

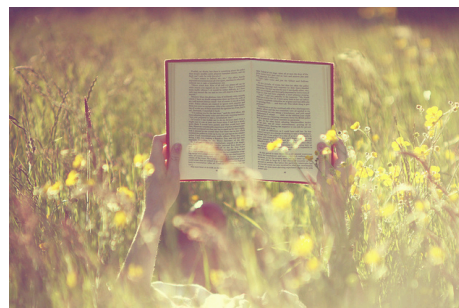
Revisão de Orientação a Objetos

Leticia Duboc
IME / DICC - UERJ

2º Semestre de 2015

Bibliografia

- Aula baseada em
 - Meilir Page-Jones. 2000. *Fundamentals of Object-Oriented Design in UML*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.



FlickrCCByBethan

Agenda

- Encapsulamento
- Ocultação de Informação/ Implementação
- Retenção de estado
- Identidade
- Mensagens
- Classes
- Herança
- Polimorfismo
- Coesão e Acoplamento



Quando entrei pela primeira vez no reino de OO, eu decidi resolver a definição de "Orientação a objetos" de uma vez por todas. Peguei uma dúzia de gurus do mundo orientado a objeto e os tranquei em um quarto sem comida ou água. Eu disse a eles que só estariam autorizados a sair depois que concordaram com uma definição que eu pudesse publicar para o desejoso mundo de software.

Seguiu uma hora de gritos e pancadas dentro do quarto, e então um silêncio. Temendo o pior, eu cuidadosamente abri a porta e olhei para uma visão potencialmente sangrenta. Os gurus estavam vivos, mas sentados cada um no seu canto e sem falar um do outro.

Aparentemente, cada guru começou a sessão, tentando estabelecer uma definição de orientação a objetos utilizando a prática cientificamente consagrada de repetir sua definição indefinidamente e cada vez mais alto. Quando isso não levou a nada, eles concordaram em listar as propriedades de um ambiente orientado a objeto que consideradas indispensáveis. Cada guru criou uma lista de cerca de seis a dez propriedades vitais.

Neste ponto, eles provavelmente tinham duas opções: Poderiam criar uma longa lista que era a união de suas listas individuais, ou, poderiam criar uma lista breve que era o cruzamento de suas listas. Eles escolheram a última opção e produziram uma pequena lista das propriedades que estavam em todas as listas individuais.

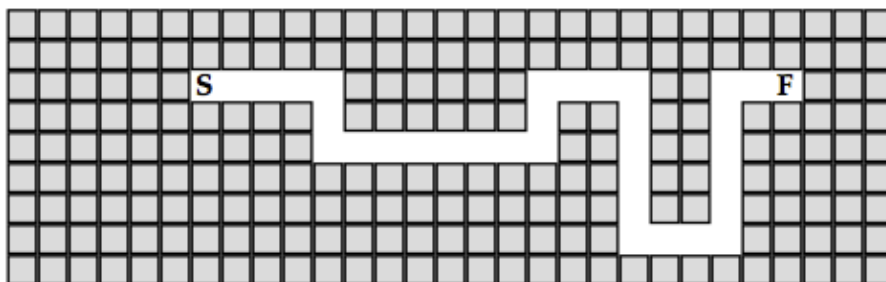
A lista era realmente muito curta: "encapsulamento".



Conceitos Básicos de OO

- Encapsulamento
- Ocultação de Informação/Implementação
- Retenção de estado
- Identidade
- Mensagens
- Classes
- Herança
- Polimorfismo
- Coesão e Acoplamento

Exemplo: Robo e Caminho



Interfaces

Robo

New: Robo
 turnLeft
 turnRight
 advance (n_squares):
 Boolean
 location: Square
 facingWall: Boolean
 display



Grid

New: Grid
 start: Square
 finish: Square
 insertRobot (r:Robot,
 location: Square):
 Boolean
 display



```

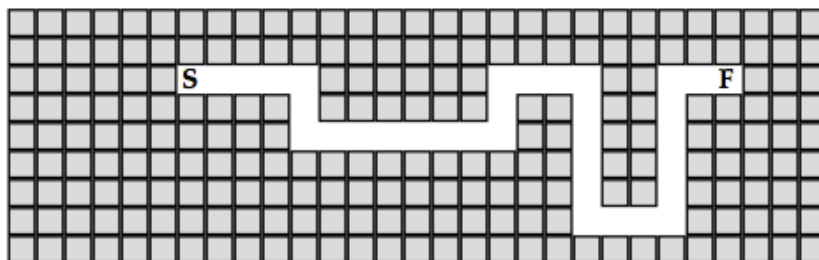
var grid: Grid := Grid.New;
var robot: Robot := Robot.New;
var insertOK: Boolean;
var advanceOK: Boolean;
var startSquare: Square;
const oneSquare = 1;
startSquare := grid.start;
insertOK := grid.insertRobot (robot, startSquare);
if not insertOK then
  abort everything !;
endif;

repeat 4 times max or until not robot.facingWall
  robot.turnLeft;
endrepeat;

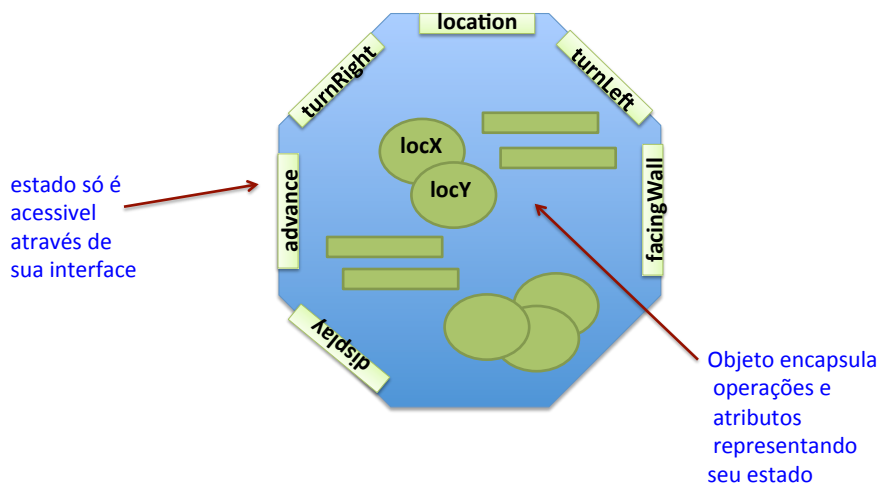
grid.display;
robot.display;

repeat until robot.location = grid.finish
  if robot.facingWall then robot.turnLeft;
  if robot.facingWall then
    robot.turnRight; robot.turnRight;
  endif;
endif;

advanceOK := robot.advance (oneSquare,);
robot.display;
endrepeat
  
```

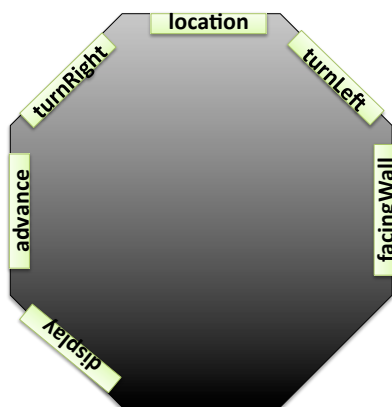


Encapsulamento

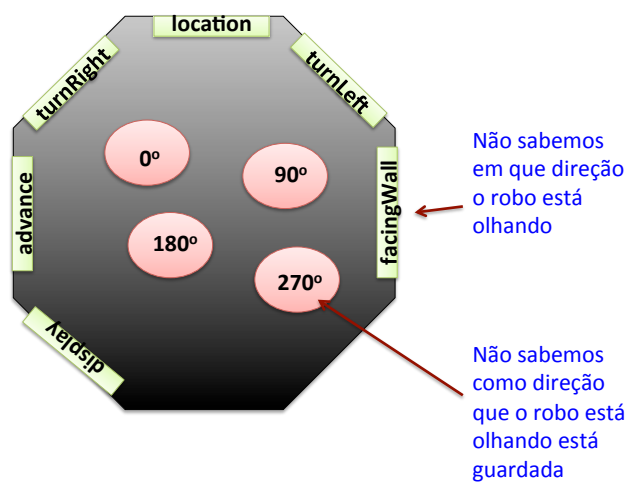


Ocultação de Informação/Implementação

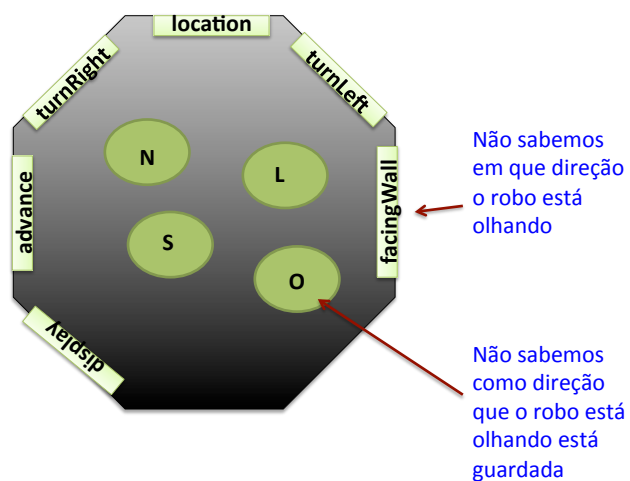
Restringe a visibilidade externa de certas informações ou decisões de implementação



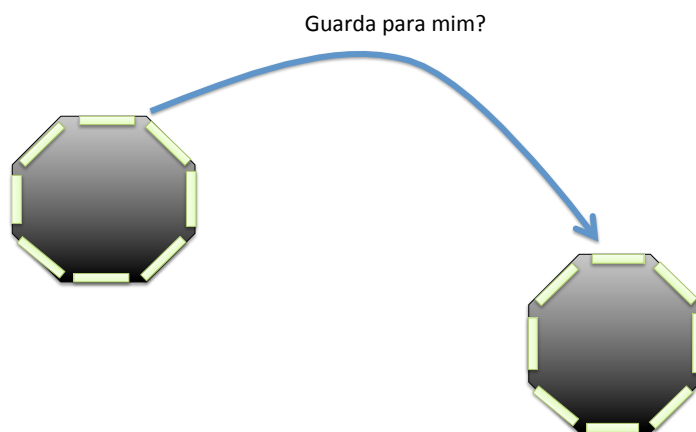
Ocultação de Informação/Implementação



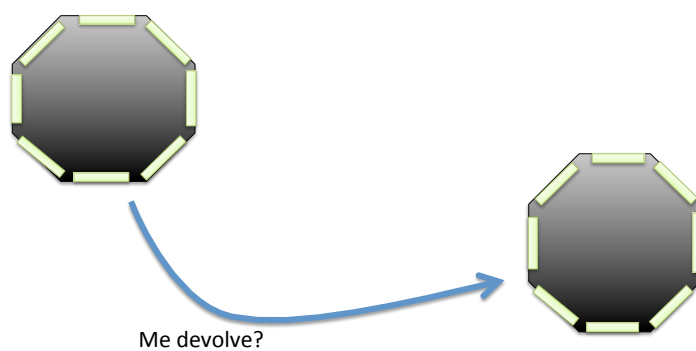
Ocultação de Informação/Implementação



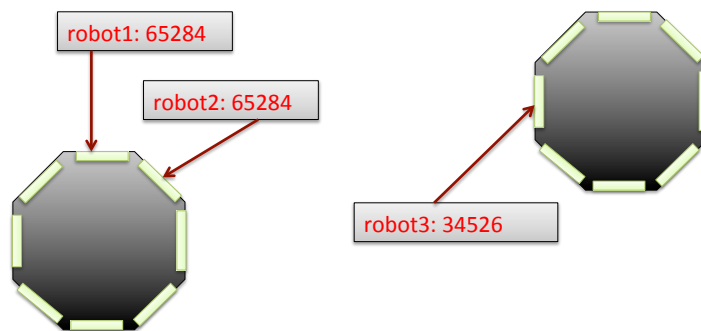
Retenção de Estado



Retenção de Estado

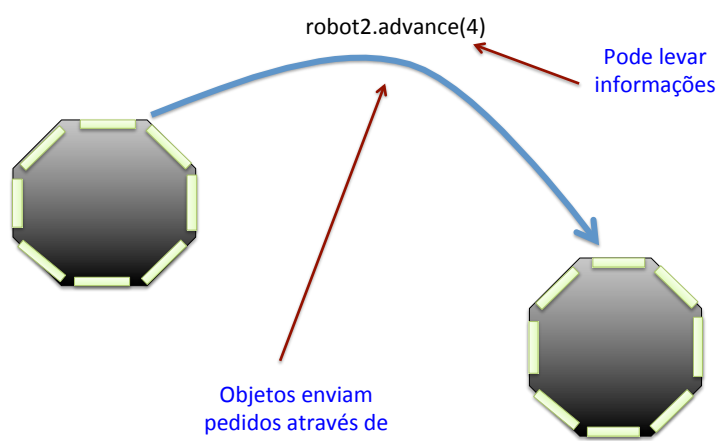


Identidade

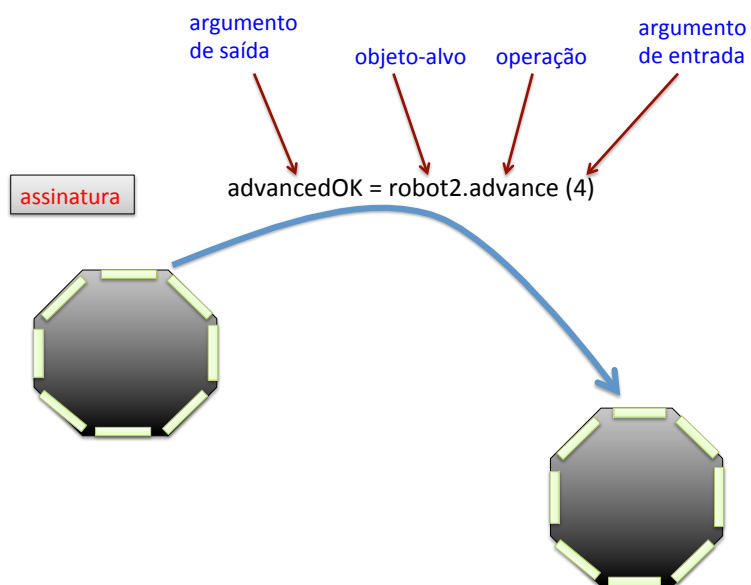


Objetos iguais em estrutura e informação
podem ser tratados como entidades distintas

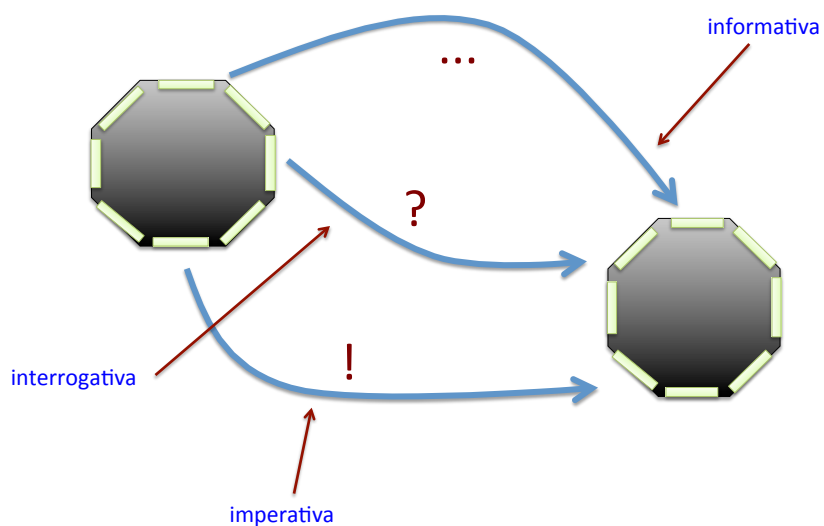
Mensagens

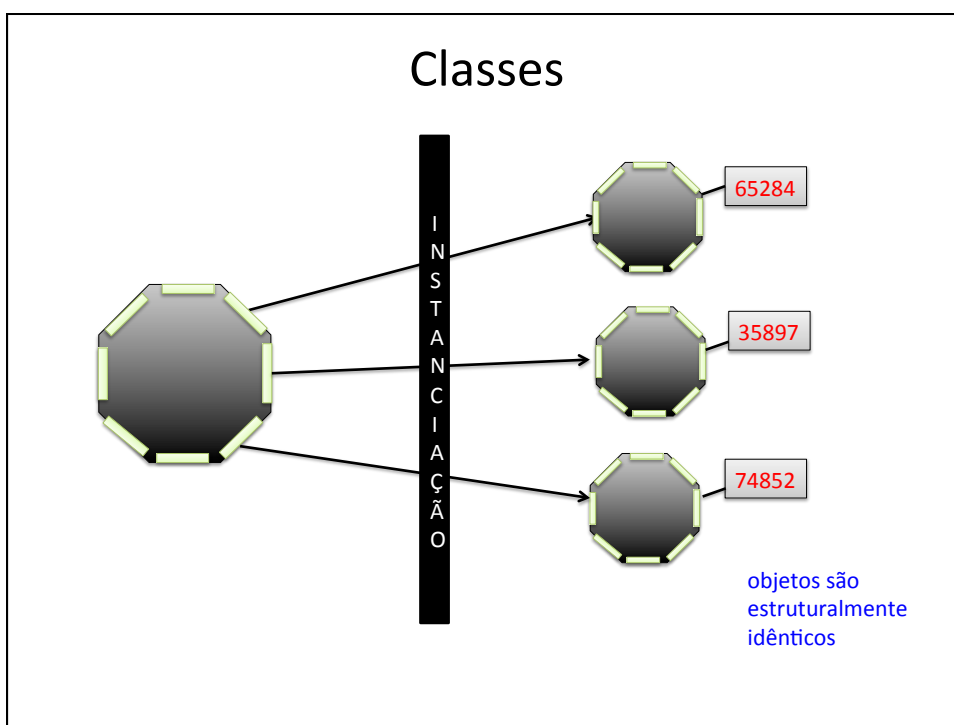


Estrutura da Mensagem

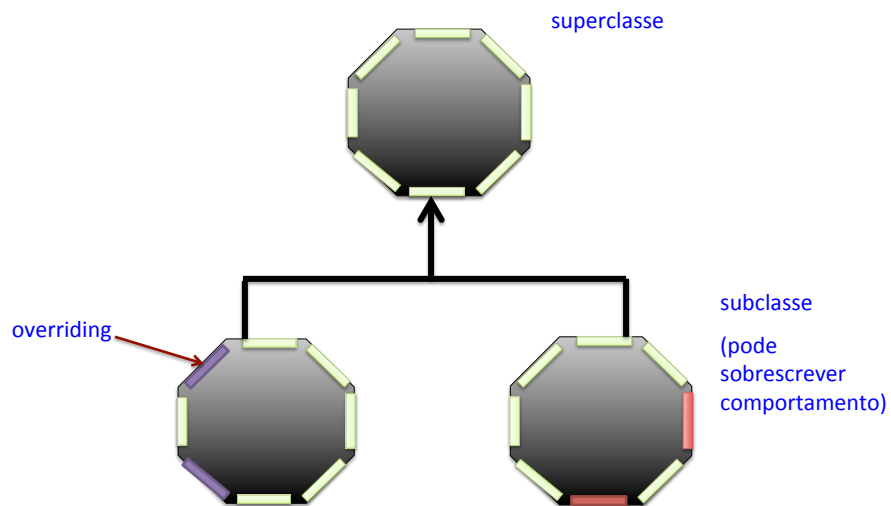


Tipos de Mensagens

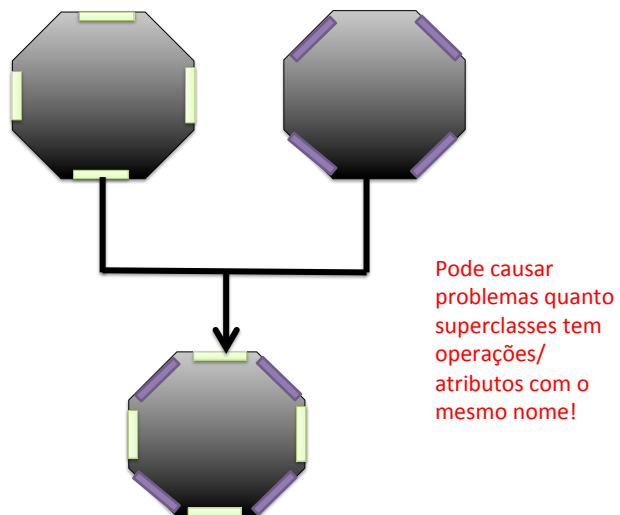




Herança

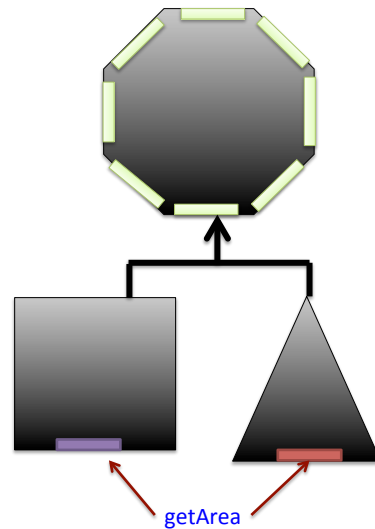


Herança Múltipla



Polimorfismo

```
var p: Polygon;  
var t : Triangle := Triangle.New;  
var s : Square := Square.New;  
...  
if user says OK  
then p := t  
else p := s  
endif;  
...  
p.getArea
```



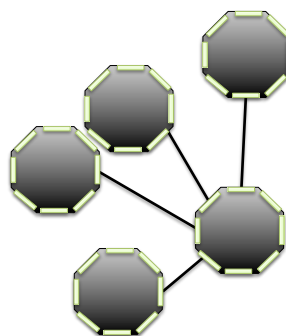
Coesão

- **Coesão**
 - Quanto as responsabilidades de um elemento estão fortemente relacionadas e focalizadas.
- **Problemas da baixa coesão**
 - são difíceis de compreender
 - são difíceis de reutilizar
 - são difíceis de manter
 - são delicadas; constantemente afetadas por modificações



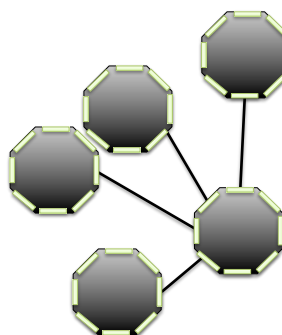
Acoplamento

- **Acoplamento**
 - Quão fortemente um elemento está conectado a, tem conhecimento de, ou depende de outros elementos.
- **Problemas do alto acoplamento**
 - Modificações locais forçadas decorrentes de modificações em classes relacionadas.
 - São mais difíceis de entender isoladamente.
 - São mais difíceis de reutilizar, pois seu uso requer a presença adicional de classes das quais é dependente.



Coesão e Acoplamento

- **Coesão**
 - operações e atributos devem estar relacionadas a esta responsabilidade
- **Acoplamento**
 - número e intensidade das conexões entre classes



Deve-se buscar alta coesão e baixo acoplamento

Interfaces

Robo

New: Robo
 turnLeft
 turnRight
 advance (n_squares):
 Boolean
 location: Square
 facingWall: Boolean
 display



Grid

New: Grid
 start: Square
 finish: Square
 insertRobot (r:Robot,
 location: Square):
 Boolean
 display



Revisando: Conceitos Básicos de OO

- Encapsulamento
- Ocultação de Informação/Implementação
- Retenção de estado
- Identidade
- Mensagens
- Classes
- Herança
- Polimorfismo
- Coesão e Acoplamento