

Paradigma orientado a objetos

CST em Análise e Desenvolvimento de Sistemas

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

Licenciamento



Slides licenciados sob [Creative Commons "Atribuição 4.0 Internacional"](#)

Desenvolvimento de software

Abstração do problema

- Princípio básico do desenvolvimento de software
- **Retirar do domínio do problema detalhes relevantes** e representá-los na linguagem da solução

Desenvolvimento de software

Abstração do problema

- Princípio básico do desenvolvimento de software
- **Retirar do domínio do problema detalhes relevantes** e representá-los na linguagem da solução

Paradigma – forma de como atacar um problema

- A evolução das linguagens de programação influenciaram a forma como os problemas são atacados
- A tecnologia de cada época delimitou como os problemas eram atacados

Paradigmas de programação

Estruturada e procedural

- Fundamentada sobre estruturas de **sequência, decisão e repetição**
 - Desvios condicionais são preferidos a desvios incondicionais
- A solução de cada pequena parte do problema é feita em **funções** e a solução de todo problema consiste na invocação destas funções
 - Dividir para conquistar!
 - Funções podem ser recursivas, ou seja, podem invocar a si mesmas
- **Estrutural** é um subconjunto do paradigma **procedural**
- Exemplos: Pascal, C, Fortran, Cobol

Paradigmas de programação

Orientada a objetos

- Criar objetos que representam entidades do mundo real
- **Cada objeto possui uma tarefa específica** e por meio da **troca de mensagens entre os objetos** é possível realizar uma tarefa computacional
- Ideal para o desenvolvimento de software complexo, porém traz complexidade desnecessária para projetos pequenos
- Exemplos: Smalltalk, C++, Java, Python

Conceitos da Orientação a Objetos

A Orientação a Objetos fundamenta-se sobre 5 conceitos:

- 1 Classes
- 2 Objetos
- 3 Encapsulamento
- 4 Herança
- 5 Polimorfismo

Definição de objeto

Objeto é um item **identificável**, com características (**estado**) e pode ser capaz de fazer algo (**comportamento**)

Definição de objeto

Objeto é um item **identificável**, com características (**estado**) e pode ser capaz de fazer algo (**comportamento**)



**Caneta para
quadro branco**

Cor: Preta
Tinta: 100%

Estado



Comportamento



Identidade

Objetos

Regra de ouro da orientação a objetos

Identificar os estados e comportamentos de objetos do mundo real é um grande passo para se começar a pensar em termos de programação orientada a objetos

- **Estado** (em Java: atributo)
 - Representa as características do objeto
 - Ex: Um carro possui como características uma cor, modelo, velocidade atual
- **Comportamento** (em Java: método)
 - Representa as ações que este objeto é capaz de executar
 - Ex: Um carro pode trocar de marcha, acelerar, frear, etc.

Objetos no domínio do problema

Olhe ao redor e escolha dois objetos. Para estes responda:

- Quais são suas características (**atributos**)?
- O que permitem fazer (**métodos**)?

Objetos no domínio do problema

Olhe ao redor e escolha dois objetos. Para estes responda:

- Quais são suas características (**atributos**)?
- O que permitem fazer (**métodos**)?
- Objetos possuem diferentes níveis de complexidade
 - Lâmpada vs Computador
- Objetos podem conter outros objetos
 - Computador possui um disco rígido

Objetos em sistemas computacionais

Objetos de *software* são semelhantes aos objetos reais

Um objeto armazena seu estado em **atributos** e seu comportamento se dá por meio de **métodos**

Objetos em sistemas computacionais

Objetos de *software* são semelhantes aos objetos reais

Um objeto armazena seu estado em **atributos** e seu comportamento se dá por meio de **métodos**

- **Métodos de um objeto são invocados** para realizar uma computação e podem modificar os valores de seus atributos

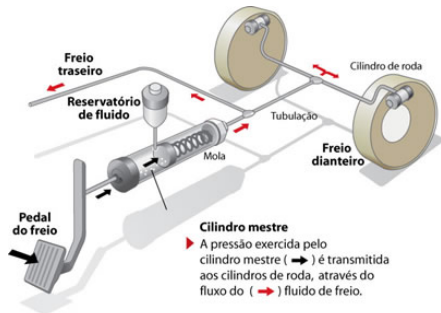
```
// Instancia um objeto da classe Random
Random r = new Random();

// realize uma computação
int i = r.nextInt(10);

// altere o valor armazenado como semente
r.setSeed(1234);
```

Encapsulamento

Exemplo Sistema de freio hidráulico



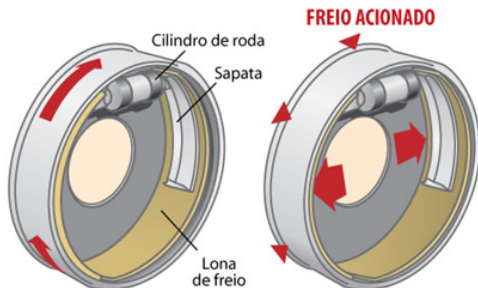
- Sistema de pistões e mangueiras por onde circula o fluido de freio
- Ao pisar no pedal de freio, aciona-se o cilindro mestre que irá pressurizar o fluido.
- Fluido transmite a pressão exercida no pedal até as rodas

Encapsulamento

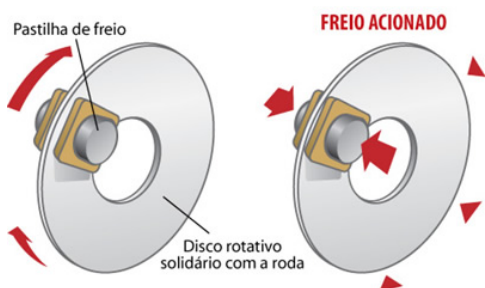
Exemplo Sistema de freio hidráulico

- 1 Como se faz para frear um carro com freio a tambor?
- 2 Como se faz para frear um carro com freio a disco?

FUNCIONAMENTO DO FREIO A TAMBOR



FUNCIONAMENTO DO FREIO A DISCO



Encapsulamento

Definição

Processo de esconder todos os detalhes de um objeto que não contribuem para as suas características essenciais

- A implementação de um método pode ser alterada sem que implique na alteração do código do objeto que o invoca
- A **interface** de um objeto corresponde ao que ele conhece e ao que ele sabe fazer, no entanto **sem descrever como ele conhece ou faz**

Encapsulamento

Exemplo: Como sortear um número

```
// Instancia um objeto da classe Random
Random r = new Random();

// Retorna um número pseudoaleatório dentro do intervalo
int i = r.nextInt(10);

// Alterar o valor da semente usada para gerar números
r.setSeed(1234);
```

- Como o número pseudoaleatório é gerado?
- Onde ele armazena essa semente?

Classes

- **Classe** é uma planta (projeto) que indica como os **objetos** deverão ser construídos

Fusca

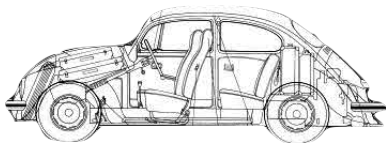
- Cada carro é construído com base em um mesmo projeto de engenharia e por consequência todos carros possuirão os mesmos componentes

Classes

- **Classe** é uma planta (projeto) que indica como os **objetos** deverão ser construídos

Fusca

- Cada carro é construído com base em um mesmo projeto de engenharia e por consequência todos carros possuirão os mesmos componentes



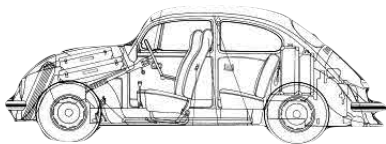
Classe

Classes

- **Classe** é uma planta (projeto) que indica como os **objetos** deverão ser construídos

Fusca

- Cada carro é construído com base em um mesmo projeto de engenharia e por consequência todos carros possuirão os mesmos componentes



Classe



Objetos

Classe Carro

Representada em Java e na Linguagem de Modelagem Unificada (UML)

```
public class Carro{  
  
    // atributos  
    private double velocidade;  
    private String modelo;  
  
    // métodos  
    public void acelerar(double intensidade){  
        this.velocidade += intensidade;  
    }  
  
    public String obterModelo(){  
        return this.modelo;  
    }  
  
    public double obterVelocidade(){  
        return this.velocidade;  
    }  
}
```

Carro
-velocidade: double -modelo: String
+acelerar(double intensidade): void +obterModelo(): String +obterVelocidade(): double

Abstração

- Processo mental que nós seres humanos **nos atemos aos aspectos mais relevantes** de alguma coisa, ao mesmo tempo que **ignoramos os aspectos menos importantes**
- Nos permite gerenciar a complexidade de um objeto e nos concentramos somente nas características essenciais



Abstração é dependente do contexto. O que é importante em um contexto pode não ser importante em outro

Abstração

Quando vou comprar um carro



Fonte: Quatro Rodas

Abstração

Quando vou alugar um carro



Utilitário
a partir de
R\$ 116/dia

Pesquisar



Luxo
a partir de
R\$ 210/dia

Pesquisar



Econômico
a partir de
R\$ 66/dia

Pesquisar



Standard
a partir de
R\$ 74/dia

Pesquisar



Minivan
a partir de
R\$ 118/dia

Pesquisar



SUV
a partir de
R\$ 107/dia

Pesquisar



Intermediário
a partir de
R\$ 69/dia

Pesquisar



Fonte: Rentcars

Abstração: Exemplo

■ Sistema para revenda de carros

- Necessita de um sistema para controlar os carros do estoque
- Quais são as características essenciais para um carro?

■ Jogo de Fórmula 1

- Um usuário deseja controlar seu carro no jogo
- Quais são as características essenciais para um carro?

Abstração: Exemplo

■ Sistema para revenda de carros

- Necessita de um sistema para controlar os carros do estoque
- Quais são as características essenciais para um carro?
 - Atributos: código, marca, modelo, ano, preço
 - Métodos: obterCódigo, obterModelo, definirPreço, etc.

■ Jogo de Fórmula 1

- Um usuário deseja controlar seu carro no jogo
- Quais são as características essenciais para um carro?
 - Atributos: código, cor, velocidadeAtual, velocidadeMaxima
 - Métodos: frear, acelerar, trocarPneus, etc.

Codificando em Java

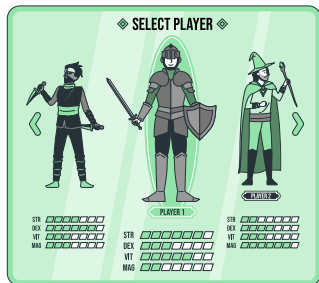
Classe para representador um contador



Fonte: Google Images

- A classe Contador possui um único **atributo**
 - `valorAtual`
- A classe provê **métodos** para:
 - Atribuir um valor ao contador
 - Incrementar em 1 o valor do contador
 - Obter o atual valor do contador

Personagem de um jogo



Fonte: Image by storyset on Freepik

- Modele um personagem de um jogo de aventura
- Quais são os atributos e métodos essenciais para esse personagem?
- Implemente uma classe Java para representar esse personagem
- Crie uma outra classe com o método `main` e instancie um objeto da classe `Personagem`

Batedeira planetária



Fonte: Google Images

- Pense em um contexto e realize o processo de abstração para coletar as informações essenciais (atributos e métodos) deste objeto para o contexto escolhido
- Implemente uma classe Java para representar esse objeto
- Crie uma outra classe com o método `main` e instancie um objeto da classe Batedeira

Disciplina em um sistema acadêmico



- Modele uma classe para representar uma disciplina de graduação em um sistema acadêmico
- Quais são os atributos e métodos essenciais para essa disciplina?
- Implemente uma classe Java para representar essa disciplina
- Crie uma outra classe com o método `main` e instancie um objeto da classe `Disciplina`