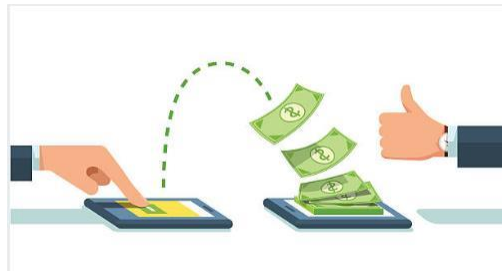
Professor Substituto Me. Anderson José Lauer

O que são objetos?

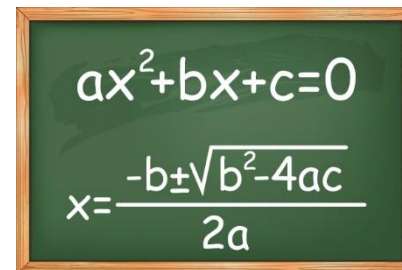
❑ Podemos entender um objeto como algo físico:



❑ No entanto um objeto também pode ser algo que não existe de forma física:



CONTA CORRENTE


$$ax^2+bx+c=0$$
$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

EQUAÇÃO

Diferentes tipos de objetos

Objetos computacionais visuais

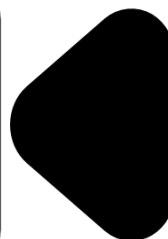
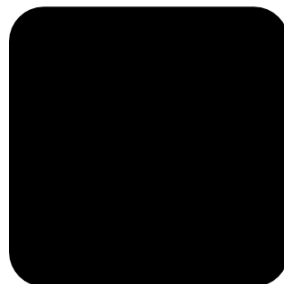
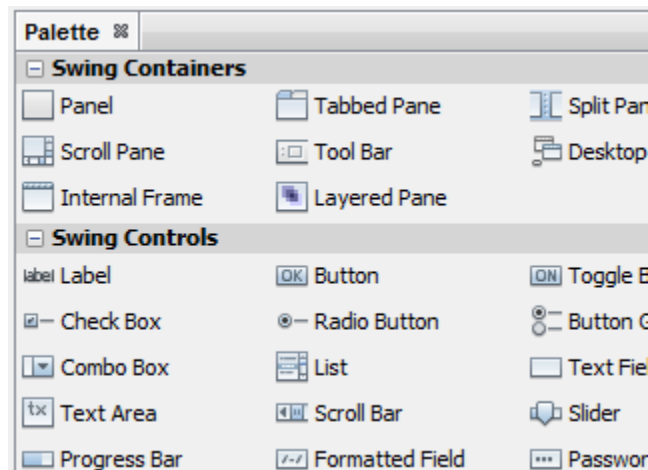
- Menus, caixas de texto, botões, listas...

Objetos com tarefa relacionada

- Objetos que contém uma ação.

Objetos multimídia

- Som, imagem, animação, vídeo...



Análise e Programação OO

☐ Análise orientada a objetos

- ☐ Consiste em definir quais objetos fazem parte de um sistema e a maneira como se comportam.

☐ Programação orientada a objetos

- ☐ Consiste em utilizar objetos computacionais para implementar as funcionalidades de um sistema.

Por que Orientação a Objetos?

- ☐ Aumento de produtividade
- ☐ Ganho de qualidade
- ☐ Ganho de confiabilidade
- ☐ Conhecimento adquirido podendo ser compartilhado
- ☐ Manutenibilidade
- ☐ Custos

Flexibilidade a Mudanças

☐ Mudanças nos processos de negócio do usuário

☐ Década de 40: 10 anos

☐ Década de 60-70: 5 anos

☐ Década de 90: 2 anos

☐ Atualmente: 6 meses ou até menos

Conceitos Básicos da OO

☐ Abstração

☐ Encapsulamento

☐ Objetos

☐ Classes

☐ Persistência

☐ Herança

☐ Polimorfismo

Objetos

- ❑ São elementos reais ou abstratos que sofrem ou executam ações.
- ❑ Um objeto é qualquer indivíduo, lugar, evento, coisa, tela, relatório ou conceito que seja aplicável ao sistema.
- ❑ Objeto é uma entidade capaz de reter um estado (informação) e que oferece uma série de operações (comportamentos) ou para examinar ou para afetar este estado.

Como é um objeto?

- ❑ Apresenta características – ESTADO
- ❑ Executa e sofre ações – COMPORTAMENTO
- ❑ Podem ser classificado por categorias ou CLASSES
- ❑ Interagem e agrupam-se formando sistemas (que podem ser considerados como objetos)

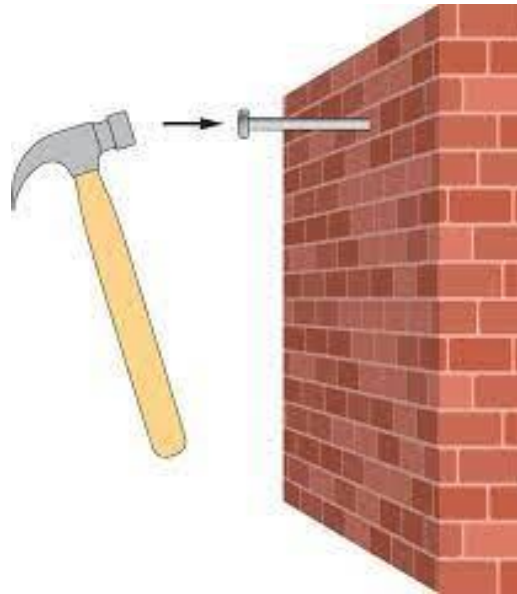
Objetos

❑ Ex: Martelo



MARTELO: Y
CABO: MADEIRA
COR: MARROM
CABEÇA: METAL

ESTADO



COMPORTAMENTO



IDENTIDADE

Objetos

❑ Exemplos:

- ❑ **Coisas:** Cadeira, Mesa, Telefone, Televisão, ...
- ❑ **Funções:** Diretor, Funcionário, Professor, Cliente,...
- ❑ **Eventos:** uma Festa, um Congresso, uma Aula, ...
- ❑ **Lugares:** uma Cidade, uma Sala, um País,
- ❑ **Processos:** uma Operação, um Procedimento, ...

Objetos: atributos

- ❑ É o **conjunto de características** que descrevem os objetos
- ❑ **Atributos de uma pessoa:** cor dos olhos, altura, peso, idade, sexo...
- ❑ **Atributos de uma conta corrente:** correntista, saldo, data de abertura...
- ❑ Somente mudam através de agentes interno ou externos.

Objetos: atributos

❑ Os valores das propriedades de um objeto definem seu Estado.

NOME:	Pedro	João	Maria	Joana
IDADE:	18	35	42	12
ENDEREÇO:	Rua xx	Rua yy	Rua zz	Rua hh
ALTURA:	1,60	1,75	1,73	1,40



Métodos ou Serviços (ações)

- ❑ São as tarefas que o objeto pode realizar
- ❑ As ações do objeto podem interagir com outros objetos
- ❑ Ou podem ser ações internas
- ❑ Ações de um cachorro: latir, babar, correr em círculos, morder...

Exercícios

- ❑ Identificar 5 **OBJETOS**.
- ❑ Identificar os **ATRIBUTOS** de cada objeto.
- ❑ Identificar as **AÇÕES** de cada objeto.

Abstração

❑ Ato de separar um ou mais elementos de uma totalidade complexa (coisa, representação, fato), os quais só mentalmente podem ser separados.

❑ Ignorar aspectos não relevantes

❑ É a capacidade de focalizar o essencial e ignorar detalhes acidentais não relacionados com o objetivo estabelecido.

❑ Ex: Folha de Pagamento



Abstração

❑ Ex: Folha de Pagamento



❑ Ex: Motorista e carro



Encapsulamento

- ❑ Ocultar aspectos não relevantes.
- ❑ Agregação de componentes formando um novo todo.
- ❑ Permite alterações sem afetar as demais partes de um sistema.
- ❑ Benefícios:
 - ❑ **Modularidade:** O código é independente de outros objetos.
 - ❑ **Informações privadas:** O objeto tem uma interface pública que outros objetos podem utilizar.
 - ❑ As informações dos objetos são privadas.

Encapsulamento

❑ Ex: Carro



❑ O motorista tem acesso ao carro, mas o que é referente ao funcionamento pertencem ao carro.

Polimorfismo

- ❑ As classes fazem a mesma operação (método), mas de forma diferente
- ❑ Ocorre quando um método definido no ancestral é redefinido no descendente com um comportamento diferente
- ❑ Todo método definido em um ancestral pode ser acessado através dos descendentes

Polimorfismo

GATO



CACHORRO



FALAR()

MIAU! MIAU! MIAU!

AU! AU! AU!

Classes

- ❑ São os modelos a partir dos quais os objetos são criados (Ex: Planta Estrutural)
- ❑ Os modelos são agrupados em classes ou categorias por similaridade de finalidade
- ❑ As categorias estruturam-se hierarquicamente
- ❑ Ex:
 - ❑ Pessoas–Clientes-Funcionários
 - ❑ Animais-Aves-Canários

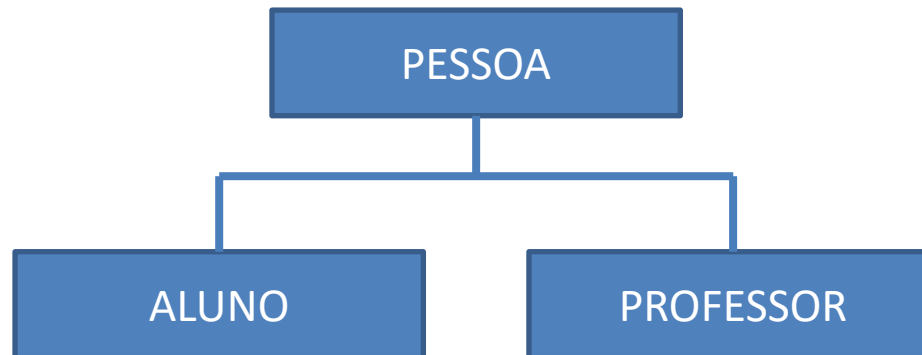
Classes

- ❑ Quais as categorias ou classes que podem ser identificadas?



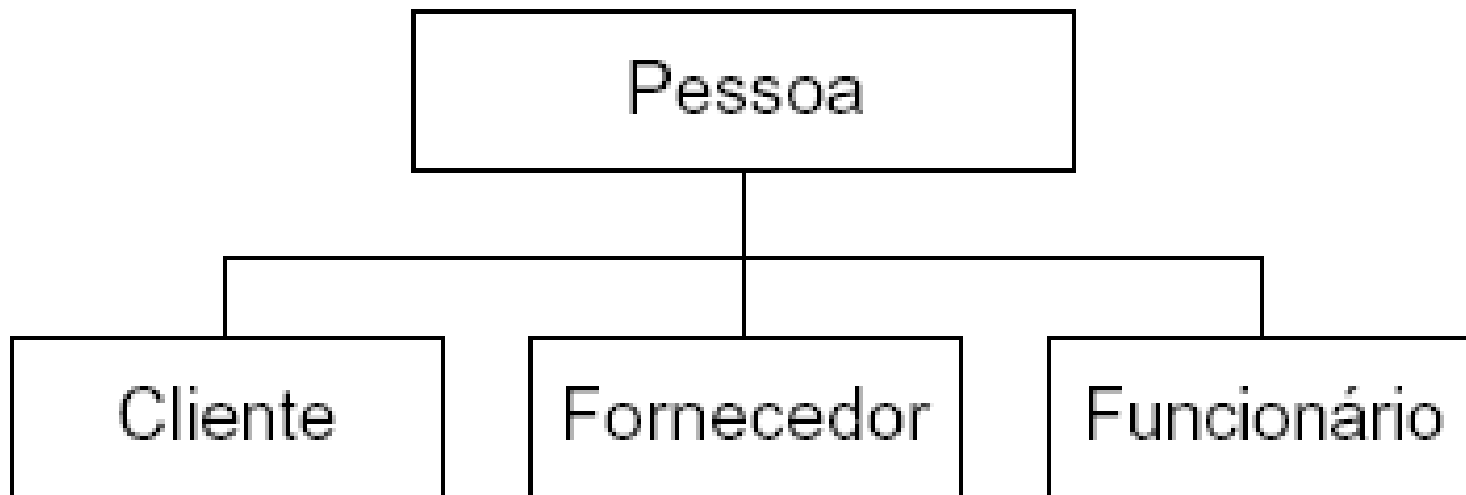
Hierarquia de Classes

- ❑ Características comuns a todos.



Classes Puras ou Abstratas

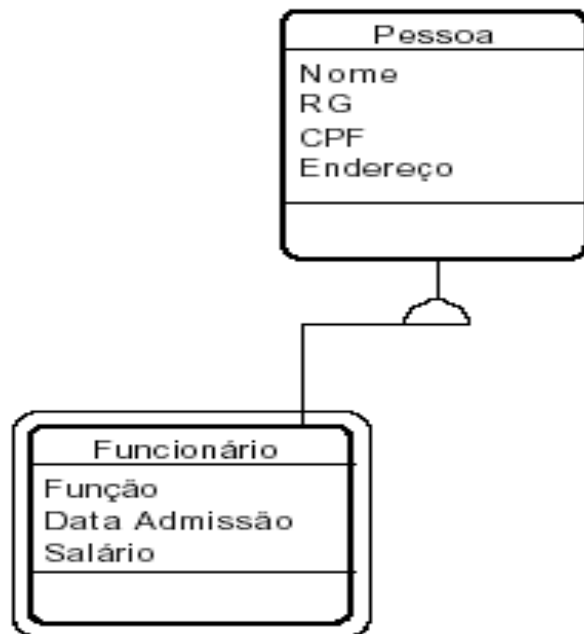
- ❑ São aquelas que nunca serão instanciadas diretamente.
- ❑ Servem de apoio para o processo de estruturação.



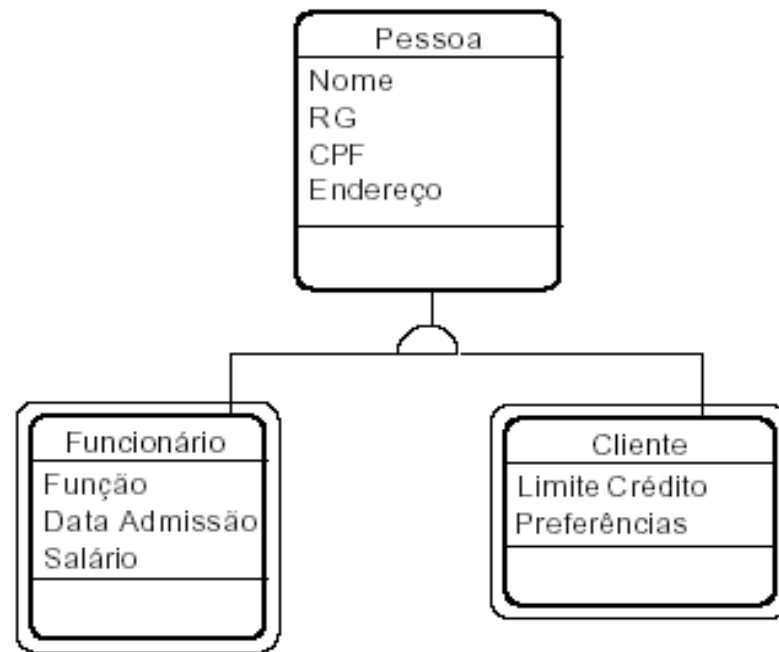
Herança

- ❑ Tudo aquilo que se herda
- ❑ Aquilo que se transmite por hereditariedade (Ex: Características genéticas dos pais)
- ❑ Classes prontas podem ser usadas para escrever novos objetos (SuperClasse e SubClasses, Ancestrais e Descendentes)
- ❑ O programador só escreve o que muda
- ❑ Alterações nos ancestrais se propagam aos descendentes
- ❑ Menos bugs, mais produtividade

Herança Simples

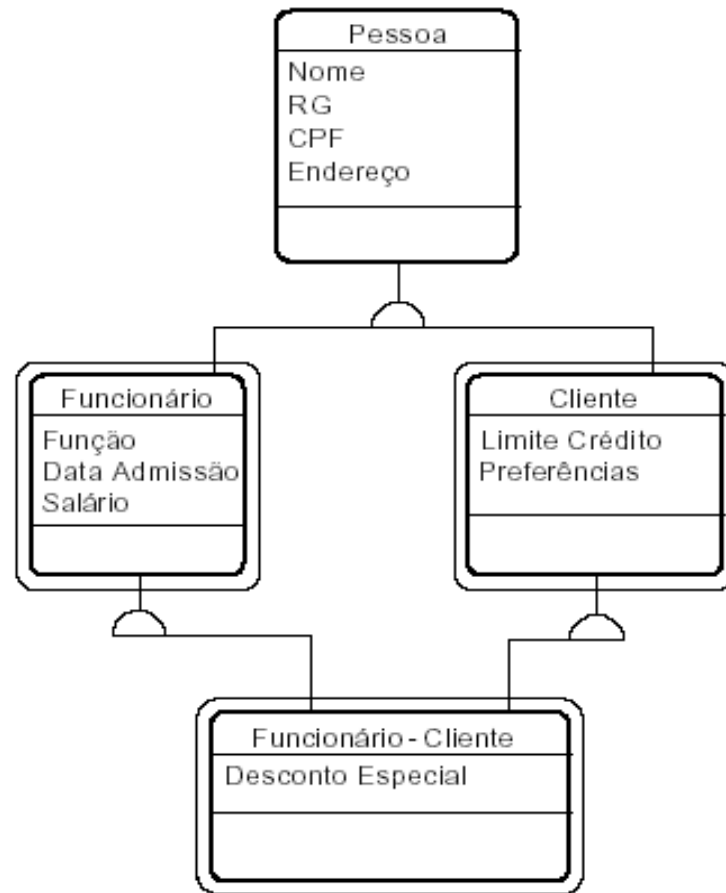


Herança simples



Subclasses da mesma superclasse

Herança Múltipla



Instanciação

- ❑ É quando a classe produz um objeto, como se ela fosse um modelo para a criação de objetos.
- ❑ Objetos nada mais são do que classes instanciadas.
- ❑ É o que dá existência ao objeto.
- ❑ Cada objeto é uma instância de sua classe

Exemplos de Instâncias

