

Aula 3 – POO 1

Classe e Objeto

Profa. Elaine Faria
UFU - 2021

Sobre o Material

- Agradecimentos
 - Aos professores José Gustavo e Fabiano, por gentilmente terem cedido seus materiais.
- Os slides consistem de adaptações e modificações dos slides dos professores José Gustavo e Fabiano

Introdução

- Ao escrever um programa OO, cria-se um modelo do mundo real
 - Modelo descrito por partes, que são os **objetos** que aparecem no domínio do problema
 - Os **objetos** possuem características próprias, denotadas por atributos.
 - Os **objetos** possuem funções (comportamentos específicos), dados por **métodos**
 - Os **objetos** podem ser categorizados, agrupados, e uma **classe** descreve todos os objetos de um tipo particular.

Objetos

- Elementos componentes de um problema
- Coisas do mundo real ou imaginário, que podemos de alguma forma identificar
 - Uma pedra, uma caneta, um copo, etc.
- Não necessita ser real, ou existir no mundo real
- Não necessita ser palpável
- Representam **entidades em um sistema de software** que representam instâncias de entidades do mundo real e de algum sistema

Objeto

- Possui propriedades que o caracterizam, e que são armazenadas no próprio objeto: **atributos**
 - Quando um objeto comunica-se com o meio, realiza ações que podem alterar seu estado: alterações nos valores de seus atributos
- Realiza operações para contribuir com o funcionamento do sistema: representadas por **métodos**
 - Serviços oferecidos pelo objeto
 - Comunicação com outros elementos (objetos) do sistema

Objeto - Características

- Identidade
 - Identificação unívoca de um objeto
 - Distinção por sua própria existência
 - Dois objetos são distintos mesmo que todos os seus atributos sejam iguais
 - Identificador único para cada objeto
- Persistência
 - Tempo de vida de um objeto: temporário ou permanente
 - Temporário: existe durante a execução do programa
 - Permanente: armazenado em um meio físico

Comunicação entre Objetos

- Mensagens
 - Elemento utilizado para prover a comunicação entre objetos
 - Definem
 - Nome do serviço requisitado
 - Informação necessária para a execução do serviço
 - Nome do requisitante
 - Na pratica, mensagens são implementadas como ativações de um **método** definido no objeto chamado, no qual:
 - Nome: identifica o método
 - Informação necessária: lista de parâmetros
 - Requisitante: objeto que realizou a chamada

Classe

- Objetos de um mesmo tipo possuem os mesmos comportamentos
- Dizemos que objetos de mesmo tipo pertencem à mesma classe
 - Uma classe é um tipo a partir do qual objetos são criados

Classe

- A classe define:
 - Os elementos de dados que um objeto contém (atributos)
 - O comportamento que o objeto possui (métodos)
 - A maneira como esses elementos de dados e métodos podem ser acessados (interface)

Classe

- Representa uma abstração de objeto
 - Tipo Abstrato de Dado: tipo definido pelo usuário
 - Contém toda a descrição da forma do objeto
 - Molde para a criação do objeto
 - Matriz geradora de objetos
 - Fábrica de objetos

Paradigma orientado a objetos

- Em POO
 - Um conceito do mundo real é um objeto
 - Escreve-se código organizado em torno de objetos (conceitos), não de funções
 - POO → modelagem conceitual
 - Procedimental → modelagem funcional (e.g. DFD)

Paradigma orientado a objetos

Qual é a diferença entre Classe e Objeto?

a) Classe:

- é definição do tipo;
- representa um conjunto de objetos de mesmo tipo;
- Classe = {obj1, obj2, obj3, ... , objN}

b) Objeto:

- é cada instância derivada da classe;
- é um elemento do conjunto representado pela classe

Paradigma orientado a objetos



Atributos

- Definições: classes
- Valores associados: objetos
- **Atributos de classe (coletivos)**
 - Mesmo valor **inicial** é dividido entre todos os objetos criados (valor padrão)
 - Valor armazenado na classe
 - Possuem o mesmo valor para todos os objetos de uma classe
 - Área de memória compartilhada por todos os objetos da classe

Atributos

- Definições: classes
- Valores associados: objetos
- **Atributos de objeto (individuais)**
 - Atributo individual armazenado no objeto
 - Cada objeto define um valor inicial, que pode coincidir com o valor de outros objetos
 - Área de memória reservada individualmente para cada objeto

Métodos

- Comportamentos, ações e reações dos objetos
- Uma função, um serviço fornecido pelo objeto para outros objetos
- Podem manipular os atributos do objeto (alteração do estado do objeto)
- **Exemplos**
 - Um automóvel tem o comportamento de se locomover
 - Um computador processa programas
 - Uma método de conversão de uma imagem colorida em tons de cinza altera o estado da imagem, convertendo cada pixel colorido em um pixel cinza

Modelagem Conceitual

- Análise dos objetos do mundo real e suas inter-relações
 - Descobrir os conceitos (classes) que compõe o domínio do problema.
 - Elementos básicos: Conceitos (classes), Atributos, Associações.
- Conceitos
 - São substantivos que representam “coisas” que o sistema manipula
 - São fortes candidatos a classes
- Diagrama de classes da UML
 - Ferramenta para modelagem conceitual e projeto da arquitetura do software

Modelagem Conceitual

- Representação UML para uma classe

NomeClasse
visibilidade nomeAtributo : tipo = valor default visibilidade nomeAtributo : tipo = valor default
visibilidade nomeMétodo(listaArgumentos) : tipoRetorno ... visibilidade nomeMétodo(listaArgumentos) : tipoRetorno

Modelagem Conceitual

Carro
+Marca: String +Cor: String +Placa: String +Velocidade_atual: Inteiro +Marcha_atual: Caractere +Freio_de_mao_puxado: booleano +chave_virada: booleano
+Ligar() +Acelerar_ate(velocidade: Inteiro) +Mudar_Marcha(marcha: Inteiro) +Parar()

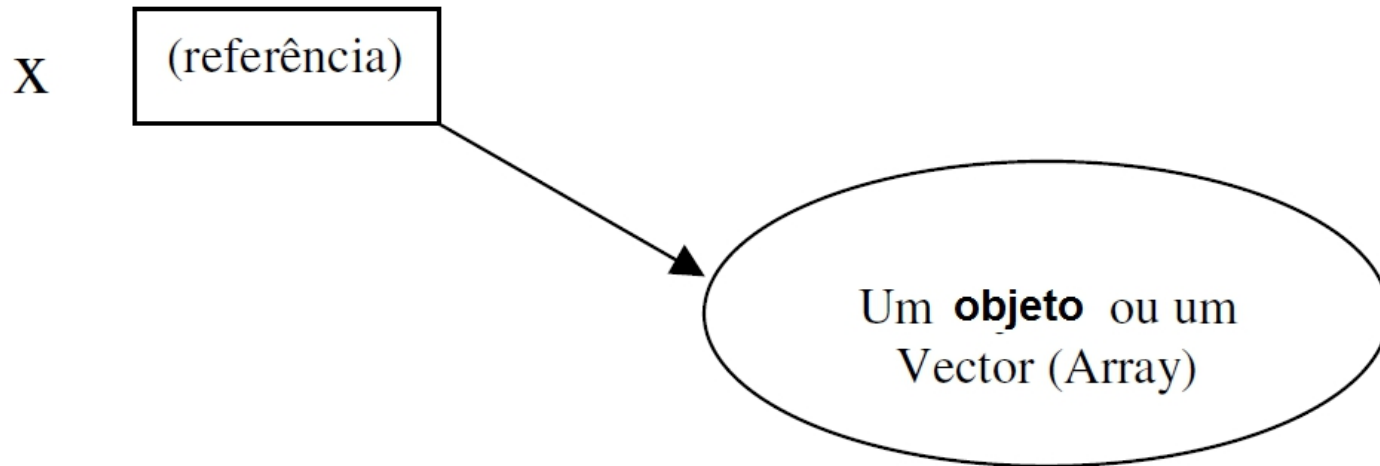
Tipos não primitivos

- Para cada tipo primitivo existe uma classe que encapsula esse tipo:
 - boolean - Boolean
 - byte - Byte
 - char - Character
 - short - Short
 - int - Integer
 - long - Long
 - float - Float
 - double - Double
- Exemplo: instanciação de um objeto da classe Integer

```
Integer i = new Integer(10);
```

Tipos referenciados

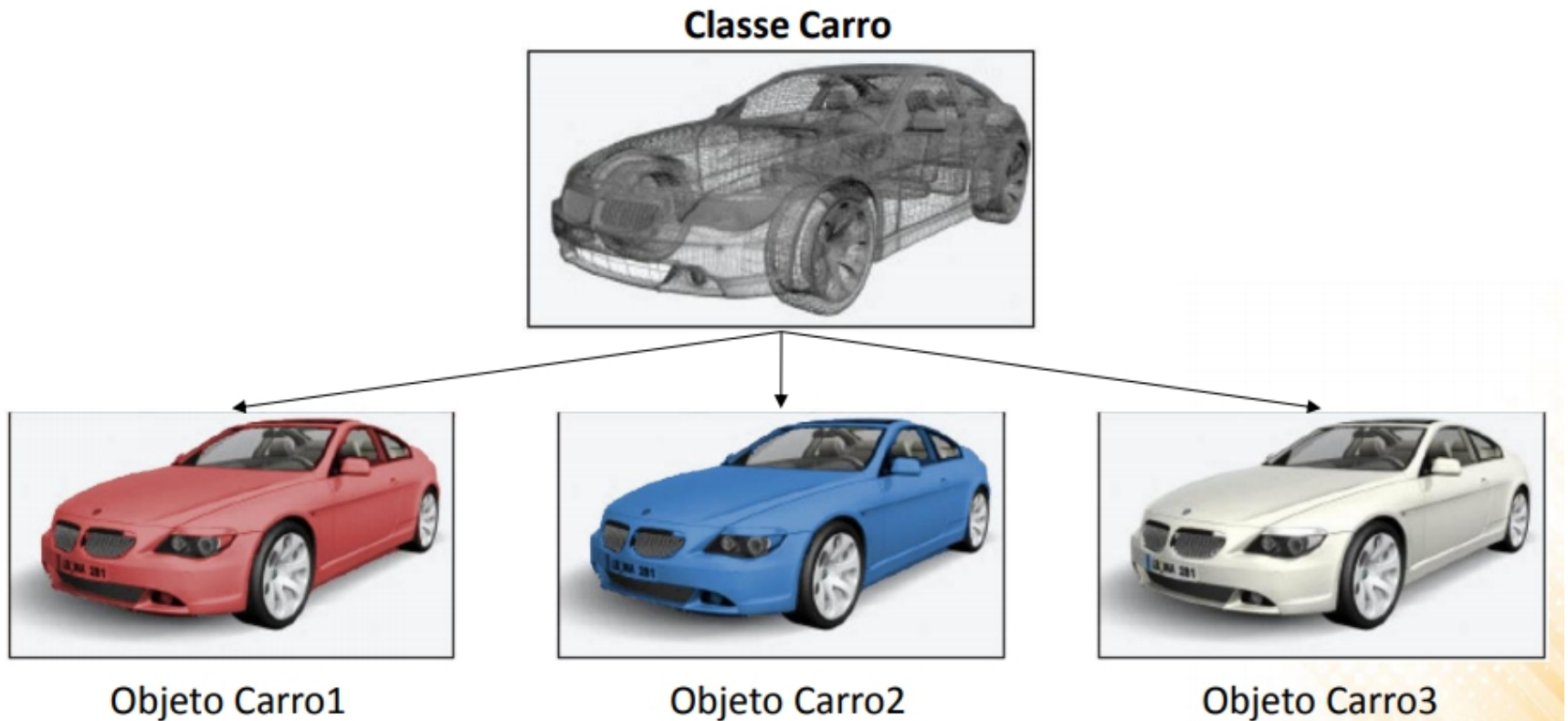
- Vetores e classes são tipos referenciados
- O valor de uma variável de um tipo referenciado é uma referência para o valor ou conjunto de valores representados pela variável



Classes e Objetos

CLASSE CARRO		OBJETO CARRO A	OBJETO CARRO B
Atributos de objeto	Marca	Ford	Mitsubishi
	Modelo	Fiesta	L-200
	Cor	branco	azul royal
	Combustível	gasolina	diesel
Métodos	ligar		
	acelerar		
	frear		

Classes e Objetos



Exemplo de Sistema O.O.

- Controle Pessoal de Mídias
 - Armazenar os dados de um acervo de CDs e DVDs pessoais
- Classes
 - CD
 - DVD
 - Usuário
 - Sistema Principal: controle dos objetos que compõem o sistema

Classes em Java

- Atributos

```
public class Nome_classe {  
    tipo atributo_1;  
    tipo atributo_2;  
    ...  
}
```

Classes em Java

- Métodos

```
public class Nome_classe {  
    tipo atributo_1;  
    tipo atributo_2;  
    ...  
    retorno metodo1(parametros) {  
    }  
    retorno metodo2(parametros) {  
    }  
    ...  
}
```

Classes em Java

- Exemplo de definição da classe **Pessoa**

```
public class Pessoa {  
    public String nome;  
    public int idade;  
  
    public void incIdade() {  
        idade++;  
    }  
}
```

Classes em Java

- Não é possível invocar um método de uma classe sem criar um objeto dela
- A palavra chave *new* cria um novo objeto da classe especificada em Java

Classes em Java

- Exemplo de definição da classe **Pessoa**

```
public class Pessoa {  
    public String nome;  
    public int idade;  
  
    public void incIdade() {  
        idade++;  
    }  
}
```

- Exemplo de utilização

```
Pessoa amiga = new Pessoa();  
amiga.nome = "Lara";  
amiga.idade = 18;
```

Métodos em Java

- Definição pertence sempre a uma classe
 - Realizada dentro do código da classe detentora
- Definição semelhante ao de uma função
 - Tipo de retorno (`void` quando não há retorno)
 - Nome
 - Lista de parâmetros

Exemplo - Atributos de Classes

```
public class Pessoa {  
    public String nome;  
    public static int ct;  
    public void alteraCt() {  
        ct++;  
    }  
    public int getCt() {  
        return ct;  
    }  
}
```

Exemplo - Atributos de Classes

```
public class Principal {  
    public static void main(String a[]) {  
        Pessoa p = new Pessoa();  
        Pessoa p2 = new Pessoa();  
        p.nome = "Joao";  
        System.out.println(p.getCt()+"|"+p2.getCt());  
        Pessoa.ct = 3;  
        System.out.println(p.getCt()+"|"+p2.getCt());  
        p.alteraCt();  
        System.out.println(p.getCt()+"|"+p2.getCt());  
        p2.ct = 44;  
        System.out.println(p.getCt());  
    }  
}
```


Exercício

- Defina uma classe que represente uma conta bancária
 - A conta armazena o nome e cpf do cliente, bem como o saldo do mesmo
 - Deve ser possível realizar operações de saque, depósito e exibição de saldo
 - Considere que não podem ser feitos saques em contas sem saldo

Exemplo – Conta Bancária

```
public class ContaBancaria {  
    public String cpf;  
    public String nome  
    public float saldo;  
  
    public float obtemsaldo(){  
        return saldo;  
    }  
  
    public void saque(float valor){  
        if (saldo >= valor){  
            saldo = saldo - valor;  
        }  
    }  
  
    public void deposito(float valor){  
        saldo = saldo + valor;  
    }  
}
```

Exemplo – Conta Bancária

```
ContaBancaria cElaine = new ContaBancaria();  
ContaBancaria cJoao = new ContaBancaria();  
cElaine.deposito(10000000);  
System.out.println(cElaine.saldo);  
  
cJoao.nome = "Joao";  
cJoao.deposito(100);  
System.out.println(cJoao.saldo);
```

Exercício

- Crie a classe Quadrado que contém
 - O atributo lado
 - Métodos para obter e alterar o valor do lado
 - O método calcula área
 - O método calcula perímetro
- Crie também uma classe TestaQuadrado que faz a chamada da classe Quadrado testando todos os seus métodos

Referências

- DEITEL, H. M., DEITEL, P. J., **Java: Como Programar**, Bookman, São Paulo, 2002
- DORÇA, F., **Notas de Aula de Programação Orientada a Objetos**, disponível em <http://www.facom.ufu.br/~fabiano>
- PAIVA, J. G. S., **Notas de Aula de Programação Orientado a Objetos**