








# Programação Desktop

*Fabício Curvello Gomes*





# Orientação a Objetos

INFORMA. FORMA. TRANSFORMA.

# Orientação a Objetos

Na compreensão do mundo, os seres humanos utilizam-se de três métodos de organização dos pensamentos:

- 1 – Diferenciação
- 2 – Distinção entre todo e parte
- 3 – Classificação

A orientação a objetos classifica o mundo real em objetos, descritos na programação orientada a objetos.

3

INFORMA. FORMA. TRANSFORMA.

# Orientação a Objetos (Cont.)

Ex:  
*Mundo Real*

Casa Branca

Marcelo

Corolla

Mangueira

---

*Modelo*

Casa Branca


Marcelo

Corolla

Mangueira

← Mora na
Dirige →

4





# Objetos

Apesar de existirem diferentes tipos de objetos, todos possuem duas características principais:


- **Estado** (Propriedades)  
Ex: Cor, Modelo, Vel Máx, Ano
- **Comportamento** (Ações possíveis)  
Ex: Mostrar Velocidade, Acelerar, Frear

Chamaremos cada estado de **atributo**, e cada comportamento de **método**.






5




# Classes





Uma classe “molda” um grupo de objetos com propriedades semelhantes (atributos), mesmos comportamentos (operações), mesmos relacionamentos com outros objetos (associações e agregações), e mesma semântica.

Um objeto é uma instância de uma classe.

Os objetos são criados a partir de classes. Os objetos são instâncias de classe que permanecem em memória e mantêm seus valores de maneira individual.

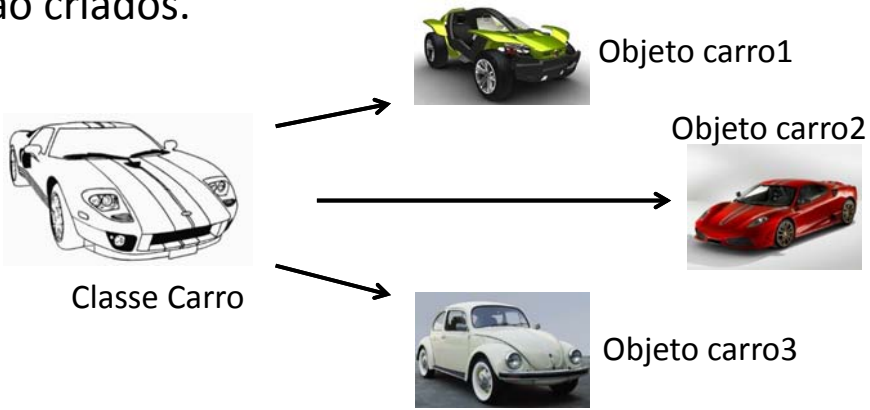



6





INFORMA. FORMA. TRANSFORMA.

# Classes (Cont.)

A classe é um “molde”, a partir do qual os objetos são criados.







7



INFORMA. FORMA. TRANSFORMA.

# Exemplos de Classes

- Professor
  - *Atributos*: Nome, Matrícula, Data de Contratação, Titulação
  - *Operações*: DefineNome(), AlteraNome(), TempoServiço(), DefineTitulação(), AlteraTitulação(), ...
- Turma
  - *Atributos*: Cod, Nome, Local, Créditos, Horário, Capacidade.
  - *Operações*: DefineCod(), AlteraCod(), DefineNome(), AlteraNome(), NumCreditos(), AdicionaAluno(), EliminaAluno(), VerificaEstado(),...


8



INFORMA, FORMA, TRANSFORMA.

# Encontrando Classes


Uma classe deveria capturar uma e somente uma abstração chave.



## Abstração Ruim

Classe "Aluno" que conhece a informação do aluno e as disciplinas que o aluno está matriculado.

## Abstração Boa


Separar uma classe para Aluno e uma classe para Disciplina.


9



INFORMA, FORMA, TRANSFORMA.


# Pacotes


Pacote é um diretório no sistema de arquivos que contém classes que são logicamente relacionadas.



Para indicar que uma classe pertence a determinado pacote, ela deve ser colocada na pasta adequada e o comando ***package*** deve indicar o nome desta pasta. Este comando é único e deve ser o primeiro do arquivo:


***package*** *nome do package* ;



10



# Padrão MVC

- **Pacote model**  
Armazena Classes de Negócio (Classes responsáveis em dar forma e funcionalidade a futuros objetos).
- **Pacote view**  
Armazena Classes que interagem com usuário (Telas).
- **Pacote controller**  
Armazena classes que controlam o sistema fazendo a interação entre o view e o model.

11




# Atributos

O estado de um objeto é dado por valores de atributos e por ligações que tem com outros objetos.

Todos os objetos de uma classe são caracterizados pelos mesmos atributos, ou variáveis de instâncias.

O mesmo atributo pode ter valores diferentes de objeto para objeto.


Atributos são definidos ao nível da classe, enquanto que os valores dos atributos são definidos ao nível do objeto.

12





INFORMA, FORMA, TRANSFORMA.

## Exemplos de Atributos e Valores

Classe Carro	objeto carro1	objeto carro2	objeto carro3
			
Atributos:	Valores:	Valores:	Valores:
-Fabricante	-Fiat	-Ferrari	-Volkswagen
-Modelo	-Bugster	-F430	-Fusca
-Cor	-Verde	-Vermelho	-Branco
-Ano	-2011	-2009	-1986



13



INFORMA, FORMA, TRANSFORMA.

## Operador new

Para trabalhar com um objeto, é necessário criá-lo.  
Esta criação envolve dois passos:

1. Criação de uma variável.
2. Criação do objeto utilizando o operador new.


14

## Exercício *03\_ExemploNew*

```
package model;
public class Usuario {
    public String nome;
    public String endereco;
}
```

Pacote model

Pacote controller

```
package controller;
import model.Usuario;
public class Cadastro {
    public static void main(String[] args) {
        Usuario user = new Usuario();
        user.nome="João da Silva Sauro";
        user.endereco="Rua Sem Fim, 72";
    }
}
```



15

## Convenções



### Métodos e variáveis:

- Palavra simples - Tudo em minúsculo
- Palavra composta - Começa com minúscula e a inicial das outras palavras em maiúscula.


### Constantes:

- Tudo em maiúsculo.
- Palavras compostas separadas com \_ (sublinhado).



16







## Convenções (Cont.)

**Classes:**


- Palavra simples - Começa com maiúscula.
- Palavra composta - Começa com maiúscula e a inicial das outras palavras em maiúscula.

**Arquivos:**

- Mesmas convenções das classes.




17





## Variáveis de instância

Variáveis de instância são aquelas declaradas dentro da definição da classe.

Cada objeto criado a partir da classe possui sua cópia de cada variável de instância.



18




INFORMA, FORMA, TRANSFORMA.



## Variáveis de instância (Ex:)

```

public class Empregado {
    private int idade;
    //tipos primitivos numéricos => Valor default = 0
    private boolean permiteEscrita;
    //tipos primitivos booleanos => Valor default = false
    private String nome;
    //objetos => Valor default = null
    ...
}

```


19




INFORMA, FORMA, TRANSFORMA.

## Exercício *04\_ClassesEObjetos*

Desenvolva a classe abaixo e crie pelo menos 3 objetos a partir dela.


Carro
+ fabricante
+ modelo
+ cor
+ ano

Criar pacote **model** para incluir esta Classe, e pacote **controller** com a Classe **Cadastro** para inserir os objetos.


20

Sistema

FIRJAN




INFORMA. FORMA. TRANSFORMA.

# Mensagens

Uma mensagem é uma solicitação feita por um objeto A à um objeto B.

Como resultado desta solicitação, o objeto B irá modificar seu estado ou irá retornar algum valor.


O conceito de mensagem está diretamente associado ao conceito de operação.



21

Sistema


FIRJAN





INFORMA. FORMA. TRANSFORMA.

# Mensagens (Cont.)

A interação entre os objetos é feita através da troca de mensagens.


  
Raphael

  
Compra

  
Parati

Status Inicial:  
À Venda

Status Final:  
Vendido



22

**Exercício de Fixação**

No projeto **04\_ClassesEObjetos**, crie as classes abaixo e pelo menos 3 objetos para cada uma destas classes:

23



**Projeto InfoNote\_01**

1 - Criar todas as classes conforme este diagrama  
 2 - Criar no pacote *controller* a classe *Teste* com:

- 2.1 - Um objeto *usuário*
- 2.2 - Um objeto *endereço*
- 2.3 - Três objetos *notebook*
- 2.4 - Um objeto *pedido*
- 2.5 - Dois objetos *item de pedido*


3 - Na classe *Teste* exiba os objetos criados.

24





INFORMA. FORMA. TRANSFORMA.

# Dúvidas?




25




INFORMA. FORMA. TRANSFORMA.


# Bibliografia



Java Como Programar 8ª Edição  
Paul Deitel e Harvey Deitel  
Ed. Pearson



Java 7 Ensino Didático  
Sérgio Furgeri  
Ed. Érica



Fundamentos de Computação e Orientação a Objetos Usando Java  
Francisco A. C. Pinheiro  
Ed. LTC

26