Universidade Presbiteriana Mackenzie

Projeto de Software

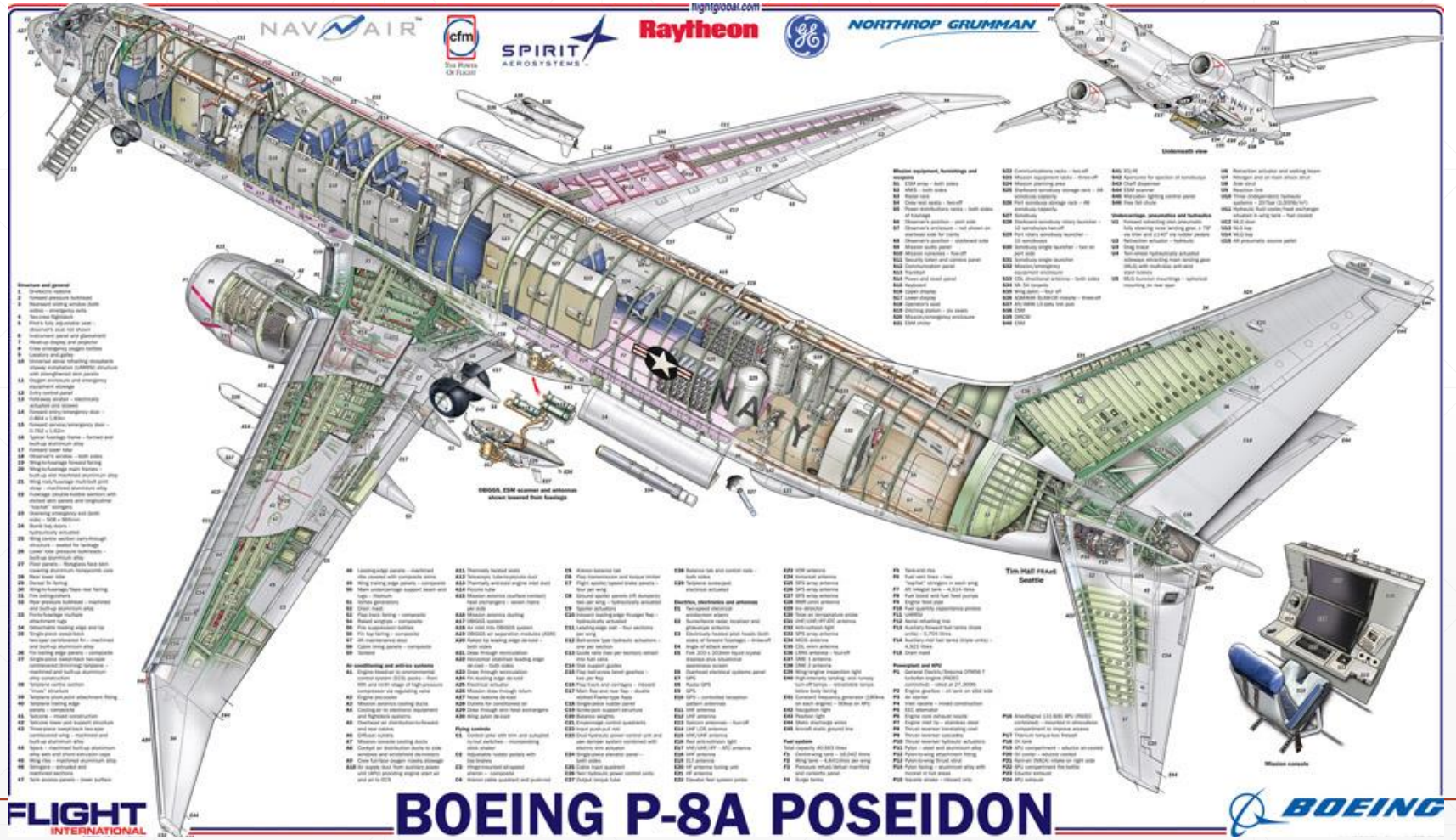
Ana Claudia Rossi

Faculdade de Computação e Informática

Tópicos Abordados

- Contexto da Disciplina
- Fases de Desenvolvimento
- Importância da Modelagem de Sistemas de Software
- Orientação a Objetos

Abstração do Avião



Sistemas de Informação

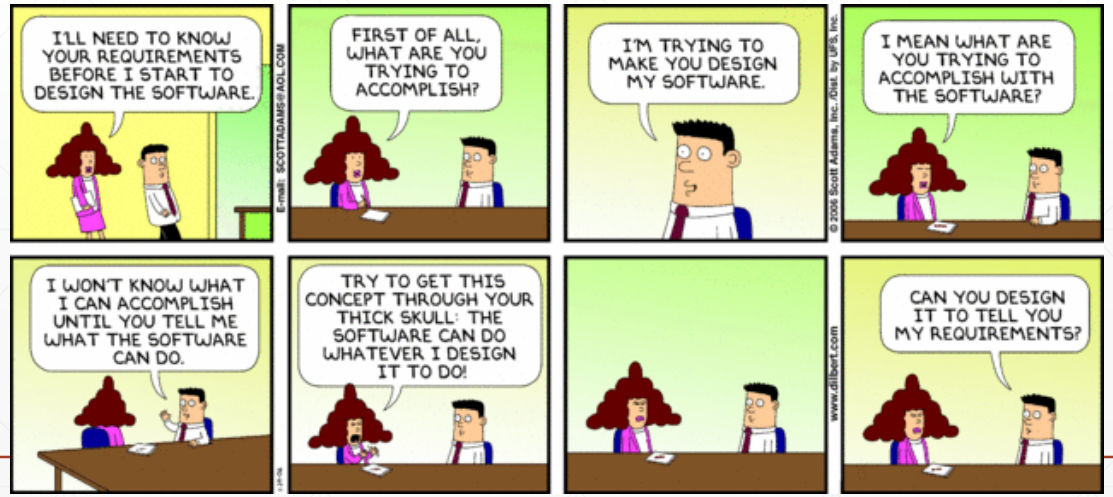
“Combinação de pessoas, dados, processos, interfaces, redes de comunicação e tecnologia que interagem com o objetivo de frnecer suporte e melhorar o processo de negócio de uma organização com relação as informações que fluem nela”

“Desejo de compartilhar recursos”

Modelagem de Sistemas

- Característica de Projeto → Complexidade
  - Casa de Cachorro → Casa Térrea → Sobrado → Prédio

- E no caso de Software → Complexidade?
  - O que é um software simples?
  - O que é um software complexo?

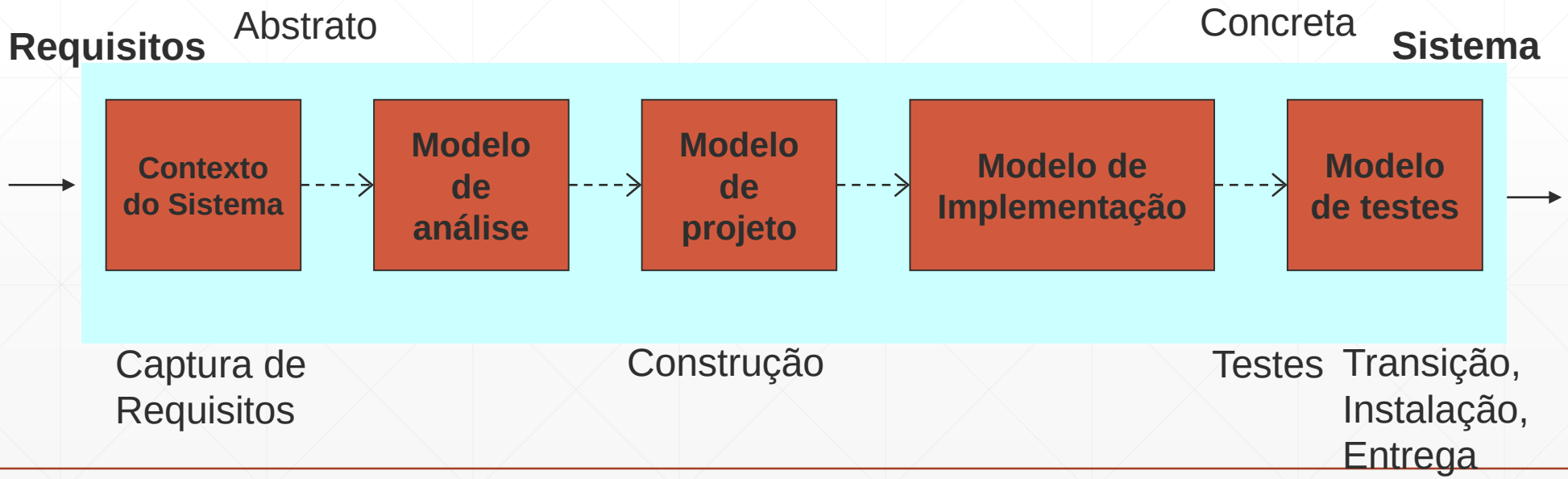


Por que modelar?

- Gerenciamento da complexidade
- Comunicação entre as pessoas envolvidas
- Redução dos custos de desenvolvimento
- Previsão do comportamento futuro do sistema.

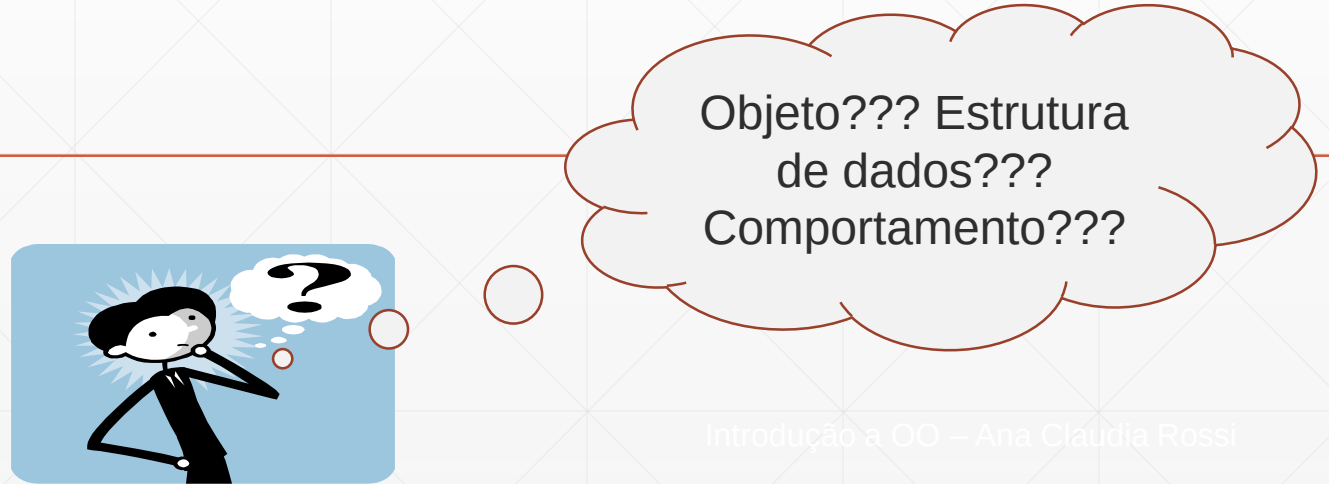
Modelagem de Sistemas de Software consiste na utilização de notação gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando as várias perspectivas diferentes e complementares.

Principais atividades do desenvolvimento de software



Orientado a Objetos

Organizamos o software como uma coleção de objetos distintos, que incorporam estrutura de dados e comportamento

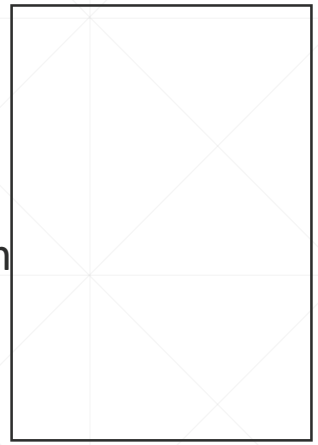


Um pouco de história

- Alay Curtis Kay – um dos pais da Orientação a Objetos
  - Formulou a analogia biológica
    - Sistema de software funciona-se como um ser vivo → cada célula interagiria com outras células através do envio de mensagens para realizar um objetivo em comum. Além disso, cada célula se comportaria como uma unidade autônoma.
  - Princípios da Orientação a Objetos – Alan Kay
    1. Qualquer coisa é um objeto
    2. Objetos realizam tarefas por meio da requisição de serviços a outros objetos
    3. Cada objeto pertence a uma determinada classe. Uma classe agrupa objetos similares
    4. A classe é um repositório para comportamento associada ao objeto
    5. Classes são organizadas em hierarquias

Retângulo

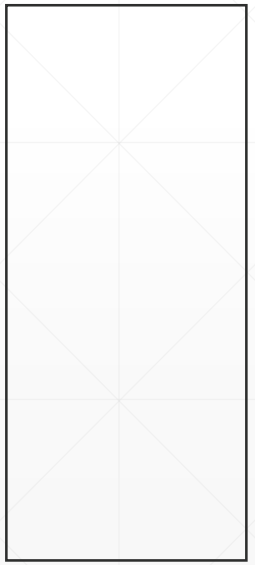
P→ perímetro
A → Área



P = 2*4+2*3 = 14 cm
A = 4*3= 12 cm^2

P = 2*1+2*6 = 14 cm
A = 1*6= 6 cm^2

P = 2*2+2*3 = 10 cm
A = 2*3= 6 cm^2

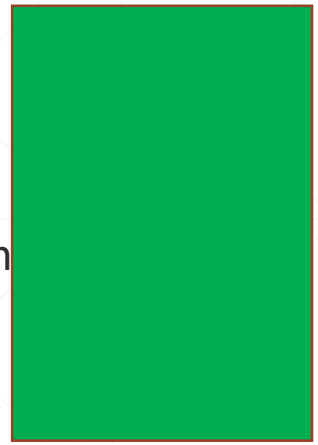


P = 2*5+2*2 = 14 cm
A = 5*2= 10 cm^2

Retângulo

P→ perímetro
A → Área

Cor: verde



Altura: 4 cm

Base: 3 cm

P = 2*4+2*3 = 14 cm
A = 4*3= 12 cm^2

Cor: Rosa

Altura: 1cm

Base: 6 cm

P = 2*1+2*6 = 14 cm
A = 1*6= 6 cm^2

Cor: Roxo

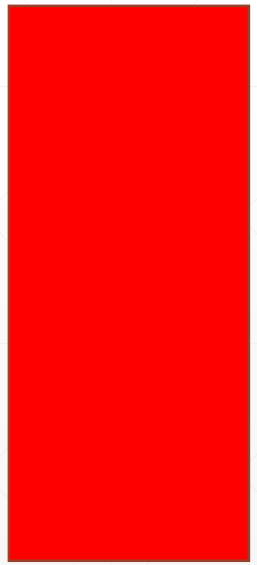
Altura:2 cm

Base: 3 cm

P = 2*2+2*3 = 10 cm
A = 2*3= 6 cm^2

Cor: vermelho

Altura: 5 cm



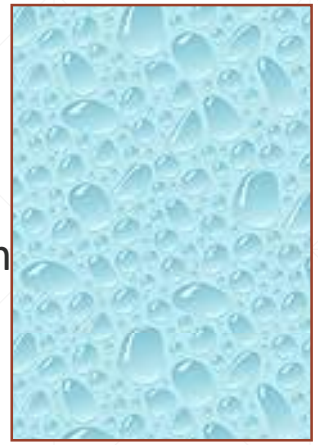
Base: 2 cm

P = 2*5+2*2 = 14 cm
A = 5*2= 10 cm^2

Retângulo

P→ perímetro
A → Área

Textura: gotas.jpg



Altura: 4 cm

Base: 3 cm

P = 2*4+2*3 = 14 cm
A = 4*3= 12 cm^2

Textura: juta.jpg

Altura: 1cm

Base: 6 cm

P = 2*1+2*6 = 14 cm
A = 1*6= 6 cm^2

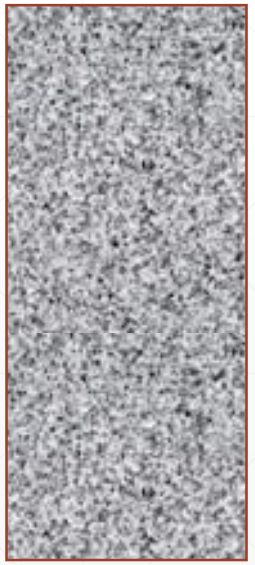
Textura: papelamassado.jpg

Altura:2 cm

Base: 3 cm

P = 2*2+2*3 = 10 cm
A = 2*3= 6 cm^2

Textura: granito.jpg



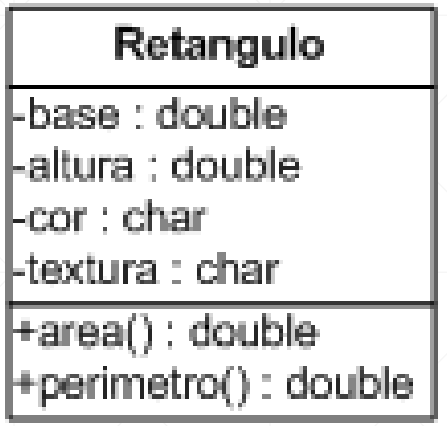
Altura: 5 cm

Base: 2 cm

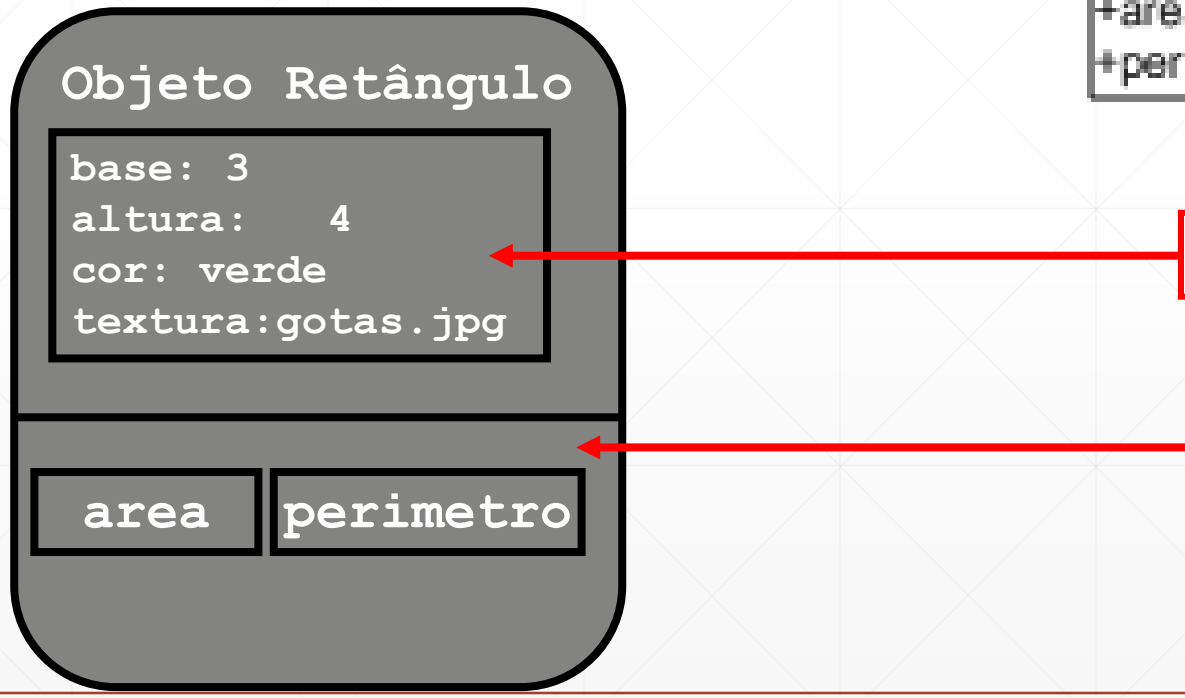
P = 2*5+2*2 = 14 cm
A = 5*2= 10 cm^2

Orientação a Objetos

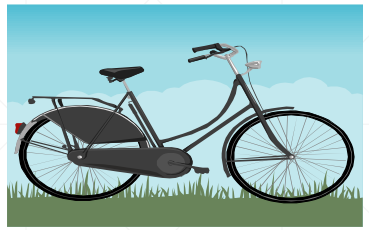
Classe



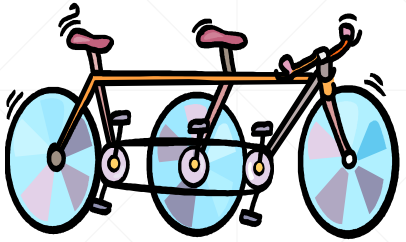
Analogia:



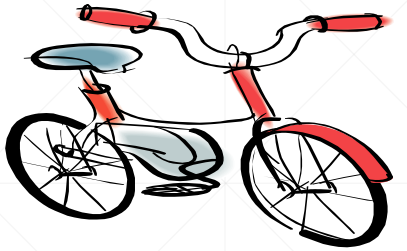
Diferentes Bicicletas – cada uma com sua identidade



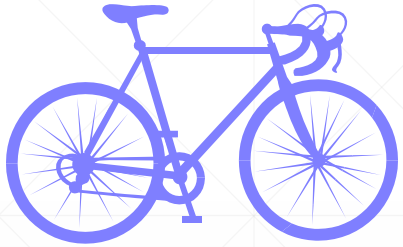
Bicicleta do João



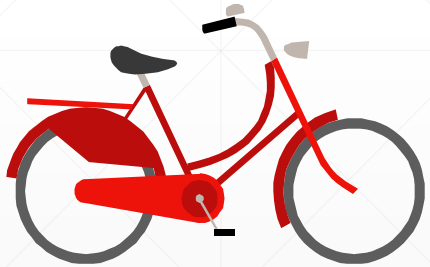
Bicicleta do casal
Roberta e Francisco



Bicicleta do
Pedrinho



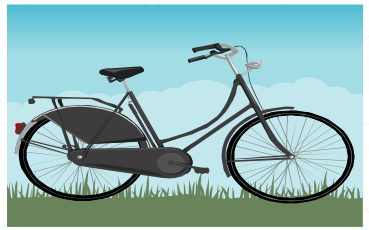
Bicicleta do atleta
José Ouro



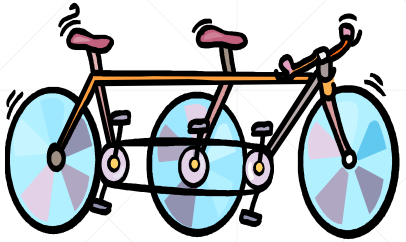
Bicicleta da Maria

Identidade→ significa que os dados são quantizados em entidades distintas e distinguíveis chamadas objetos

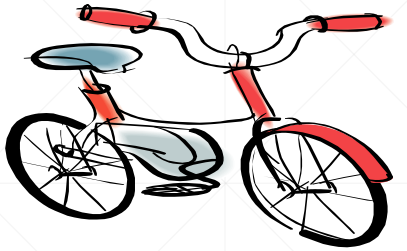
Diferentes Bicicletas – cada uma com sua identidade



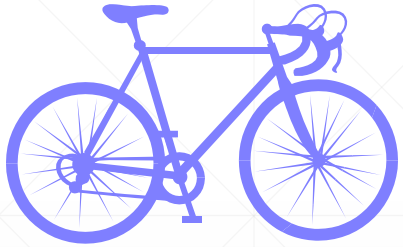
Bicicleta do João



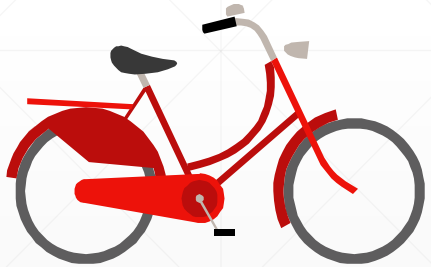
Bicicleta do casal
Roberta e Francisco



Bicicleta do
Pedrinho



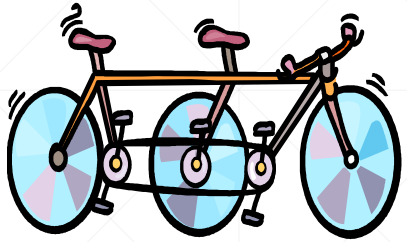
Bicicleta do atleta
José Ouro



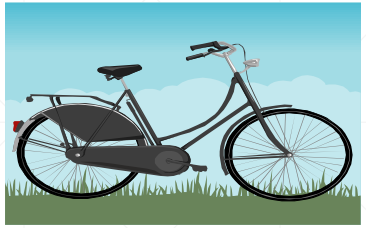
Bicicleta da Maria

Identidade→ Cada objeto tem a identidade inerente, ou seja, dois objetos distintos mesmo que todos os seus valores de atributos (como nome e tamanho) seja idênticos.

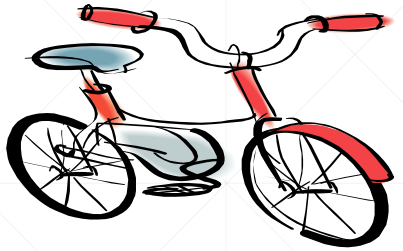
Diferentes Bicicletas – cada uma com sua identidade



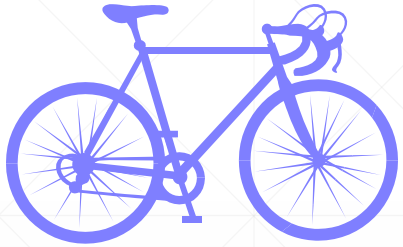
Bicicleta do casal
Roberta e Francisco



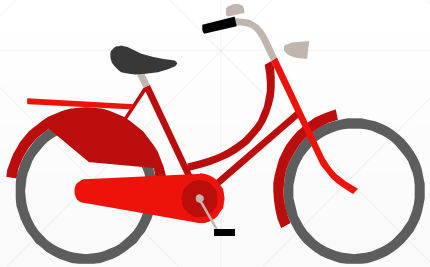
Bicicleta do João



Bicicleta do
Pedrinho



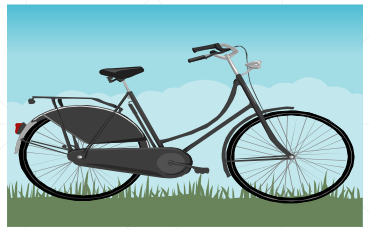
Bicicleta do atleta
José Ouro



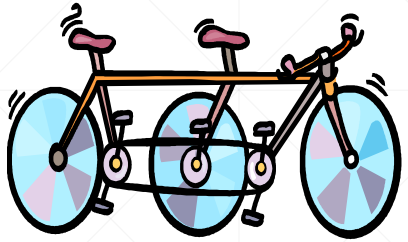
Bicicleta da Maria

Identidade→ No mundo real o objeto simplesmente existe. Mas em linguagem de programação, cada objeto possui uma referência única pela qual ele pode ser acessado (cada linguagem implementam de várias maneiras, endereços, índice de um vetor, ou número artificial)

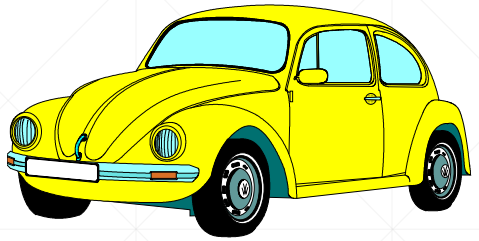
Diferentes Bicicletas e carros – cada uma com sua identidade



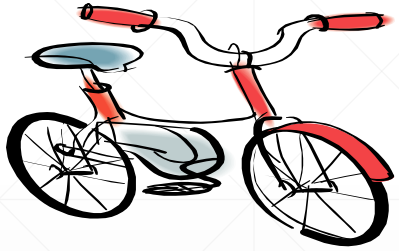
Bicicleta do João



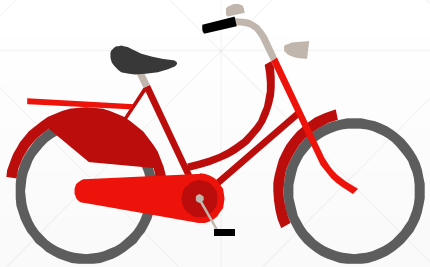
Bicicleta do casal
Roberta e Francisco



Carro do
José Silva



Bicicleta do
Pedrinho

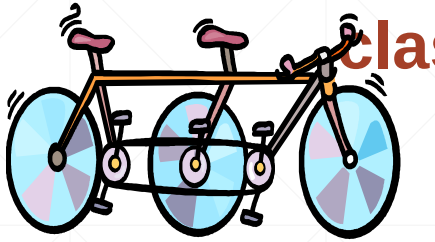


Bicicleta da Maria

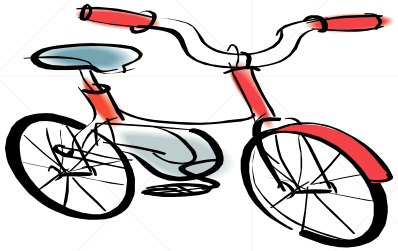
Carro do
João Oliveira

Identidade→ Essas referências dos objetos são uniformes e independentes do conteúdo dos objetos, permitindo a criação de coleções mistas de objetos

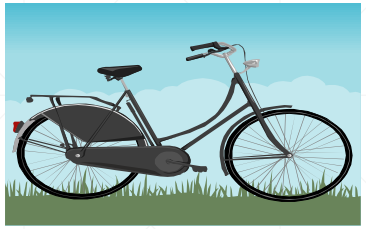
Diferentes Bicicletas – mesma classificação



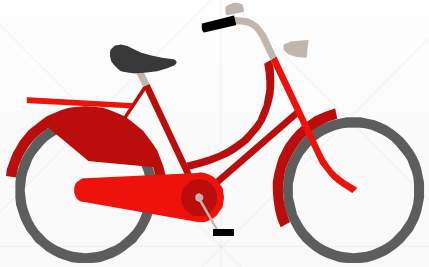
Bicicleta do casal
Roberta e Francisco



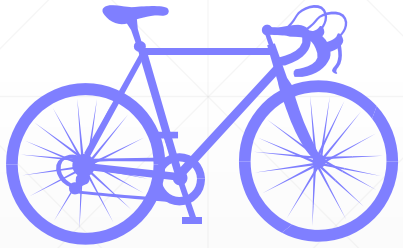
Bicicleta do
Pedrinho



Bicicleta do João



Bicicleta da Maria



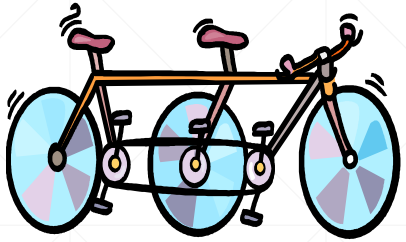
Bicicleta do atleta
José Ouro

Atributos
Tamanho do quadro
Tamanho da roda
Número de marchas
Material

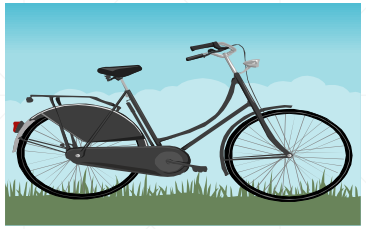
Operações
Muda de Marcha
Move
Freia

Classificação→ os objetos com a mesma estrutura de dados (atributos) e comportamento (operações) são agrupados em uma classe

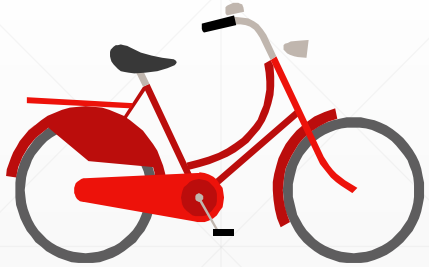
Diferentes Bicicletas – mesma classificação



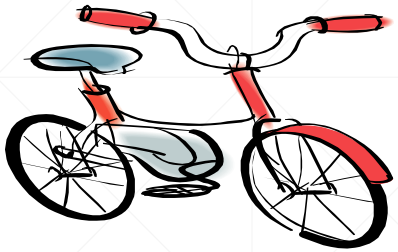
Bicicleta do casal
Roberta e Francisco



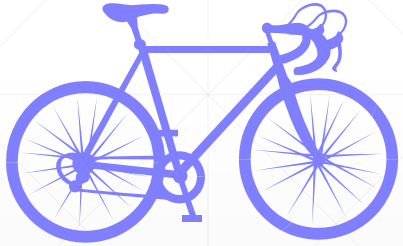
Bicicleta do João



Bicicleta da Maria



Bicicleta do
Pedrinho



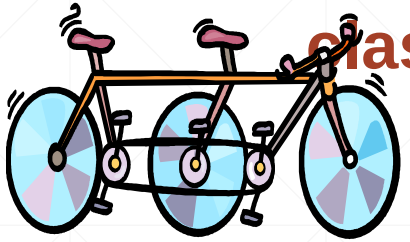
Bicicleta do atleta
José Ouro

Atributos
Tamanho do quadro
Tamanho da roda
Número de marchas
Material

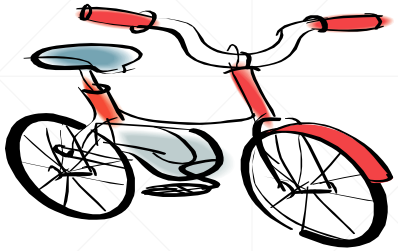
Operações
Muda de Marcha
Move
Freia

Classificação→ os objetos com a mesma estrutura de dados (atributos) e comportamento (operações) são agrupados em uma classe

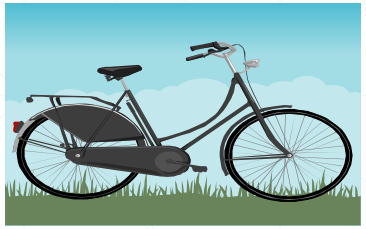
Diferentes Bicicletas – mesma classificação



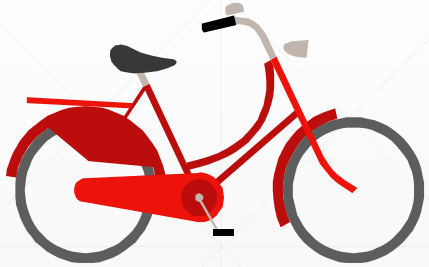
Bicicleta do casal
Roberta e Francisco



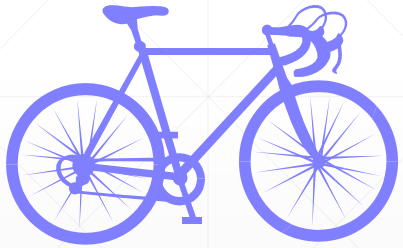
Bicicleta do
Pedrinho



Bicicleta do João



Bicicleta da Maria



Bicicleta do atleta
José Ouro

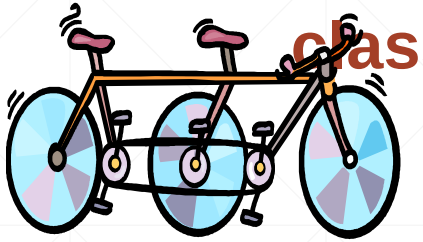
Abstrair para

Classe

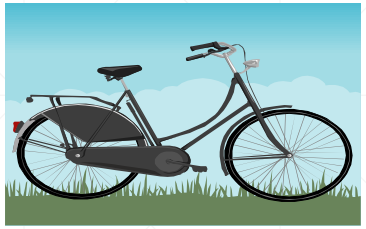
Bicicleta
-tamanhoQuadro : int -tamanhoRoda : int -nroMarcha : int -material : char
+mudaMacha() +move() +freia()

Classe → é uma abstração que descreve propriedades importantes para uma aplicação e ignora as demais

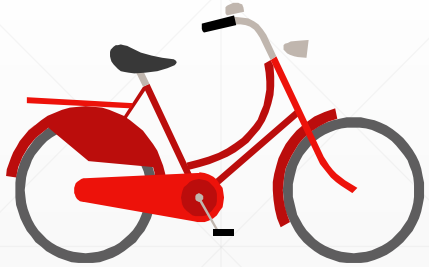
Diferentes Bicicletas – mesma classificação



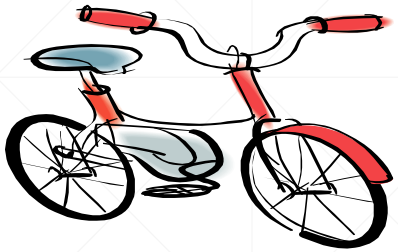
Bicicleta do casal
Roberta e Francisco



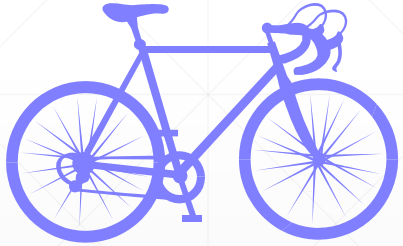
Bicicleta do João



Bicicleta da Maria



Bicicleta do
Pedrinho



Bicicleta do atleta
José Ouro

Abstrair
para

Classe

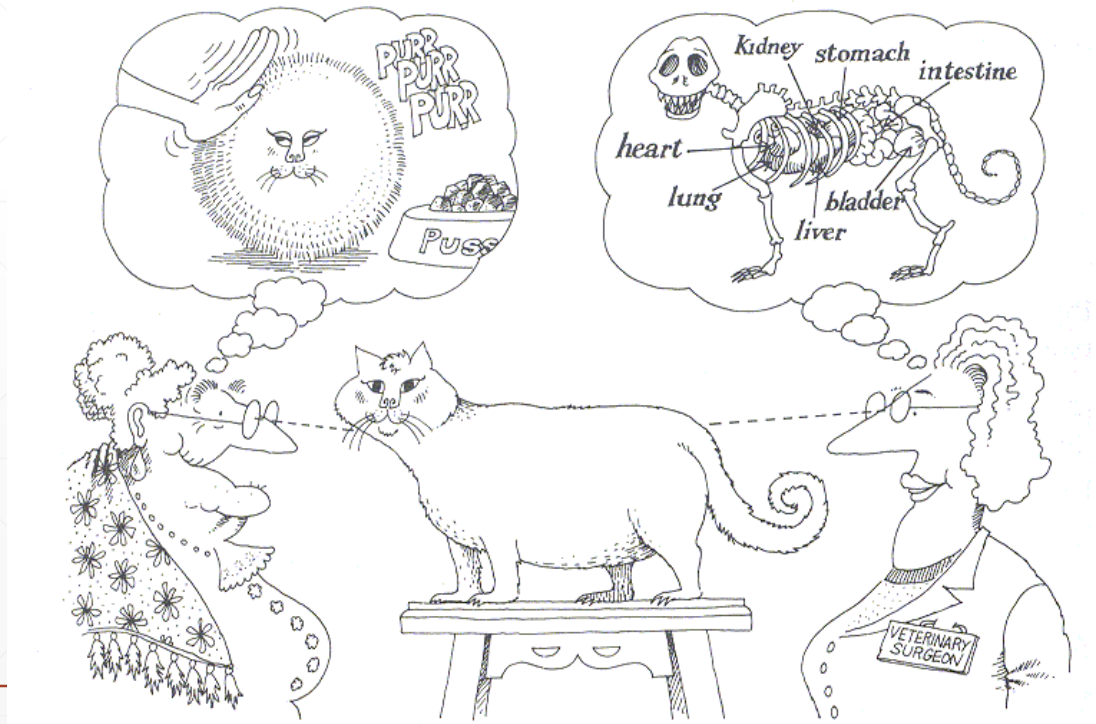
Bicicleta
-tamanhoQuadro : int -tamanhoRoda : int -nroMarcha : int -material : char
+mudaMacha() +move() +freia()

Classe → descreve um conjunto possivelmente infinito de objetos individuais.
Cada objeto é considerado uma instância de sua classe.

Abstração

- Permite que você se concentre nos aspectos essenciais de uma aplicação, ignorando os detalhes
  - Focalizar o que um objeto é e faz, antes de implementá-lo

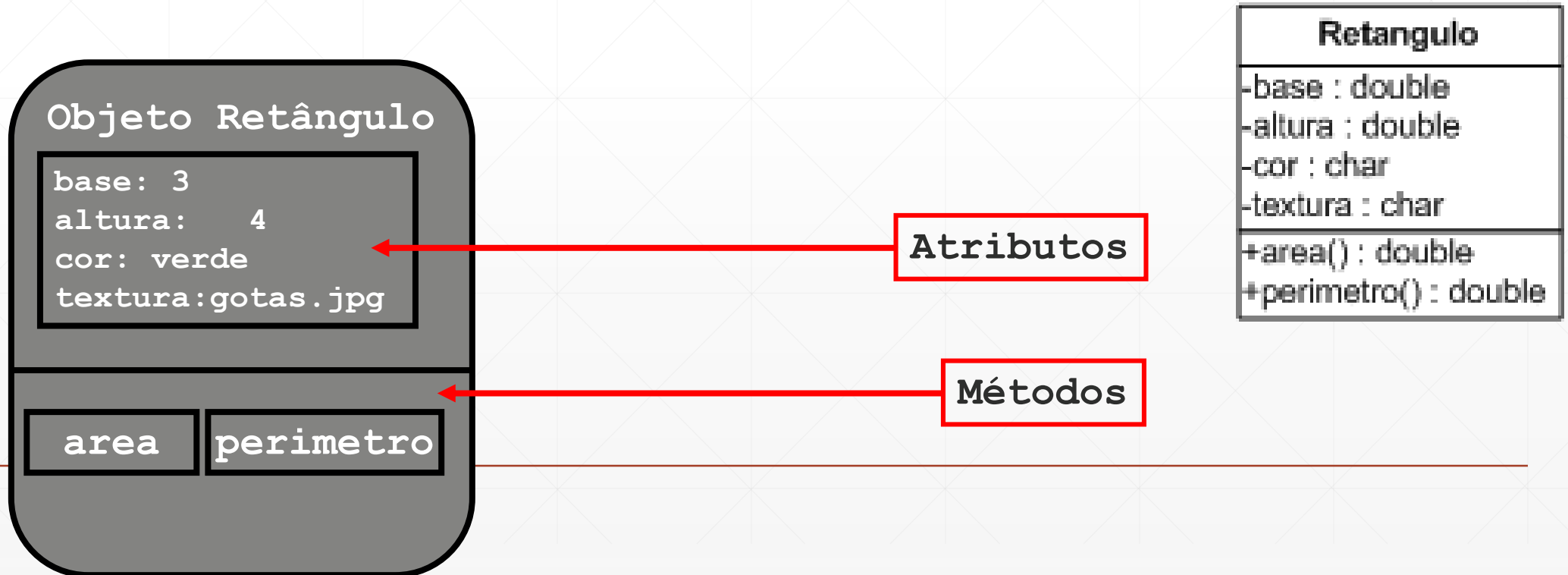
Abstração depende do observador



Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

Classes e objetos

- **Importante**: uma classe é uma ***abstração*** das características ***relevantes*** de um grupo de coisas do mundo real.
  - Na maioria das vezes, um grupo de objetos do mundo real é muito complexo para que *todas* suas características sejam representadas em uma classe.



Obrigado

Ana Claudia Rossi

ana.rossi@mackenzie.br