

Introdução à programação orientada a objetos

Bibliografia

- BARKER, J. **Beginning Java Objects**. From Concepts to Code. New York: Springer, 2ª ed. 2005.
- SANTOS, Rafael. **Introdução à programação orientada a objetos usando JAVA.2**. ed. Rio de Janeiro : Elsevier, 2013. 313 p, il.
- WINBLAD, Ann L; EDWARDS, Samuel D; KING, David R. **Software orientado ao objeto**. Sao Paulo : Makron Books, 1993. xxvi, 314p, il.

Paradigmas de programação

Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa

```
program Fibonacci;

function fib(n: Integer): Integer;
var a: Integer = 1;
    b: Integer = 1;
    f: Integer;
    i: Integer;
begin
  if (n = 1) or (n = 2) then
    fib := 1
  else
    begin
      for i := 3 to n do
        begin
          f := a + b;
          b := a;
          a := f;
        end;
      fib := f;
    end;
end;

begin
  WriteLn(fib(10));
end.
```

Imperativo

```
import Text.Printf

fib :: Int -> Int
fib 0 = 0
fib 1 = 1
fib n = fib (n-1) + fib (n-2)

main = printf "%d\n" (fib 10)
```

Funcional

```
fib(1, 1).
fib(2, 1).

fib(X, Y):-
  X > 1,
  X1 is X - 1,
  X2 is X - 2,
  fib(X1, Z),
  fib(X2, W),
  Y is W + Z.

main :-
  fib(10,X), write(X), nl.
```

Lógico

```
public class Fibonacci {

  public long fibonacci(int n) {
    if(n == 0) {
      return 0;
    } else if(n == 1) {
      return 1;
    } else {
      return fibonacci(n - 1) +
        fibonacci(n - 2);
    }
  }

  public static void main(String[] args) {
    Fibonacci f = new Fibonacci();
    System.out.printf(f.fibonacci(10));
  }
}
```

Orientado a objetos

Paradigma orientado a objetos

- Surgiu na década de 60
- Primeiras linguagens comerciais surgiram na década de 90
- É uma visão contemporânea que utiliza a perspectiva de objetos
- Capaz de ser usado em qualquer tipo de sistema
- Principais objetivos são:
 - Melhorar a compreensão do sistema
 - Auto grau de reutilização
 - Facilidade de manutenção
 - Facilidade de evolução
 - Maior qualidade
 - Maior produtividade e menor custo
- Em contrapartida:
 - Maior curva de aprendizagem
 - Programas maiores
 - Não recomendável para qualquer tipo de problema

Conceitos básicos de Programação orientada a objetos (POO)

Classes e objetos

- Em programação orientada a objetos:
 - Um **objeto** geralmente representa um elemento do mundo real. Todo objeto pertence a uma classe.
 - Uma **classe** é um modelo que descreve as características comuns dos seus objetos.

Fazendo uma analogia



Objetos



Classe



Objetos

Problema a ser resolvido

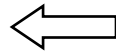
- Calcular o IMC (índice de massa corpórea) de Marta:



- Nome: Marta da Silva
- Idade: 21 anos
- Altura: 1,71 m
- Peso: 56 kg
- Cor preferida: verde
- Signo: aquário
- Naturalidade: Blumenau
- etc

Formas de caracterizar Marta

$$IMC = \frac{Peso}{Altura^2}$$

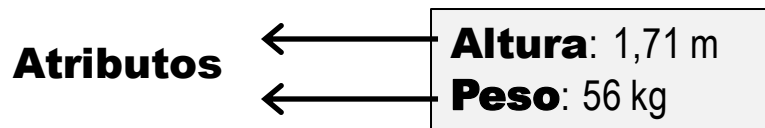


- Altura: 1,71 m
- Peso: 56 kg

*Formas uteis de caracterizar
Marta para resolver este problema*

Objetos – Atributos e operações

- Objetos são caracterizados por um conjunto de **atributos**.
Exemplo: o objeto que representa a Marta é caracterizado através da altura e do peso.



- Afirmamos que os objetos possuem um estado. O estado corresponde ao valor de seus atributos

Altura: **1,71 m**
Peso: **56 kg**

Estado do objeto

O estado do
objeto pode
mudar.

Altura: **1,71 m**
Peso: **56,5 kg**

Novo estado do objeto

- Observar que o valor de um atributo é um dado.

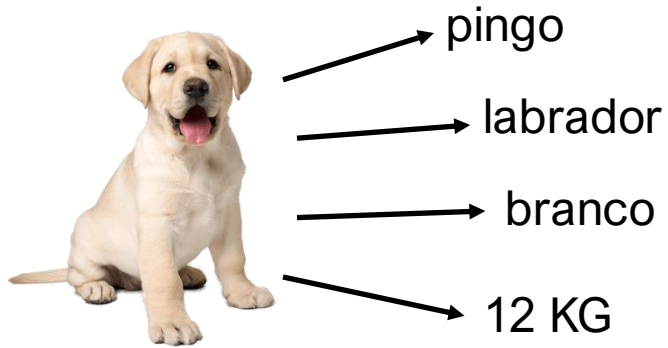
Objetos – Atributos e operações

- “No desenvolvimento de software orientado a objetos, primeiro damos foco às estruturas de dados” (BAKER, 2005).
- Os dados estão contidos dentro do objeto e pertencem apenas aquele objeto.
- Além de dados, os objetos são capazes de executar operações
 - As operações podem executar alguma ação com os dados do próprio objeto.
 - Exemplo: o objeto que representa a Marta é capaz de calcular o IMC da Marta.

Classe

- Todo objeto que se quer criar pertence a uma *classe* de objetos
- Através da classe definimos:
 - Quais atributos (características) os objetos podem possuir
 - Quais operações (comportamentos) os objetos podem realizar
- Toda classe possui um nome

Exemplo



Objeto 1 da classe Cachorro

- Nome: pingo
- Raça: labrador
- Cor: branco
- Peso: 12 KG

Classe "Cachorro":

Conjunto de atributos

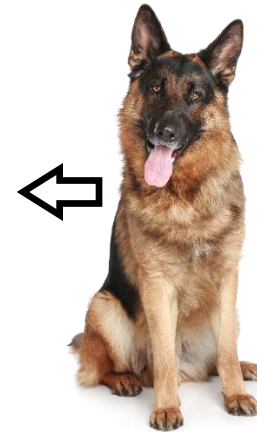
- Nome
- Raça
- Cor
- Peso

Conjunto de operações

- Latir
- Abanar o rabo
- Pegar coisas

Objeto 2 da classe Cachorro

- Nome: brutus
- Raça: pastor alemão
- Cor: marrom
- Peso: 21 KG



Resumindo...

- Conforme (BARKER, 2005):
 - Objeto é uma construção de software que empacota *estado* (dados) e comportamento (funções) que representam uma abstração do mundo real;
 - Uma classe é uma abstração que descreve as características comuns de todos os objetos num grupo de objetos comuns;
 - Uma classe pode ser vista como sendo um modelo para criar objetos.

Exercitar

- Imagine atributos para representar:
 - Aluno da FURB;
 - Sala de aula da FURB;
 - Carro em uma concessionária;
 - Obs: no máximo 5 atributos para cada opção.