

Programação Estruturada e Orientada a Objetos

Introdução à Orientação a Objetos



PROF. PAULO HENRIQUE MAIA

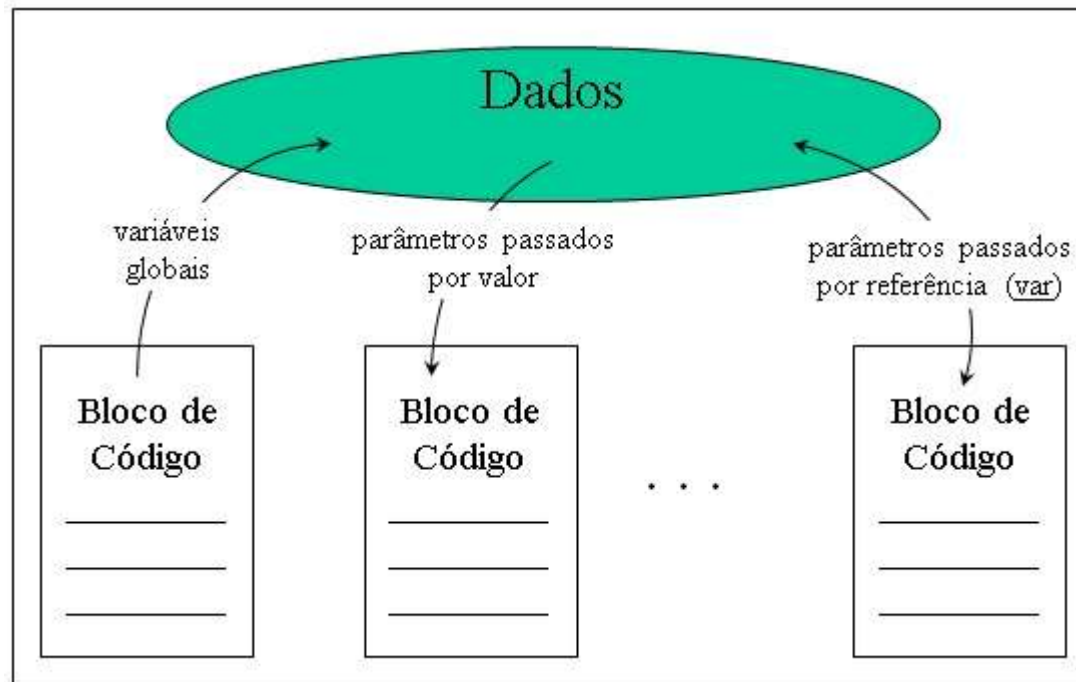
Histórico

- *Um paradigma é uma forma de abordar um problema.*
- O paradigma da orientação a objetos surgiu no fim dos anos 60.
- Hoje em dia, praticamente suplantou o paradigma anterior, o *paradigma estruturado...*

O paradigma estruturado

- Um programa é composto por dados centralizados e diversos módulos
- Cada módulo é composto por diversas funções
- Cada função é responsável por parte da solução do problema
- Fraco acoplamento entre dados e funções
- O que leva um programador a mudar do paradigma procedimental para um novo?
 - Complexidade crescente dos sistemas
 - Limitações da capacidade humana de compreensão de um sistema com um todo.

O paradigma estruturado



O paradigma da Orientação a Objetos

- Um paradigma de programação que permite representar programaticamente elementos do mundo real e suas interações
- Características marcantes:
 - Alta reutilização de software
 - Desenvolvimento sempre evolutivo
 - Construções complexas a partir de construções simples
 - Manutenção tende a ser simples
- Promete, quando bem utilizada, oferecer ganhos em termos de rapidez, custo, confiabilidade, flexibilidade e facilidade de manutenção

O paradigma da Orientação a Objetos

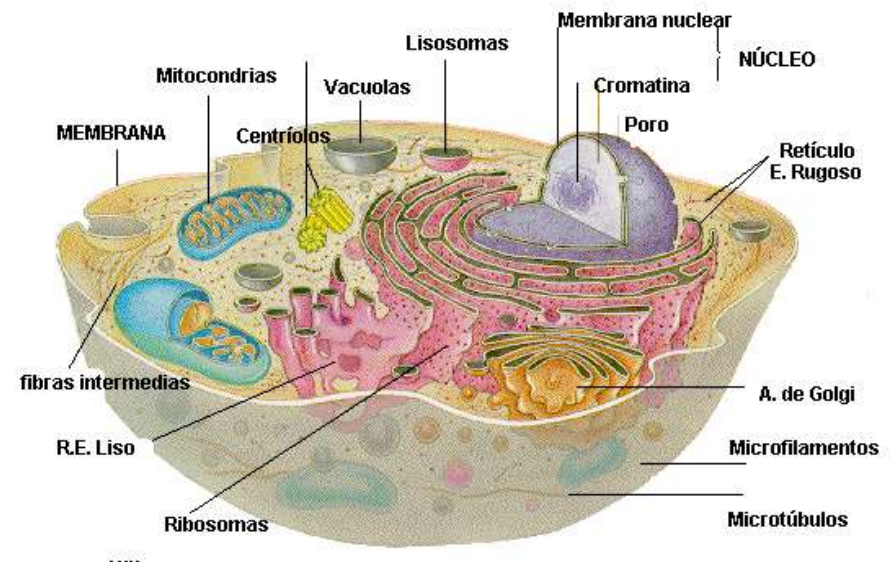
■ Alan Kay, um dos pais do paradigma da orientação a objetos, formulou a chamada **analogia biológica**.

■ “Como seria um sistema de software que funcionasse como um ser vivo?”



Analogia Biológica

- Cada “célula” interagiria com outras células através do envio de mensagens para realizar um objetivo comum.
- Adicionalmente, cada célula se comportaria como uma unidade autônoma.



Analogia Biológica

- De uma forma mais geral, Kay pensou em como construir um sistema de software a partir de agentes autônomos que interagem entre si.
- Com isso, ele estabeleceu os princípios da **orientação a objetos**.

Orientação a Objetos - Princípios

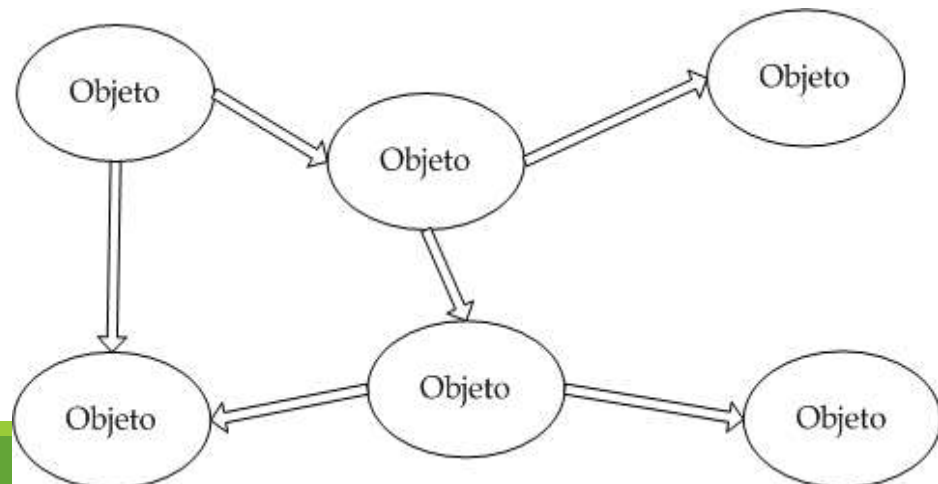
■ Tudo é um objeto

Pense em um objeto como uma super variável: ele armazena dados, mas você também pode fazer requisições a esse objeto, pedindo que ele faça operações sobre si próprio. Em teoria, você pode representar qualquer elemento conceitual no problema que você está tentando resolver (cachorros, livros, sócios, empréstimos, etc.) como um objeto no seu programa.

Orientação a Objetos - Princípios

■ **Um programa é uma coleção de objetos dizendo uns aos outros o que fazer**

Para fazer uma requisição a um objeto você “manda uma mensagem” para este objeto. Mais concretamente, você pode pensar em uma mensagem como sendo uma chamada de um procedimento ou função pertencente a um objeto em particular.



Orientação a Objetos - Princípios

■ Um objeto pode ser composto por vários outros objetos

Em outras palavras: você pode criar um novo tipo de objeto empacotando objetos existentes. Dessa forma, você pode adicionar complexidade a um programa e escondê-la por trás da simplicidade de uso dos objetos.

Orientação a Objetos - Princípios

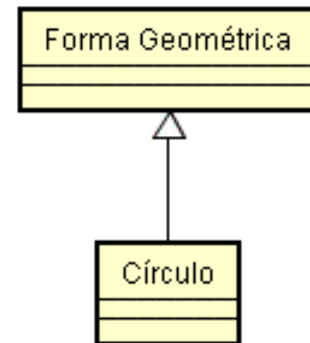
■ **Todo objeto tem um tipo**

Usando as palavras certas, cada objeto é uma instância de uma classe, onde classe é um sinônimo de tipo. A questão mais importante relativa a uma classe é “que mensagens eu posso enviar para uma instância dessa classe?”

Orientação a Objetos - Princípios

■ Todos os objetos de um dado tipo podem receber as mesmas mensagens

Além disso, uma vez que, por exemplo, um objeto do tipo “círculo” é também um objeto do tipo “forma geométrica”, o objeto “círculo” aceita qualquer mensagem endereçada a uma “forma geométrica”. Essa capacidade de “substituição” de um objeto por outro é um dos mais poderosos conceitos em orientação a objetos.



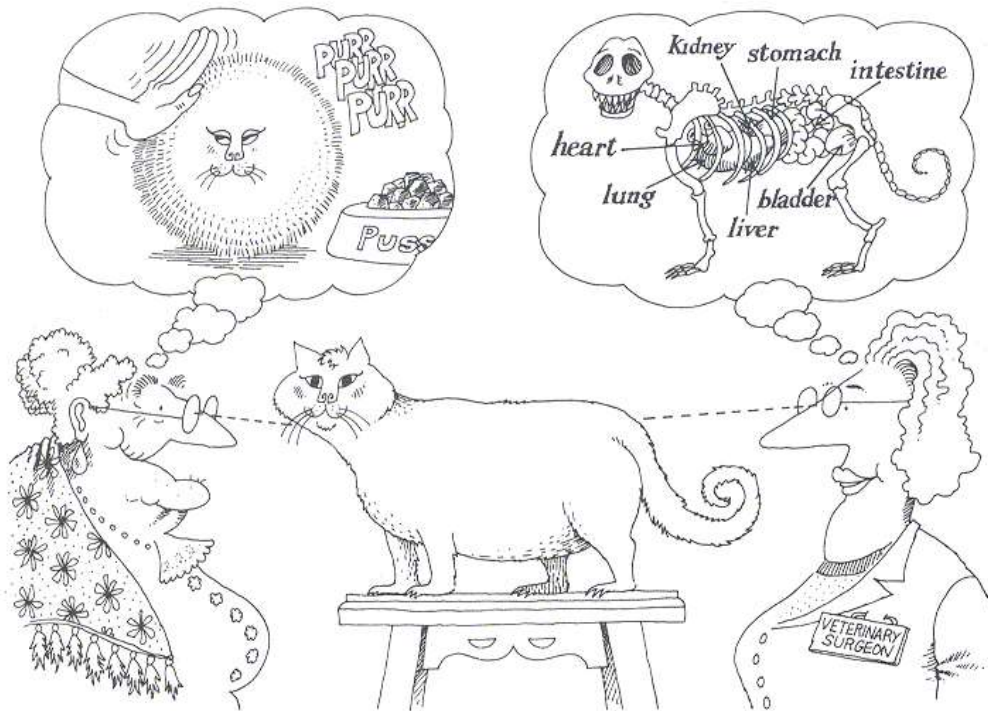
Orientação a Objetos

■ Principais características

- Abstração
- Objetos
- Classes
- Encapsulamento
- Herança
- Polimorfismo

Abstração

- Uma abstração é qualquer modelo que inclui os aspectos relevantes de alguma coisa, ao mesmo tempo em que ignora os menos importantes.



Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

Objeto

- Definição:
 - Um conceito, uma abstração com significado específico em um contexto
 - Entidade de software que reflete de alguma forma o mundo real
- Propósito:
 - Representar uma entidade do mundo real
- Objetos possuem:
 - **Identidade**
 - Conjunto de **características** que determinam seu estado
 - **Comportamentos** específicos definidos por um conjunto de ações

Exemplos

Beija-Flor



Identidade: ‘o beija-flor que vem ao meu jardim’

Características:

penas azuis

bico fino

vôo rápido

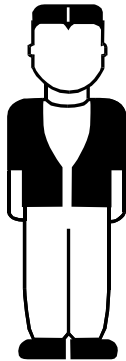
Comportamento:

voar

piar

Exemplo

■ Pessoa



Identidade: ‘Mário’

Características:

olhos pretos

nasceu em 16/02/70

pesa 70kg

mede 1,70m

Comportamento:

andar

falar

comer

rir

Exemplo

Telefone



Identidade: número 2576-0989

Características:

azul

2.4 GHz

tone

Comportamento:

tocar

discar

Exemplo

Ônibus



Identidade: placa LXY 7684

Características:

cor amarela

30 assentos

a diesel

Comportamento:

frear

andar

correr

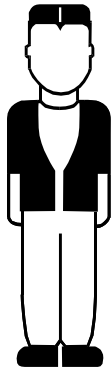
buzinar

acelerar

Objeto

Representação gráfica

Características
(estado)



Mário

Funcionário
nome = Mário Sá
nasc = 16/02/1970
salário = R\$ 3.000,00
informarSalário()
calcularIdade()

Identidade

Comportamento

Serviços,
Interface ou
protocolo

Exercício

Identifique as características e comportamentos das seguintes entidades e faça sua representação gráfica

1. Uno é um carro da Fiat que possui 4 portas, cor branca, pneus aro 13 e câmbio manual. Veloster é um carro da Hyundai com 3 portas, cor prata, pneus aro 14 e câmbio automático. Ambos permitem seu condutor acelerar, frear e passar marchas.

Exercício

Identifique as características e comportamentos das seguintes entidades e faça sua representação gráfica

2. André e Erica são alunos da UECE. Ele é do curso de Medicina e nasceu em 1990, ela cursa Psicologia e nasceu em 1992. O número de matrícula dele é 1234 e dela 5678. Ambos podem se matricular em uma disciplina e imprimir seus históricos.

Classe

■ Definição:

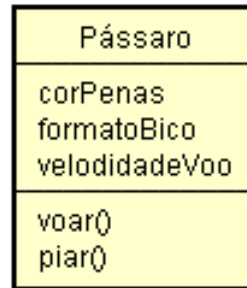
- Abstrações utilizadas para representar um conjunto de objetos com *características e comportamento idênticos*

■ Uma classe pode ser vista como uma “fábrica de objetos”

■ Tecnicamente falando, objetos são “instâncias” em tempo de execução de uma classe

- Todos os objetos são instâncias de alguma classe
- Todos os objetos de uma classe são idênticos no que diz respeito a sua interface e implementação (*o que difere um objeto de outro é seu estado e sua identidade*)

Exemplo



classe



Identidade: o beija-flor que vem ao meu jardim

Características:

cor das penas: azuis

formato do bico: fino

velocidade de vôo: rápida

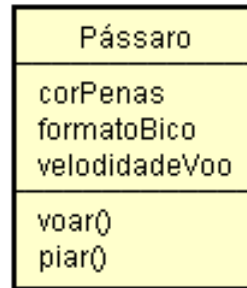
Comportamento:

voar

piar

instância da
classe (objeto)

Exemplo



classe

Identidade: meu pombo correio

Características:

cor das penas: cinza

formato do bico: curto

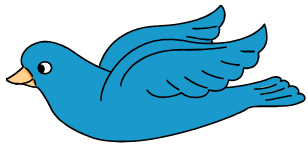
velocidade de vôo: média

Comportamento:

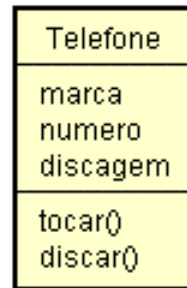
voar

piar

instância da
classe (objeto)



Exemplo



classe



Identidade: ‘Telefone da minha casa’

Características:

marca: Siemens
número: 2576-0989
discagem: pulso

Comportamento:

tocar
discar

instância da
classe (objeto)

Exemplo

Telefone
marca numero discagem
tocar() discar()

classe

Identidade: ‘Meu celular’



Características:

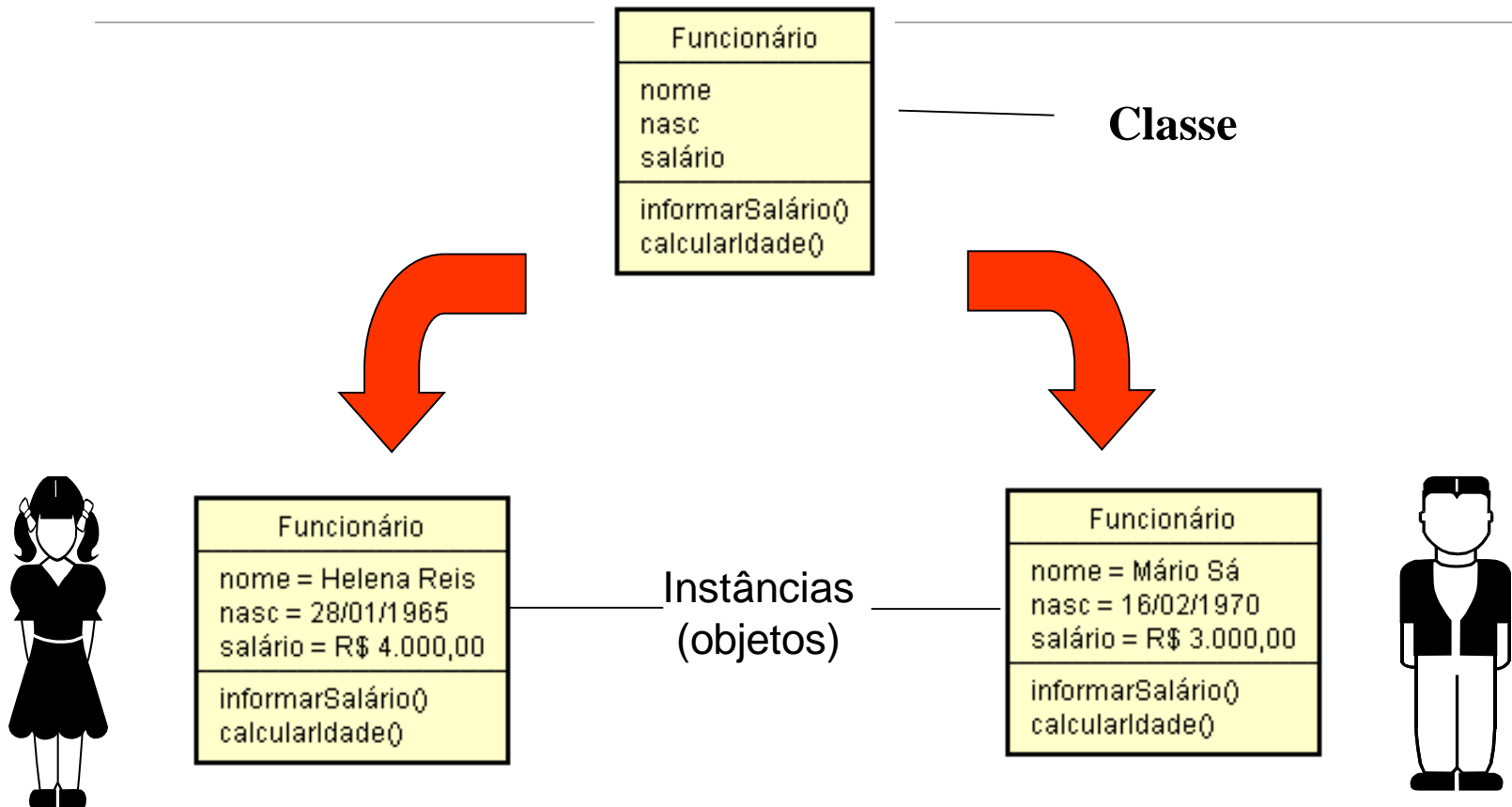
marca: Nokia
número: 99193467
discagem: tom

instância da
classe (objeto)

Comportamento:

tocar
discar

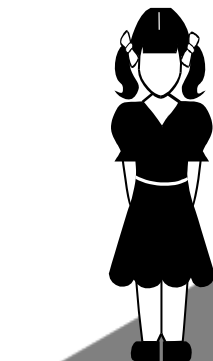
Classes



Atributos

- Descrevem as **características** das instâncias de uma classe
- Seus valores definem o **estado** do objeto
- O estado de um objeto pode mudar ao longo de sua existência
- A identidade de um objeto, contudo

Funcionário
nome
nasc
salário
informarSalário()
calcularIdade()



Funcionário_Helena
Nome= Helena Reis
Nasc= 28/01/1965
Salário = 4.000
InformarSalário
CalcularIdade



Funcionário_Mário
Nome= Mário Sá
Nasc= 16/02/1970
Salário = 3.000
InformarSalário
CalcularIdade

Serviços/Operações

- Representam o **comportamento** das instâncias de uma classe
- Correspondem ao **protocolo** ou **ações** das instâncias de uma classe



Classes em Java

```
public class NomeDaClasse {  
    // lista de atributos  
    // lista de métodos  
}
```


Classes em Java

```
public class NomeDaClasse {  
    // lista de atributos  
    <tipo> nomeAtributo;  
    // lista de métodos  
    <tipo retorno> nomeMetodo (<tipo> param1, ...,  
                                <tipo>paramN) {  
        // implementação  
    }  
}
```

Classes em Java

```
public class Aluno {  
    String nome;  
    String curso;  
    int matricula;  
    int ano;  
  
    void mostrarDados(){  
        System.out.println("Nome = " + nome);  
        System.out.println("Curso = " + curso);  
        System.out.println("MATricula = " + matricula);  
    }  
  
    int calcularIdade(int anoAtual){  
        int idade = anoAtual - ano;  
        return idade;  
    }  
}
```

Nome de classe começando com letra maiúscula

Nome de atributo, método e parâmetro começando com letra minúscula

Nomes compostos começando com letra minúscula concatenado com letra maiúscula

Classes em Java

```
public class Aluno {
```

```
String nome;  
String curso;  
int matricula;  
int ano;
```

```
void mostrarDados(){
```

```
    System.out.println("Nome = " + nome);  
    System.out.println("Curso = " + curso);  
    System.out.println("MATricula = " + matricula);  
}
```

```
int calcularIdade(int anoAtual){
```

```
    int idade = anoAtual - ano;  
    return idade;  
}
```

```
}
```

Identação



Criando objetos

```
public class Principal {  
    public static void main(String[] args) {  
        Aluno a1 = new Aluno();  
        a1.mostrarDados();  
        int idade = a1.calcularIdade(2014);  
        System.out.println("Idade=" + idade);  
    }  
}
```

Aluno() é o método construtor

Execução de método

Qual a saída desse programa?

Criando objetos

```
public class Principal {  
  
    public static void main(String[] args) {  
        Aluno a1 = new Aluno();  
        a1.nome = "André";  
        a1.curso = "Medicina";  
        a1.matricula = 1234;  
        a1.ano = 1990;  
        a1.mostrarDados();  
        int idade = a1.calcularIdade(2014);  
        System.out.println("Idade=" + idade);  
    }  
}
```

Acesso aos atributos



Qual a saída desse programa?

Como criar um objeto para a aluna Erica?

Exercício

- 1) Crie uma classe que represente uma conta bancária. Toda conta possui um titular, um número e um saldo. Crie um método para depositar um valor e outro para realizar um saque (lembre que saldo não pode ser negativo)
- 2) Crie uma classe principal que instancie duas contas, configure seus atributos e realize saques e depósitos