

Programação Orientada a Objetos

Formalismos e Programação OO Objetos e Classes

André Santanchè e Oscar Rojas
Institute of Computing - UNICAMP
Março 2019

As Duas Faces da OO

- Abordagem de abstração
- Estrutura de dados

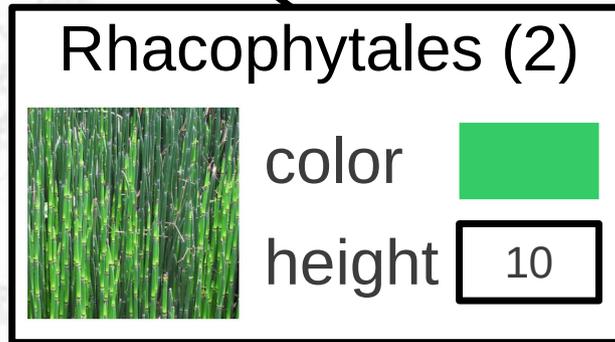
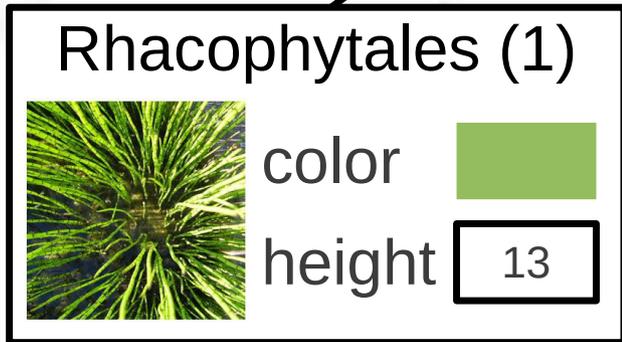
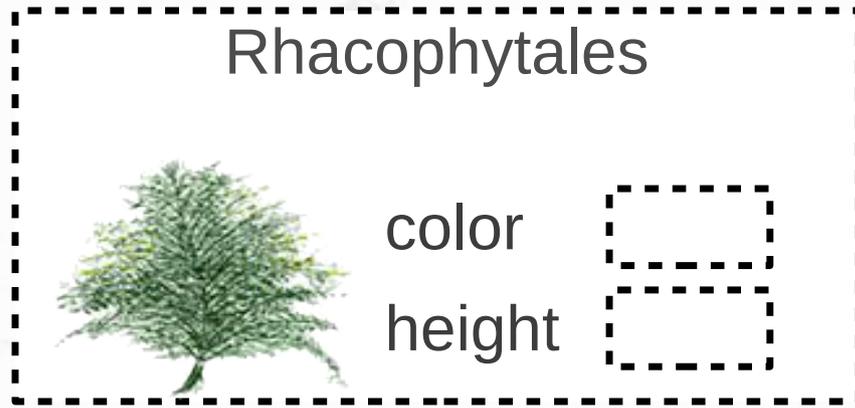
Formal Estereótipos / Classes

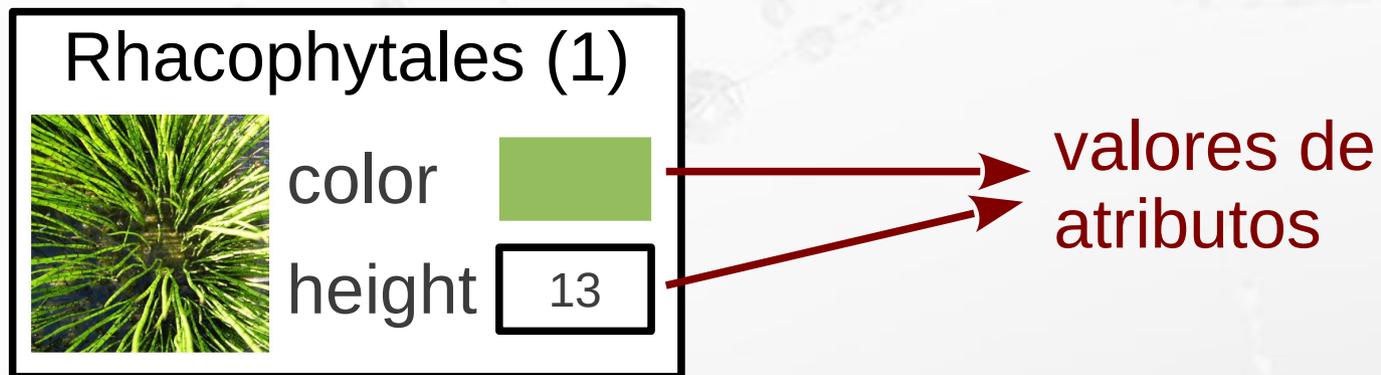
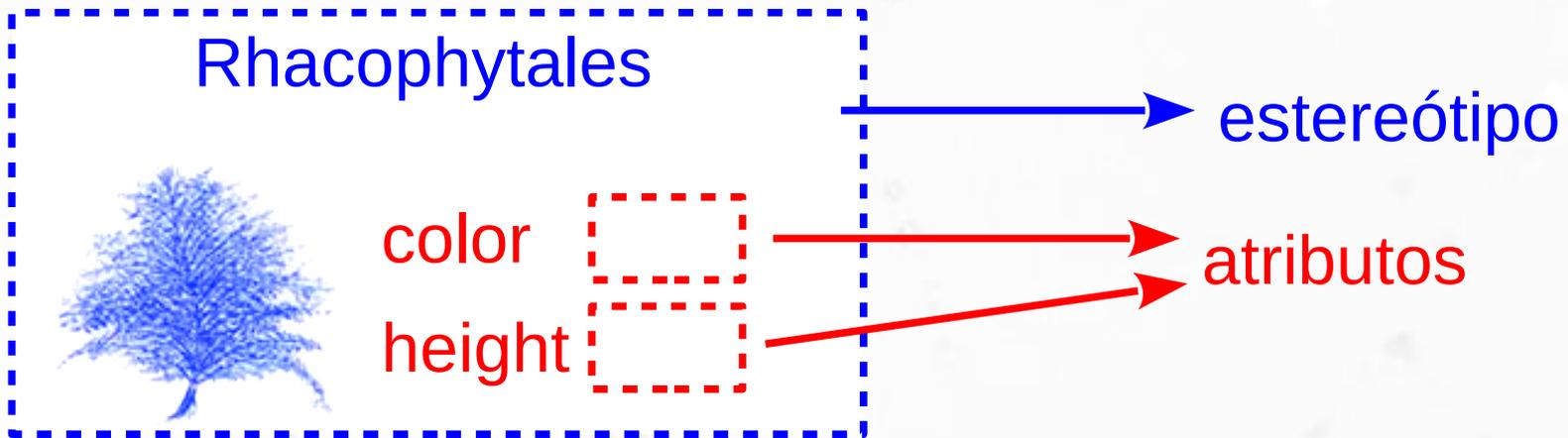
Modelo

Generalização

Instancias

Universo de Discurso







Objeto

● Objeto

“O que se apresenta à percepção com um caráter fixo e estável”. (Ferreira, 1989)

- Objetos são caracterizados por:
 - identidade;
 - atributos;
 - comportamento.

● Objeto

- Objeto do universo de discurso
- Identificável distintamente
- Existência independente



Pessoa



Livro

Departamento
de uma
Organização

Categoria
de um
Livo

Objeto



Pessoa



Livro

Departamento
de uma
Organização

Categoria
de um
Livro

Objetos são caracterizados por:

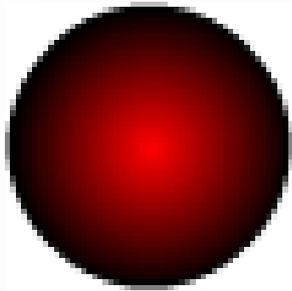
- identidade;
- atributos;
- comportamento.

Exemplo de Objeto

Esfera Vermelha

Objeto Esfera

Atributos (nome, valor)



(**peso**, 200 g)

(**raio**, 60 cm)

(**elasticidade**, alta)

(**cor**, vermelha)

Comportamento

aumentar, diminuir, se mover

Exemplo de Objeto

Um Financiamento

Objeto Financiamento

Atributos (nome, valor)



(valor, R\$ 150)

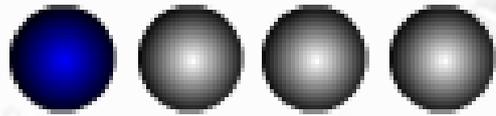
(número de parcelas, 3)

(percentual de juros, 1%)

Comportamento

calcula parcela

00: Classe



Classe

"Numa série ou num conjunto, grupo ou divisão que apresenta características ou atributos semelhantes." (Ferreira, 1989)

- Classificação de Carl Linné



Amphibia



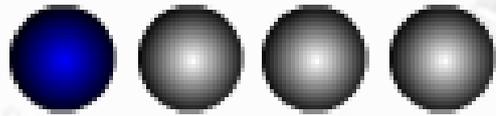
Reptilia



Aves

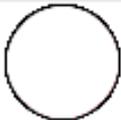


Mammalia



Classe

- Quando realizamos uma classificação de objetos, identificamos o seu comportamento e as características que eles possuem em comum.
- Classes definem:
 - Atributos que irão descrever o objeto;
 - Métodos que definem o comportamento dos mesmos.

Classe	Objeto	Objeto	Objeto
 peso raio cor	 peso: 200 g raio: 60 cm cor: vermelha	 peso: 200 g raio: 60 cm cor: azul	 peso: 50 g raio: 30 cm cor: verde

Objetos e Classes

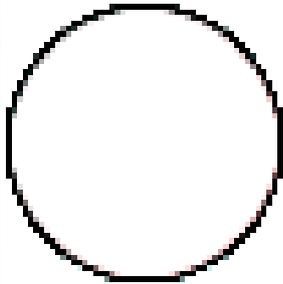
- Os objetos são organizados/divididos em grupos chamados classes.
- Objetos da mesma classe têm:
 - o mesmo conjunto de atributos (os valores dos atributos podem ser diferentes);
 - o mesmo conjunto de métodos.

Exemplo de Classe

Esfera

Classe Esfera

Atributos (nome, tipo)



(**peso**, real)

(**raio**, real)

(**elasticidade**, string)

(**cor**, color)

Comportamento

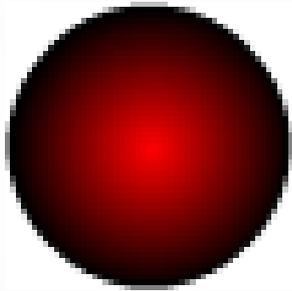
aumentar, diminuir, se mover

Exemplo de Objeto

Esfera Vermelha

Objeto Esfera

Atributos (nome, valor)



(**peso**, 200 g)

(**raio**, 60 cm)

(**elasticidade**, alta)

(**cor**, vermelha)

Comportamento

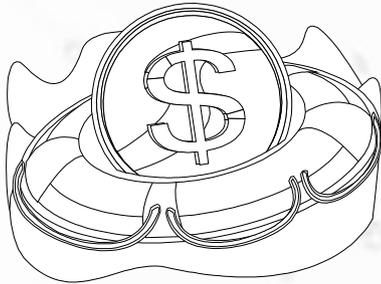
aumentar, diminuir, se mover

Exemplo de Classe

Financiamento

Classe Financiamento

Atributos (nome, tipo)



(**valor**, real)

(**número de parcelas**, inteiro)

(**percentual de juros**, real)

Comportamento

calcula parcela

Exemplo de Objeto

Um Financiamento

Objeto Financiamento

Atributos (nome, valor)



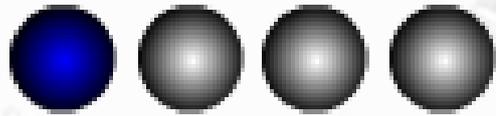
(valor, R\$ 150)

(número de parcelas, 3)

(percentual de juros, 1%)

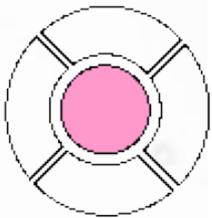
Comportamento

calcula parcela

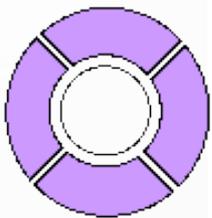


Classe

- Em Programação Orientada ao Objeto:



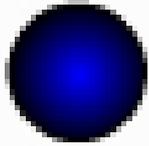
Atributos: dados que pertencem a cada instância da classe (objeto); são definidos sob a forma de variáveis.



Métodos: definem o comportamento do objeto; representados por módulos.

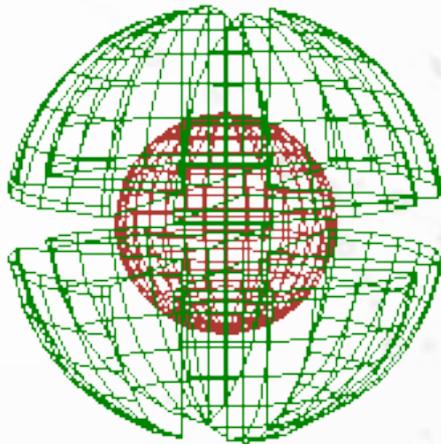
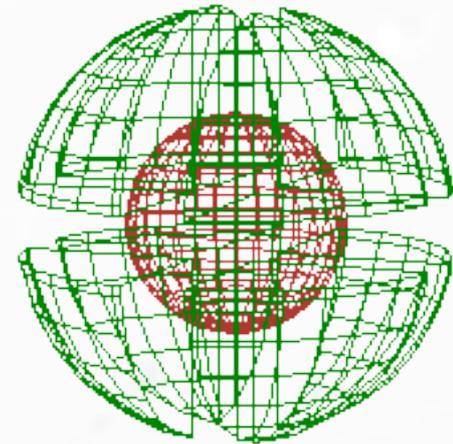
Objeto instância de Classe

- Um Objeto consiste em uma instância de uma Classe
- A instância define:
 - identidade única
 - estado (representado pelos valores de seus atributos).



Objeto

A classe pode ser importada de uma biblioteca ou definida pelo programador.



Para se instanciar um objeto utiliza-se o método **Construtor**.



Construtores e Destrutores

- **Construtor (mesmo nome da classe)**
 - Todo o objeto deve ser instanciado (criado) através da ativação do método construtor.
- **Destrutor (finalize)**
 - O destrutor é o inverso do construtor, ele é ativado automaticamente quando o objeto está sendo destruído a fim de liberar a memória ocupada pelo mesmo.
- ***Garbage Collection* (Coleta de Lixo)**
 - O mecanismo de gerência automática de memória que destrói o objeto quando ele não está mais sendo usado.

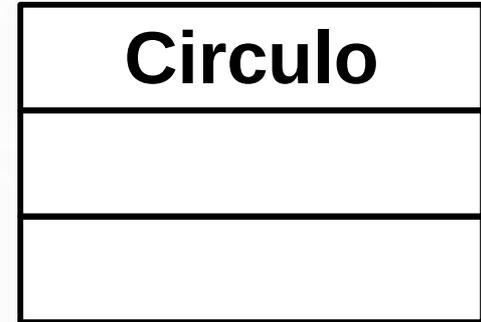
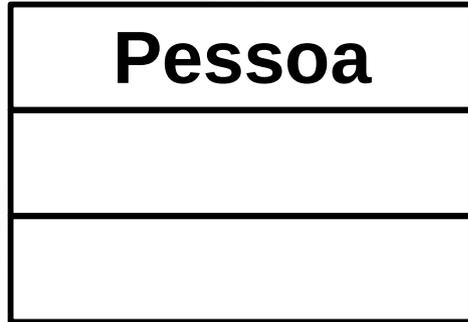
UML

Unified Modeling Language

- <http://www.uml.org/>
- Desenvolvida entre 1994-96
- Criadores
 - Grady Booch, Ivar Jacobson and James Rumbaugh na Rational Software
- Padrão OMG em 1997
 - OMG - Object Management Group
 - <http://omg.org/>

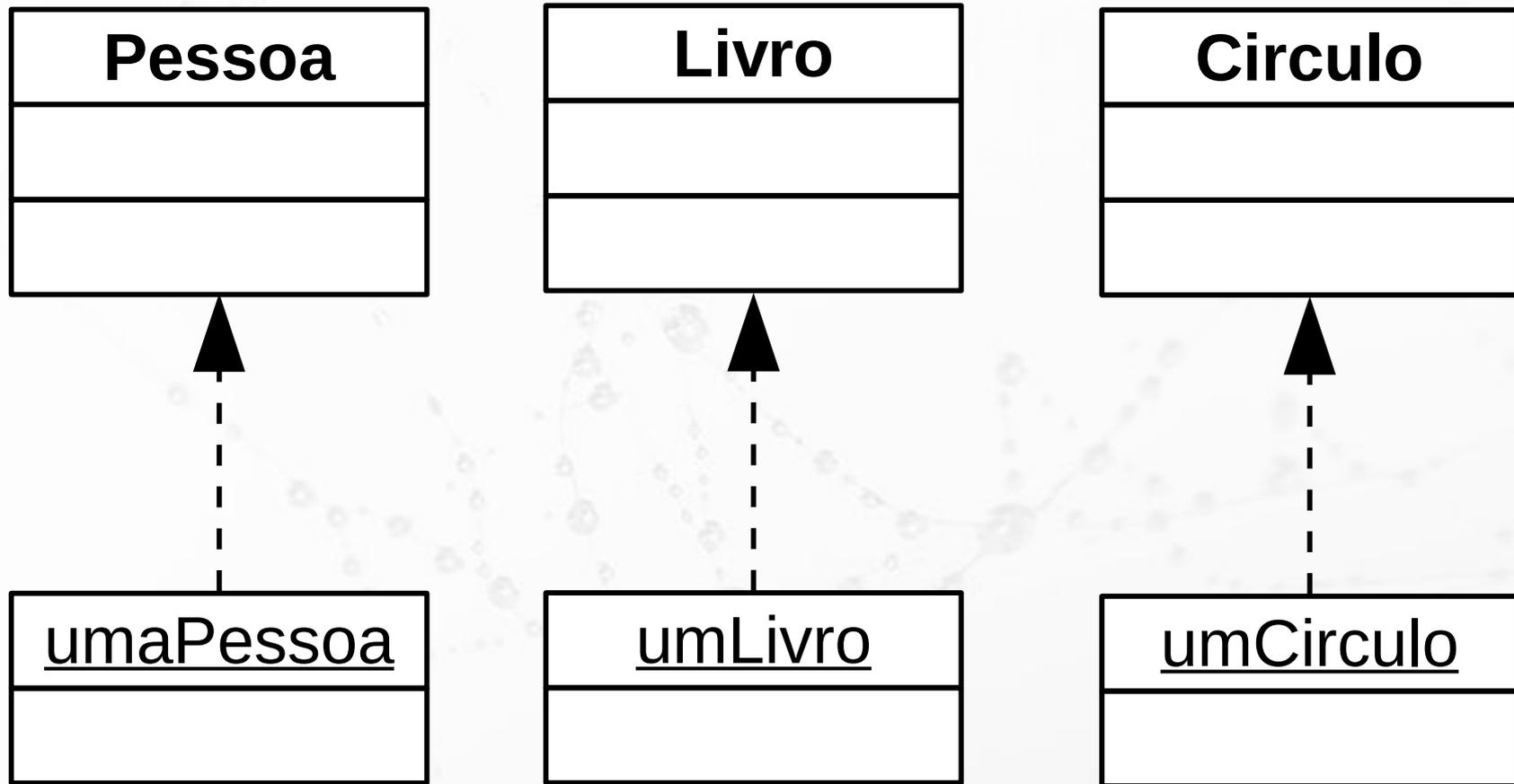
(Wikipedia, 2015)

UML Classe



UML

Instância de Classe



UML

Instância de Classe (alternativa)

umaPessoa: Pessoa

umaLivro: Livro

umCirculo: Circulo

Atributos

- Cada instância da classe (objeto) define valores para atributos que a descrevem



Pessoa

codigo: String
nome: String
telefone: int

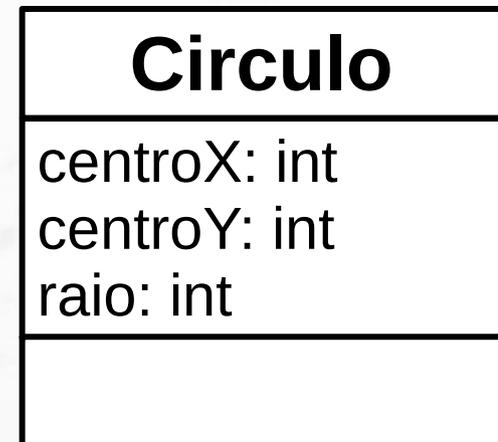
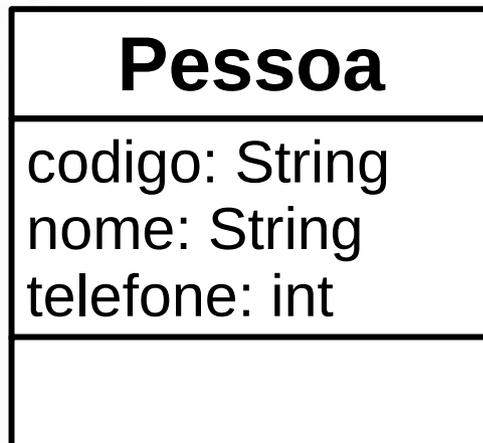


Livro

isbn: String
titulo: String
autor: String
ano: int
categoria: String

UML

Atributos (propriedades)



Classe Circulo - Java

Atributos

```
package pt.c0200.s02classe.s01circulo01;

public class Circulo {
    int centroX, centroY;
    int raio;
}
```

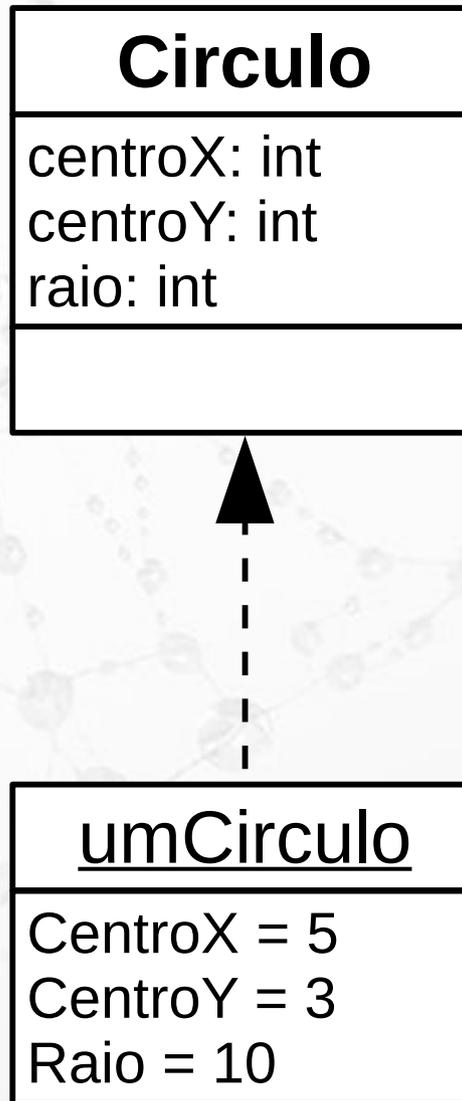
```
package pt.c0200.s02classe.s01circulo01;

public class AppCirculo01 {
    public static void main(String args[]) {
        Circulo circ = new Circulo();
        circ.centroX = 5;
        circ.centroY = 3;
        circ.raio = 10;

        System.out.println("Circulo: centro(" + circ.centroX + ", " +
            circ.centroY + "), raio " + circ.raio);
    }
}
```

UML

Instância com valores de atributos

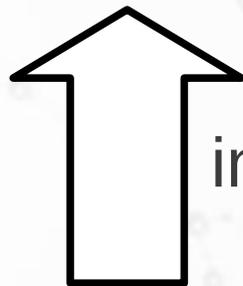
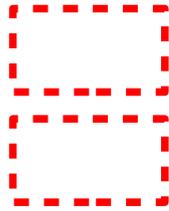


Rhacophytales



color

height



instância

Rhacophytales (1)



color

height



Rhacophytales

color: ColorType

height: int



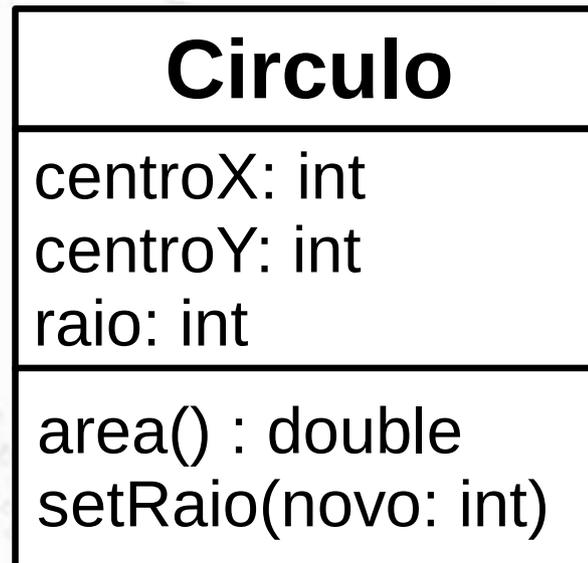
R1

color = green

height = 13

UML

Métodos (operações)



Classe Circulo - Java

Métodos

```
package pt.c0200.s02classe.s01circulo05;
public class Circulo {
    int centroX, centroY;
    int raio;
    Circulo(int centroX, int centroY, int raio) {
        this.centroX = centroX;
        this.centroY = centroY;
        this.raio = raio;
    }

    double area() {
        return Math.PI * raio * raio;
    }
}
```

```
package pt.c0200.s02classe.s01circulo05;
public class AppCirculo05 {
    public static void main(String args[]) {
        Circulo circ = new Circulo(5, 3, 10);
        System.out.println("Circulo: centro(" + circ.centroX + ", " +
circ.centroY + "), raio " + circ.raio);
        System.out.println("Area do circulo: " + circ.area());
    }
}
```

Compilando Duas Classes

```
santanche@tapejara: ~/Documents/java/pt/c02classe/s01circulo05
File Edit View Search Terminal Help
santanche@tapejara:~$ cd Documents/java/pt/c02classe/s01circulo05/
santanche@tapejara:~/Documents/java/pt/c02classe/s01circulo05$ ls
AppCirculo05.java Circulo.java
santanche@tapejara:~/Documents/java/pt/c02classe/s01circulo05$ javac *.java
santanche@tapejara:~/Documents/java/pt/c02classe/s01circulo05$ ls
AppCirculo05.class AppCirculo05.java Circulo.class Circulo.java
santanche@tapejara:~/Documents/java/pt/c02classe/s01circulo05$
```

Execução de Classes Interligadas

- Classes no mesmo pacote são automaticamente interligadas



Objeto em Java



- A instanciação do objeto se dá através do comando **new**.
- Quando o objeto é instanciado é acionado um método especial denominado construtor que tem o mesmo nome da classe.

Auto-referência e `this`

- Sem o `this`:

```
Circulo(int pCentroX, int pCentroY, int pRaio) {  
    centroX = pCentroX;  
    centroY = pCentroY;  
    raio = pRaio;  
}
```

- Com o `this`

```
Circulo(int centroX, int centroY, int raio) {  
    this.centroX = centroX;  
    this.centroY = centroY;  
    this.raio = raio;  
}
```

Auto-referência e `this`

- Para realizar referência a si próprio o objeto pode usar a referência **`this`**

```
public class Circulo {  
    int centroX, centroY;  
    int raio;  
  
    Circulo(int pCentroX, int pCentroY, int pRaio) {  
        centroX = pCentroX;  
        centroY = pCentroY;  
        raio = pRaio;  
    }  
}
```

Estudo de Caso

Bastião

o*o

*
o*o

*
o*o

Atributos

idade (1 a 3 anos)
estado (acordado, dormindo)

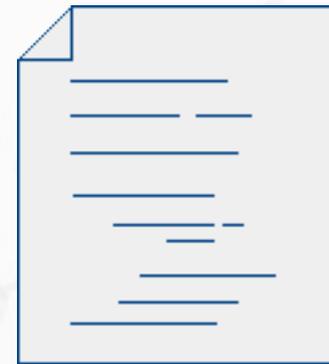
Métodos

aparecer, crescer, dormir e acordar

Estudo de Caso



Classe Bastiao no arquivo
Bastiao.java



Classe Principal no arquivo
Principal.java

Instanciação

Declaração da Referência

```
Bastiao theBastian;
```

Instanciação do Objeto (chamada do construtor)

```
theBastian = new Bastiao();
```

Chamada de Método

```
theBastian.aparece();
```

Atributos Estáticos

- Recebem o prefixo `static`
- Atributos que pertencem à classe
 - mesmo valor do atributo compartilhado por todos os objetos

Métodos Estáticos

- Recebem o prefixo `static`
- Métodos de classe
 - não exigem instância para serem acionados
 - só podem acessar os atributos estáticos
 - só podem acionar outros métodos estáticos

Objetos da Biblioteca Java

Vetor

- Objeto especial do Java

- Declaração

`<tipo>[] <declaração1>, ..., <declaraçãon>;`

`<tipo> <declaração1>[], ..., <declaraçãon>[];`

- `<declaração>`

- Sintaxe: `<nome> = <inicialização>`
- Chaves são usadas para inicializar cada dimensão
- Ex.: `int primos[] = {1, 2, 3, 5, 7};`

Vetor

- A inicialização inline instancia automaticamente um objeto
- Quando a inicialização não é inline o vetor ou matriz precisa ser instanciado com new

```
<nome> = new <tipo>[<tamanho>]
```

▫ Ex.:

```
int primos[];  
primos = new int[5];
```

Estruturas de Dados Dinâmicas

Vector e ArrayList

- vetores dinâmicos
- Vector - sincronizado
 - mais seguro, mais lento
- ArrayList - não sincronizado
 - menos seguro, mais rápido
- Métodos
 - size()
 - add(<elemento>)
 - get(<posição>) → <elemento>

Estruturas de Dados Dinâmicas

Hashtable

- Tabela hash
- Métodos:
 - `put(<chave>, <valor>)`
 - `get(<chave>) → <valor>`

Estruturas de Dados Dinâmicas

Stack

- Pilha
- Métodos:
 - `push(<elemento>)`
 - `pop() → <elemento>`

Referências Bibliográficas

- Almeida, Charles Ornelas , Guerra, Israel; Ziviani, Nivio (2010) **Projeto de Algoritmos** (transparências aula).
- Bloom, Paul (2007) **Introduction to Psychology** - transcrição das aulas (aula 17). Yale University.
- Ferreira, Aurélio B. H. (1989) **Minidicionário da Língua Portuguesa**. Rio de Janeiro, Editora Nova Fronteira.
- Houaiss, Instituto Antônio. **Dicionário Houaiss da língua portuguesa** (2006) Editora Objetiva, Março.
- IBM - International Business Machines Corporation. **IBM Smalltalk Tutorial** [Online] <http://www.wi2.uni-erlangen.de/sw/smalltalk/>
- Liskov, Barbara; Zilles, Stephen. **Programming with abstract data types** (1974) ACM SIGPLAN Notices, 9 (4) p. 50.

Referências Bibliográficas

- Meyer, Bertrand (1997) **Object-Oriented Software Construction - Second Edition**. USA, Prentice-Hall, Inc.
- Miller, Robert (2004) **6.831 User Interface Design and Implementation (lecture notes)**. MIT OpenCourseware.
- Rocha, Heloisa Vieira da, Baranauskas, Maria Cecilia Calani (2003) **Design e Avaliação de Interfaces Humano-Computador**. NIED/UNICAMP.
- Santos, L. R., & Hood, B. M. (2009). **Object representation as a central issue in cognitive science**. The Origins of Object Knowledge: The Yale Symposium on the Origins of Object & Number Representation. Oxford: Oxford University Press.
- Shaw, M. **Abstraction Techniques in Modern Programming Languages** (1984) IEEE Software, 1, 4, 10-26.

Referências Bibliográficas

- Tenenbaum, Aaron M.; Langsam, Yedidyah; Augenstein, Moshe J. **Data Structures Using C** (1990) Prentice Hall, Upper Saddle River, NJ.



André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Imagens Externas

- Havang(nl) [http://commons.wikimedia.org/wiki/User:Havang%28Gld,_NL%29.jpg]
url (ver licença específica):
http://commons.wikimedia.org/wiki/File:Bomenpark_Meijhorst,_Nijmegen_%28Gld,_NL%29.jpg



- Eric Gaba [<http://commons.wikimedia.org/wiki/User:Sting>]
url (ver licença específica):
http://commons.wikimedia.org/wiki/File:Easter_Island_map



- Kharker [<http://en.wikipedia.org/wiki/User:Kharker>]
url (ver licença específica):
http://commons.wikimedia.org/wiki/File:Ardf_map.png

