

Programação Orientada a Objeto

Stephenson Galvão

stepgalvao[arroba]gmail[ponto]com

Roteiro

- Paradigmas de programação
- Modelos
- Abstração
- Programação OO
- Objetos, classe, atributos e operações
- Classes em java

Paradigmas de programação

Paradigma é um conjunto de regras que estabelecem fronteiras e descrevem como resolver os problemas dentro destas fronteiras.

Forma de desenvolvimento de uma solução de software.

Paradigmas de programação

- Alguns paradigmas
 - Imperativo.
 - Estrutural.
 - Funcional.
 - Orientado a Objetos.

Paradigmas de programação

- Paradigma imperativo

- Baseá-se na arquitetura de Von Neumann.
- Mais antigo.
- Características centrais
 - Variáveis.
 - Atribuições.
 - Sequências.
- Linguagens
 - C, Modula-2, Pascal.

```
var fat: inteiro;  
var n: inteiro;  
var r: inteiro;  
inicio  
    fat := 1;  
    n:=10;  
    enquanto (n>1) faça:  
        fat := fat * n;  
        n := n - 1;  
    fim enquanto;  
    r := fat;  
fim;
```

Paradigmas de programação

- Paradigma Estruturado

- Subconjunto do Imperativo.
- Mais utilizado.
- Resume-se em 3 estruturas:
 - Sequência, Decisão e Interação.
- Programação modular.
 - Sub-rotinas e funções.
- Linguagens
 - C; Pascal; Cobol

```
funcao fatorial (n: inteiro):inteiro
inicio funcao
    var fat: inteiro;
    fat := 1;
    enquanto (n>1) faca
        fat := fat * n;
        n := n - 1;
    fim enquanto;
fim funcao
inicio
    var r: inteiro;
    r=fatorial(10)
end;
```

Paradigmas de programação

- Paradigma Funcional
 - Programas como funções matemáticas.
 - Fatorial
 - $X=0, 1$
 - $X>0, X * \text{fatorial}(x-1)$.
 - Não possuem alterações de valores
 - As funções sempre devem possuir retorno.
 - Linguagens
 - Hask, LISP, Ocamel, Lambda.

fatorial 0 = 1

fatorial n = n * fatorial (n - 1)

Paradigmas de programação

- Paradigma Orientado a Objetos.
 - Baseada em objetos
 - Estados.
 - Ações e Mensagens.
 - Linguagens de programação:
 - Java, C++, C#, Python, Smalltalk

```
public Fatorial{
    int fat;

    public Fatorial(int fat){
        this.fat=fat;
    }

    public int calcular(){
        If (fat = 0) {
            return 1;
        }else
            fat--;
            calcular();
        }
    }
}

Programa{
    Fatorial fat = new Fatorial(3)
    fat.calcular();
}
```


Paradigma Orientado a Objeto

- Comunidade de agentes (objetos) que interagem entre si
- Cada agente tem um papel a cumprir, oferece um serviço ou realiza uma ação que é usada por outros objetos.
- Objetos são definidos através de modelos (classe).

Modelo Circulo
inicio modelo

var raio:Real

funcao calcularArea():
inicio funcao
 variavel area:Real
 area = raio*raio*3,14;
 imprime(area);

 fim funcao
fim modelo

Programa AreaCirculo

variavel circulo: Circulo

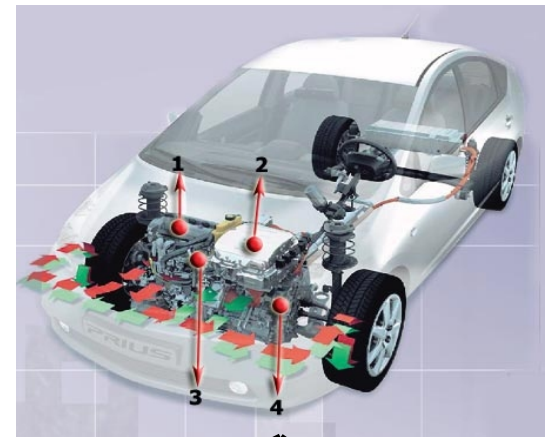
Inicio programa

circulo= novo objeto Circulo;
circulo.raio=10;
circulo.calcularArea();

Fim programa

Paradigma Orientado a Objeto

- Sistemas orientado a objeto (exemplos)
 - Visão mais realista.



Paradigma Orientado a Objeto

- Quais são os “objetos” presentes no sistema IFPI?



Paradigma Orientado a Objeto



- Quais são os “objetos” presentes no sistema IFPI?
 - Professor
 - Aluno
 - Zelador
 - Diretor
 - Guarda

Paradigma Orientado a Objeto

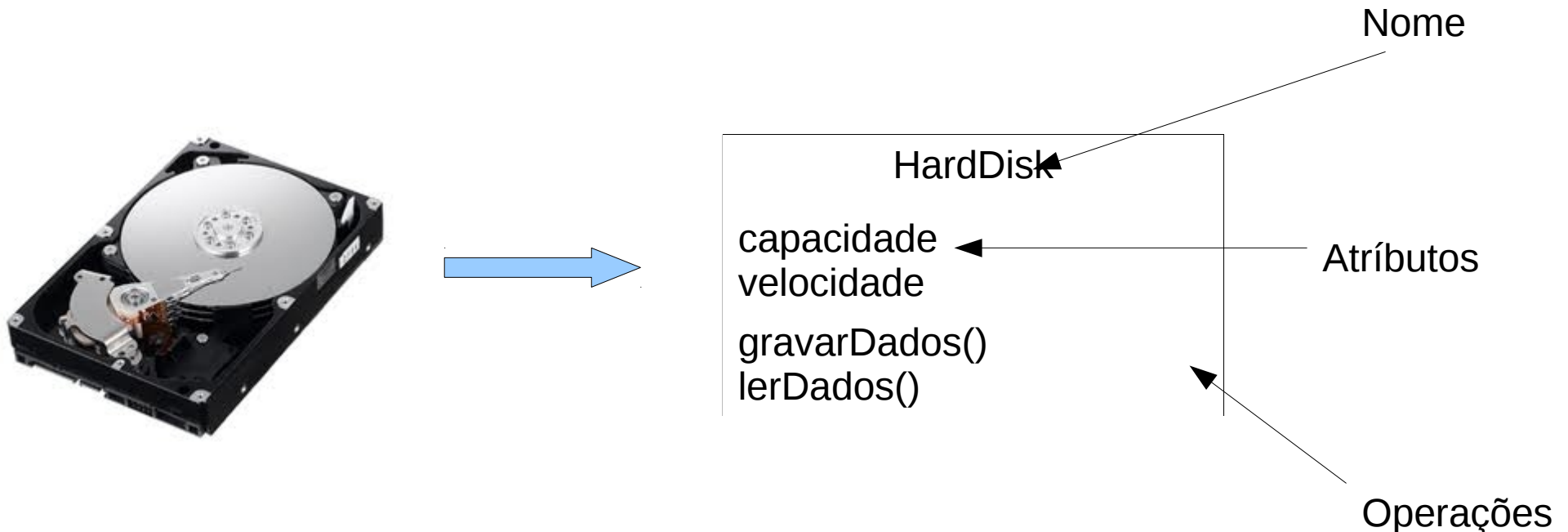
- Objeto

Um objeto é uma **entidade** do sistema que tem **características** e **comportamentos próprios**. Podem representar entidades concretas ou abstratas.



Modelos

- Representações simplificadas de objetos.
- Descreve os atributos e ações de uma conjunto de objetos.
- Denominados de classe.



Modelos

- Modelo do “objeto” aluno?



Modelos



Aluno

nome
matricula
frequencia
notas
adicionarPresenca(dia,situacao)
adicionarNota(nota)
calcularMedia()

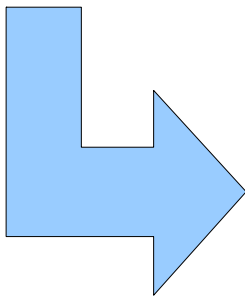
Modelos

Qual é o modelo correto?



Aluno

nome
matricula
frequencia
notas
adicionarPresenca(dia,situacao)
adicionarNota(nota)
calcularMedia()

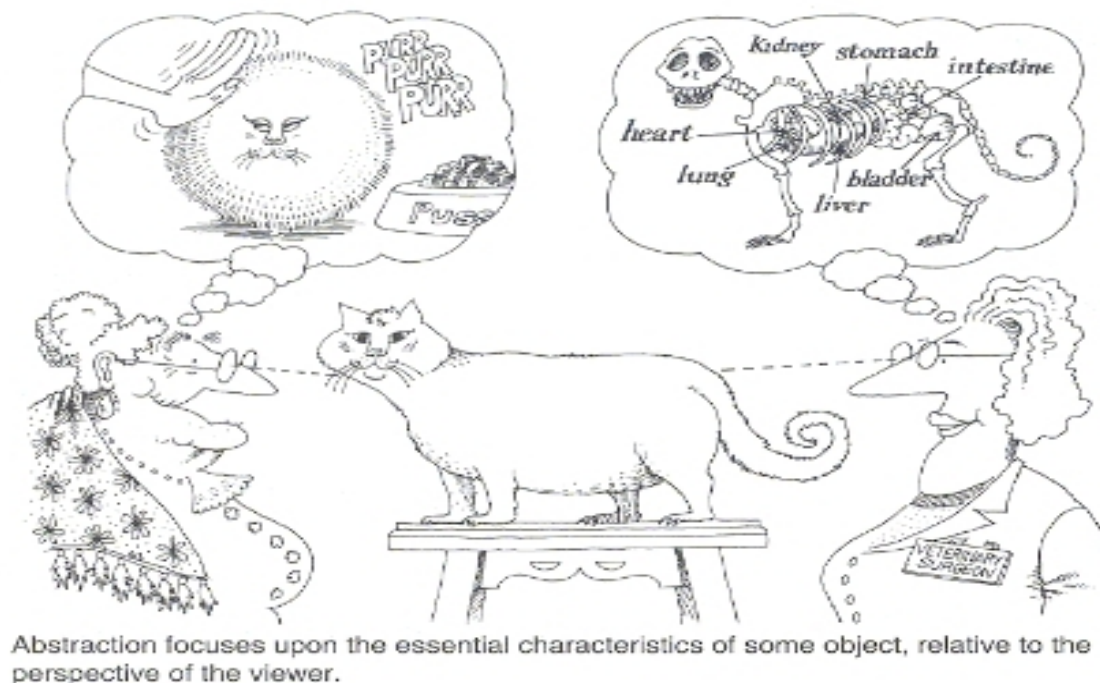


Aluno

nome
matricula
altura
idade
matricularDisciplina(disciplina)
cursarDisciplina(disciplina)

Abstração

Abstração consiste de focalizar nos aspectos essenciais para o sistema inerentes a uma entidade.



- Modelar o objeto Professor na visão CCA e visão o RH

Modelos

- Conta Bancária



Modelo para uma
conta bancária?

Modelos

- Conta Bancária



Conta
saldo
deposito(valor) saque(valor)

Modelos

- Representações

Conta
saldo
deposito(valor) saque(valor)

modelo Conta
inicio do modelo

dado saldo;

operacao deposito(valor)

inicio

saldo= saldo+valor;

fim

operacao saque(valor)

inicio

saldo=saldo-valor;

fim

fim do modelo

Modelo

- Representação Intermediária

```
modelo <NomeModelo>  
inicio do modelo  
    dado <nomeDado>  
  
    operacao <nomeOperacao> (valor)  
    inicio  
        <comandos>  
    fim  
fim do modelo
```

```
modelo Conta  
inicio do modelo  
  
    dado saldo;  
  
    operacao deposito(valor)  
    inicio  
        saldo= saldo+valor  
    fim  
  
    operacao saque(valor)  
    inicio  
        saldo=saldo-valor  
    fim  
fim do modelo
```

Orientações Sobre o Modelo

- Notação
 - Distribuição vertical.
 - Indentação horizontal (hierarquia).
 - Comentário = //.
 - Cada comando termina com “;” .
 - Comparação (==), atribuição (=).
 - Nome do Modelo
 - Inicial de cada palavra maiuscula.
 - Não utilizar assentuação nem espaço.
 - Nome de dados e Operações:
 - 1º palavra inicial minuscula demais inicial maiuscula.
 - Não utilizar assentuação nem espaço.

Modelos

- Exemplo

modelo Carro
inicio do modelo

dado velocidade

operacao acelerar()

inicio

velocidade++;

fim

operacao frear(valor)

inicio

velocidade--;

fim

fim do modelo

Exercício

- Na disciplina de POO cada aluno tem uma matrícula, uma média e uma frequência. No decorrer da disciplina o professor deve informar a média do aluno, assim como sua frequência. Ao final, de acordo com a média do aluno deve ser informado se aluno passou ou ficou reprovado.

```
modelo <NomeModelo>  
inicio do modelo  
    dado <nomeDado>  
  
    operacao <nomeOperacao> (valor)  
    inicio  
        <comandos>  
    fim  
fim do modelo
```

Exercício

- Para o sistema, uma lâmpada pode estar acesa ou apagada. Além disso, a entidade lâmpada deve possuir ações que possam mudar seu estado para acesa ou apagada.

```
modelo <NomeModelo>  
  inicio do modelo  
    dado <nomeDado>  
  
    operacao <nomeOperacao> (valor)  
    inicio  
      <comandos>  
    fim  
  fim do modelo
```

Modelo Mesa

<i>Restaurante Caseiro Hipotético</i>		
Mesa 1 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>	Mesa 2 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>	Mesa 3 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>
Mesa 4 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>	Mesa 5 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>	Mesa 6 <div><input type="text"/> kg refeição <input type="text"/> sobremesa <input type="text"/> refriger.2 L. <input type="text"/> refriger.600mL. <input type="text"/> refriger.lata <input type="text"/> cerveja</div>

Tarefa de Casa

Passar os modelos de Conta, Aluno, Lampada e Mesa para classe em Java.

- Referências:
 - Capítulo 2 do livro de Rafael Santos (Recomendado).
 - Capítulo 3 do livro do Deitel (Avançado).