Programação Orientada a Objetos

FURB
UNIVERSIDADE DE BLUMENAU

Paradigmas de programação

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

Imperativo:

- **Definição**: Baseia-se em comandos sequenciais para modificar o estado do programa.
- **Exemplo**: Linguagens como C, Pascal e Python podem ser usadas de forma imperativa.

```
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.
```

Paradigmas de programação

Funcional:

•**Definição**: Trata a computação como a avaliação de funções matemáticas e evita estados e dados mutáveis.

•**Exemplo**: Haskell, Lisp.

```
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)
```

Paradigmas de programação

Lógico:

•**Definição**: estilo de programação baseado em lógica formal, onde programas consistem em fatos e regras que descrevem relações entre dados.

•**Exemplo**: Prolog, Datalog e Mercury.

```
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.
```

Paradigmas de programação

Orientado a Objetos:

•**Definição**: Organiza o código em "objetos", que são instâncias de classes, combinando dados e comportamentos.

•**Exemplo**: Java, C++, Python (quando usada de forma orientada a objetos).

```java
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));
  }

}
```

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)

FURB
UNIVERSIDADE DE BLUMENAU

Fazendo uma analogia

Classe

Objetos

Objetos

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** descreve as características comuns dos seus 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

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

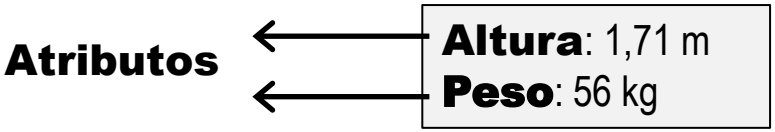
- Altura: 1,71 m
- Peso: 56 kg

Formas uteis de caracterizar Marta para resolver este problema

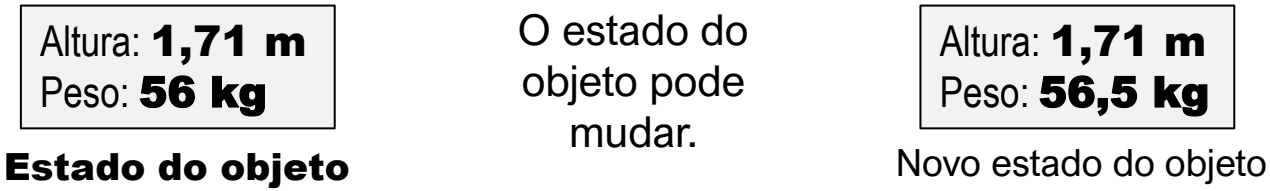
Objetos – Atributos e operações

- 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



- 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 os objetos podem possuir
  - Quais operações os objetos podem realizar

- Toda classe possui um nome

Exemplo

pingo

labrador

branco

12 KG

Classe “Cacharro”:

Conjunto de atributos
- Nome
- Raça
- Cor
- Peso

Conjunto de operações
- Latir
- Abanar o rabo
- Pegar coisas

Objeto 1 da classe Cachorro
- Nome: pingo
- Raça: labrador
- Cor: branco
- Peso: 12 KG

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.