
Programação Orientada a Objetos

Objetos, Classes, Atributos, Java e UML

Profa.: Márcia Sampaio Lima

EST - UEA

Referência: Slides Prof. Flávio José Mendes Coelho

Objetos

É uma abstração de uma coisa do mundo real que reúne características e comportamentos.

Objetos

Objeto = Características + Comportamentos

- **Características**

- descrevem, identificam ou particularizam o objeto (não são partes do objeto).

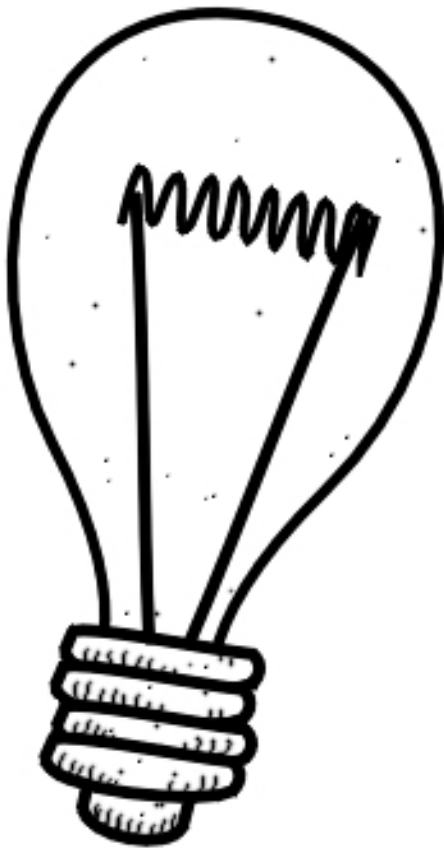
- **Comportamento**

- uma ação que um objeto sabe realizar (uma habilidade do objeto).

Objetos: Atributos e Operações

- No mundo OO (orientado a objetos):
 - características →
 - **Atributos**
 - ou campos
 - ou variáveis de instância
 - comportamentos →
 - Operações
 - ou **métodos** de instância

Objetos: Atributos e Operações



**Uma lâmpada
branca de
60W**

Objetos: Atributos e Operações

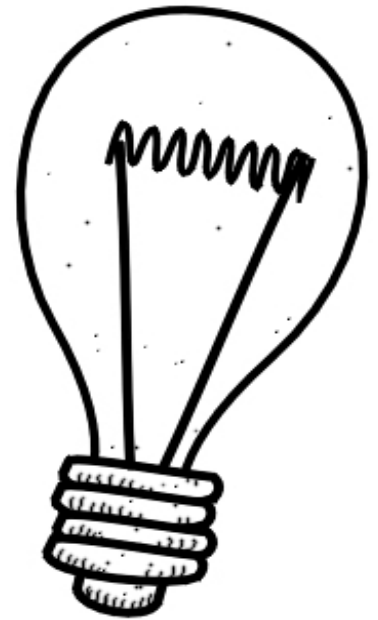
■ atributos:

- ❑ voltagem: 60W
- ❑ situação: ligada (ou desligada, ou queimada).
- ❑ cor: branca
- ❑ durabilidade: 1000 "usos".

■ operações:

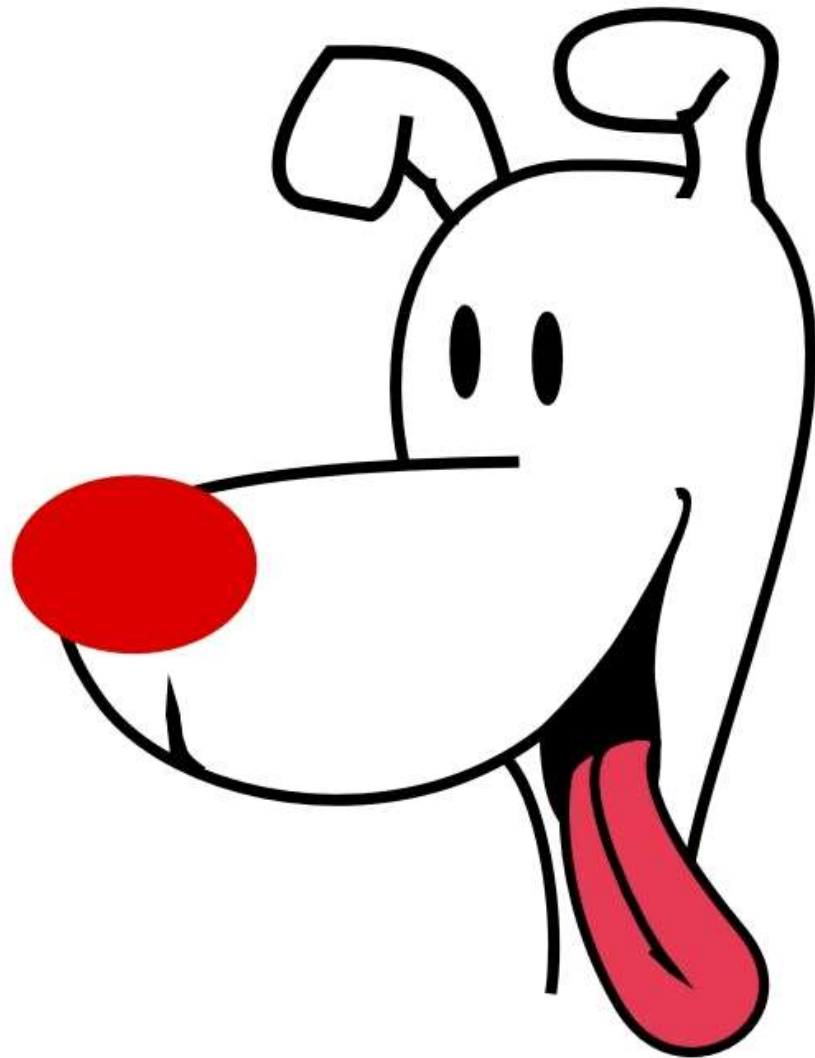
- ❑ desligar
- ❑ ligar

Uma lâmpada
branca de 60W



Objetos: Atributos e Operações

**Um cão chamado
Darth Vader**



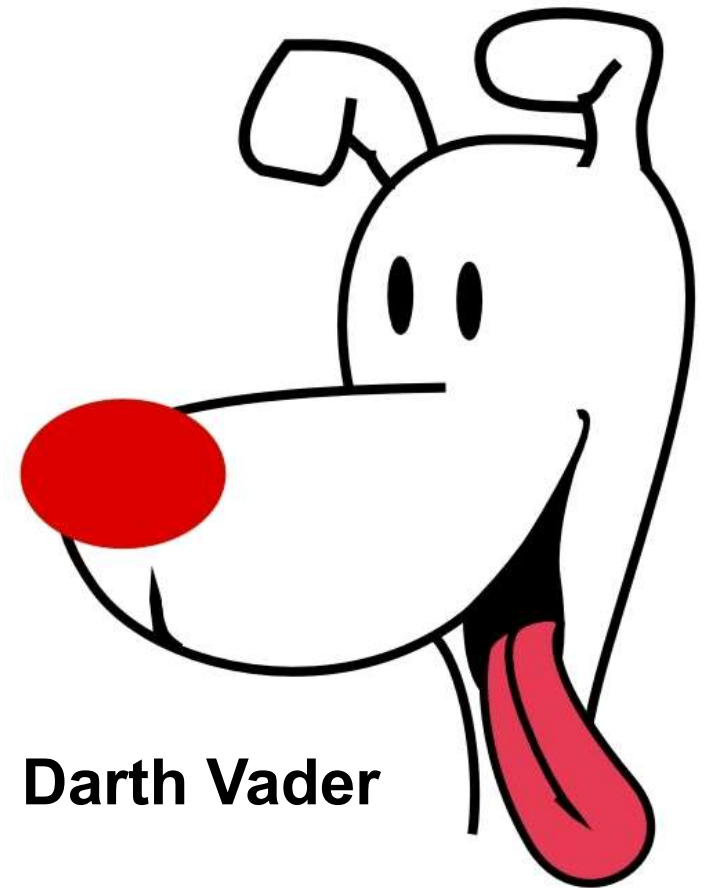
Objetos: Atributos e Operações

■ atributos:

- ❑ nome: Darth Vader
- ❑ cor: branca
- ❑ raça: vira-lata
- ❑ fome: muita (pouca, nenhuma)

■ operações:

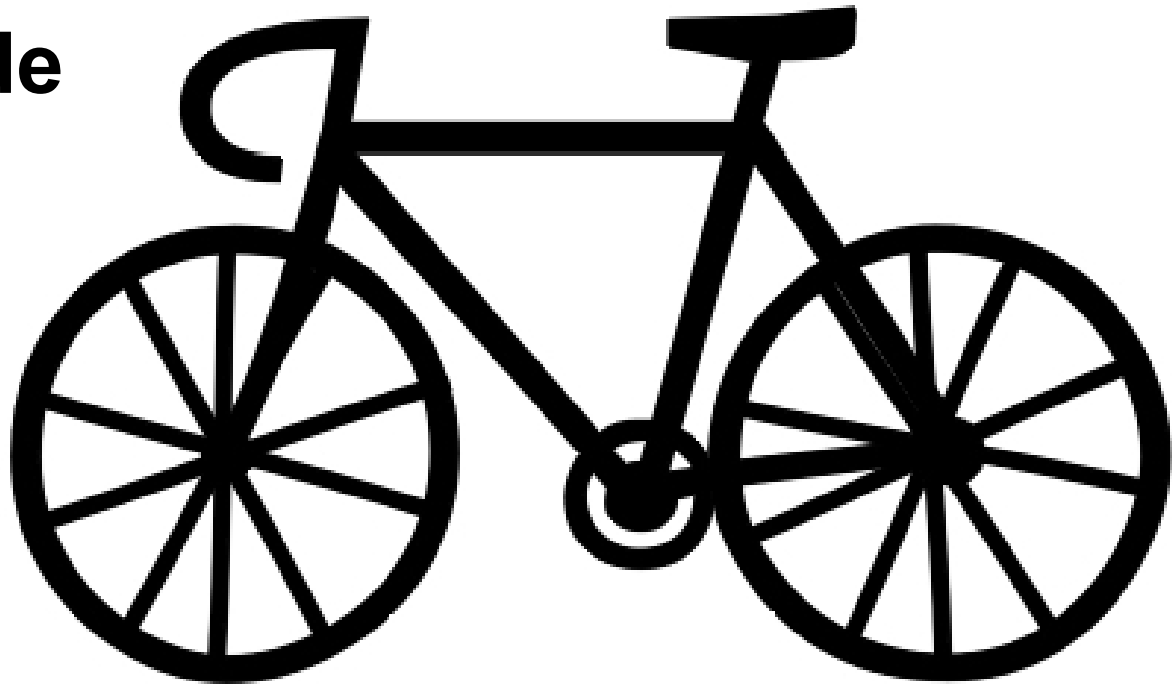
- ❑ latir, comer, correr
- ❑ sentar, dormir
- ❑ balançar o rabo



Darth Vader

Objetos: Atributos e Operações

**Minha
bicicleta de
corrida**



Objetos: Atributos e Operações

■ atributos:

- ❑ cor: Azul
- ❑ marca: Caloi
- ❑ Quantidade de marchas: 20
- ❑ Marcha corrente: 5

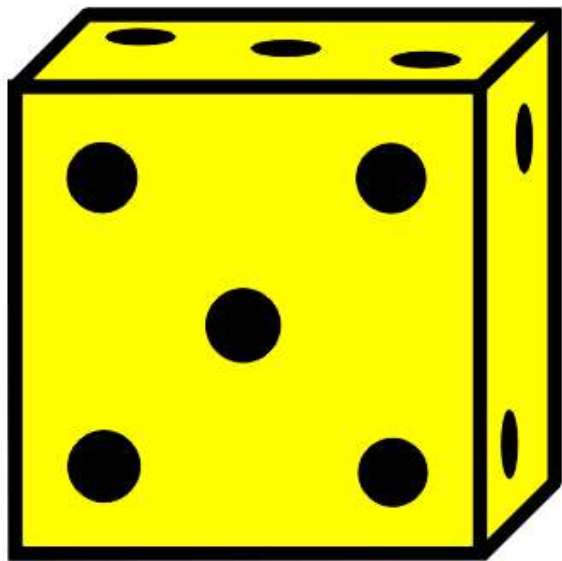
■ operações:

- ❑ frear, pedalar, soltar pedal
- ❑ aumentar marcha
- ❑ reduzir marcha



Minha bicicleta de corrida

Objetos: Atributos e Operações



Um dado

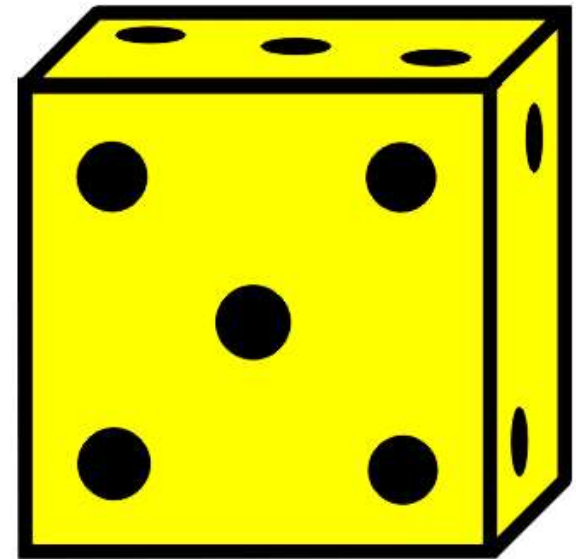
Objetos: Atributos e Operações

■ atributos:

- ❑ cor: amarelo
- ❑ valor da face superior: 3

■ operações:

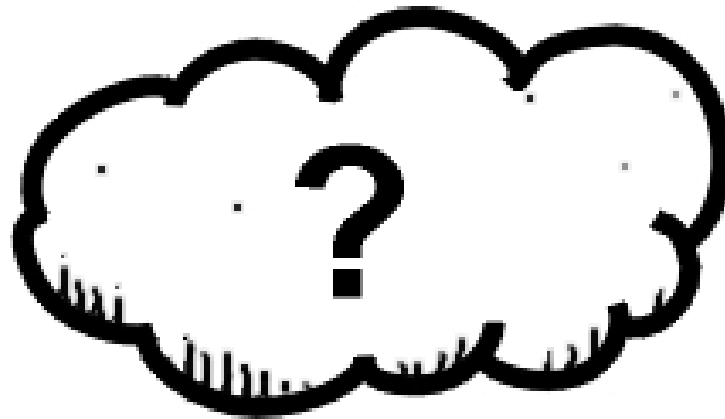
- ❑ sortear um número de 1 a 6



Um dado

Objetos: Atributos e Operações

**Minha conta
bancária**



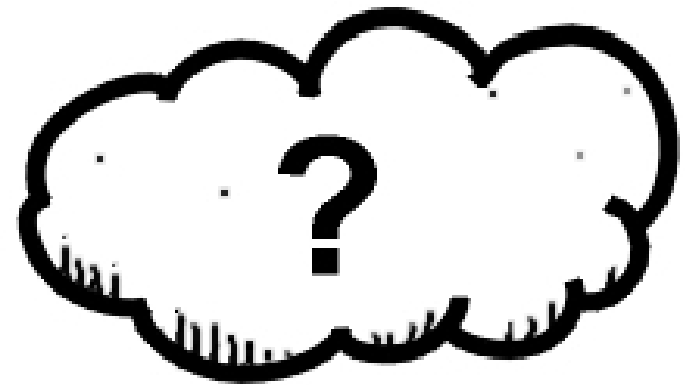
Objetos: Atributos e Operações

■ atributos:

- ❑ número da conta: 223344
- ❑ número da agência: 5577
- ❑ saldo: R\$-100,00
- ❑ tipo: corrente (poupança)

■ operações:

- ❑ sacar
- ❑ depositar
- ❑ emitir saldo
- ❑ emitir extrato



Minha conta bancária

Objetos: Estado de um Objeto

- **Estado de um objeto** é o conjunto de valores de seus atributos em um dado momento.
- Parte das operações de um objeto, em geral, pode alterar o estado do objeto.
- Conforme as operações do objeto vão sendo invocadas, o objeto pode assumir diversos estados durante sua vida.

Classes

Referência: Slides Prof. Flávio José Mendes Coelho

Classes

■ Como agrupar estes objetos?

Ford-Ranger Fiat-Palio Tom
 Piano M.Ravel
Garfield J.S.Bach Volkswagen-Fox
 Violino Violão
Chevrolet-Vectra Frajola
Trompete L.V.Beethoven Oboé
 W.A.Mozart

Classes

- Que nomes atribuir para os agrupamentos?

Ford-Ranger
Fiat-Palio
Volkswagen-Fox
Chevrolet-Vectra

Garfield
Tom
Frajola

J.S.Bach
L.V.Beethoven
W.A.Mozart
M.Ravel

Trompete Violão
Piano Violino Oboé

Classes

Carro

Ford-Ranger
Fiat-Palio
Volkswagen-Fox
Chevrolet-Vectra

Gato

Garfield
Tom
Frajola

Compositor

J.S.Bach
L.V.Beethoven
W.A.Mozart
M.Ravel

Instrumento Musical

Trompete Violão
Piano Violino Oboé

Classes

Carro

Ford-Ranger
Fiat-Palio
Volkswagen-Fox
Chevrolet-Vectra

Classes

Carro



Ford-Ranger

Atributos:

no. do chassi=128877732
velocidade atual=70km/h
kilometragem=50.000km
nível de combustível=87%
potência do motor=3.0
quantidade de portas=2

Operações:

ligar, desligar, acelerar, frear, desacelerar, avançar
marcha, reduzir marcha, anular marcha, girar volante...

Classes

Carro



Fiat-Palio

Atributos:

no. do chassi=355589001
velocidade atual=40km/h
kilometragem=23.000km
nível de combustível=40%
potência do motor=1.0
quantidade de portas=5

Operações:

ligar, desligar, acelerar, frear, desacelerar, avançar
marcha, reduzir marcha, anular marcha, girar volante...

Classes

Carro



Volkswagen-Fox

Atributos:

no. do chassi=11110078655

velocidade atual=80km/h

kilometragem=1.045km

nível de combustível=97%

potência do motor=1.8

quantidade de portas=5

Operações:

ligar, desligar, acelerar, frear, desacelerar, avançar
marcha, reduzir marcha, anular marcha, girar volante...

Classes

Carro



Chevrolet-Vectra

Atributos:

no. do chassi=1056988371192

velocidade atual=60km/h

kilometragem=23.331km

nível de combustível=32%

potência do motor=2.0

quantidade de portas=5

Operações:

ligar, desligar, acelerar, frear, desacelerar, avançar
marcha, reduzir marcha, anular marcha, girar volante...

Classes

- Quando temos vários objetos de um mesmo tipo, dizemos que eles pertencem a uma categoria ou classe. Exemplo:
 - ❑ Ford-Ranger, Fiat-Palio e Chevrolet-Vectra pertencem à classe dos **carros**.
 - ❑ Tom, Garfield e Frajola pertencem à classe dos **gatos**.
 - ❑ Bach, Ravel, e Mozart são compositores, logo, pertencem à classe dos **compositores**.

Classes

- **Classe** é uma abstração que descreve os atributos e as operações de um conjunto homogêneo de objetos.
- Isto significa que:
 - ❑ todos os objetos são do mesmo "tipo".
 - ❑ todos os objetos possuem os mesmos atributos e as mesmas operações.
 - ❑ os estados dos objetos podem ser distintos entre si.
 - ❑ cada objeto da classe possui uma identidade única (mesmo que seus estados sejam iguais).

Classes: Representações

- Como descrever uma classe, ou seja, descrever os atributos e as operações de seus objetos?
 - Algumas alternativas são:
 - ✓ Usar a língua portuguesa
 - ✓ Usar uma linguagem de programação OO
 - ✓ Usar uma linguagem de modelagem

Classes: Representações

1. Usar a língua portuguesa:
 - Exemplo: uma classe **Gato** descreveria os atributos dos objetos gato, assim:
 - um gato tem como atributos um nome, uma idade e um peso.
 - um gato deve ter as seguintes "habilidades" (operações): correr, comer e fazer aniversário.
 - problema: como descrever as ações de correr, comer e fazer aniversário?

Classes: Representações

1. Usar a língua portuguesa:

☐ Vantagem:

- é fácil porque nos expressamos naturalmente nesta linguagem.

☐ Desvantagem:

- um computador não a entende.
- é ambígua, imprecisa e muito extensa.

Classes: Representações

2. Usar uma linguagem de programação OO:

☐ Vantagem:

- um computador a entende.
- é precisa, relativamente pequena e sem ambigüidades.

☐ Desvantagem:

- expressar-se nela (programar) não é trivial.
- é carregada de detalhes.

Classes: Representações

3. Usar uma linguagem de modelagem:

☐ Vantagem:

- um computador pode entendê-la.
- é relativamente precisa, pequena e sem ambigüidades.
- sua representação gráfica aumenta a compreensão e a visão do todo.

☐ Desvantagem:

- despreza detalhes de implementação.

Classes: Representações

- Para fins de desenvolvimento de software as opções 2 e 3 são as melhores.
 - Linguagem de programação: Java.
 - Linguagem de modelagem: UML.

Representando tudo em Java e UML

Referência: Slides Prof. Flávio José Mendes Coelho

Classes: Representação em UML

- **UML – Linguagem Unificada de Modelagem** (*Unified Modeling Language*).
- UML nasceu em torno de 1995 e é fruto dos trabalhos de Booch, Jacobson e Rumbaugh.
- Ela fornece representações gráficas que auxiliam a modelagem e a programação OO.

Classes: Representação em UML



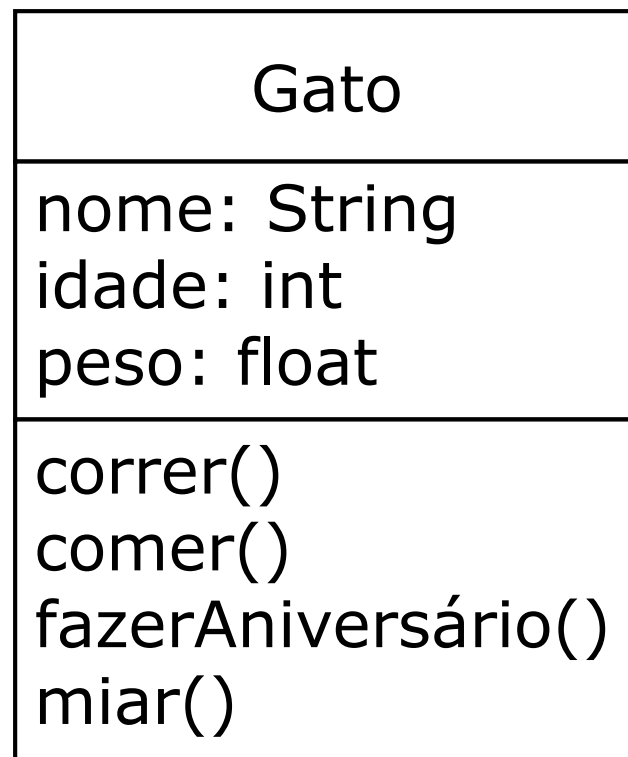
←..... Seção de nome
da classe

←..... Seção de atributos

←..... Seção de operações

←..... Ei! Mas UML não é tão colorido, assim!

Classes: Representação em UML

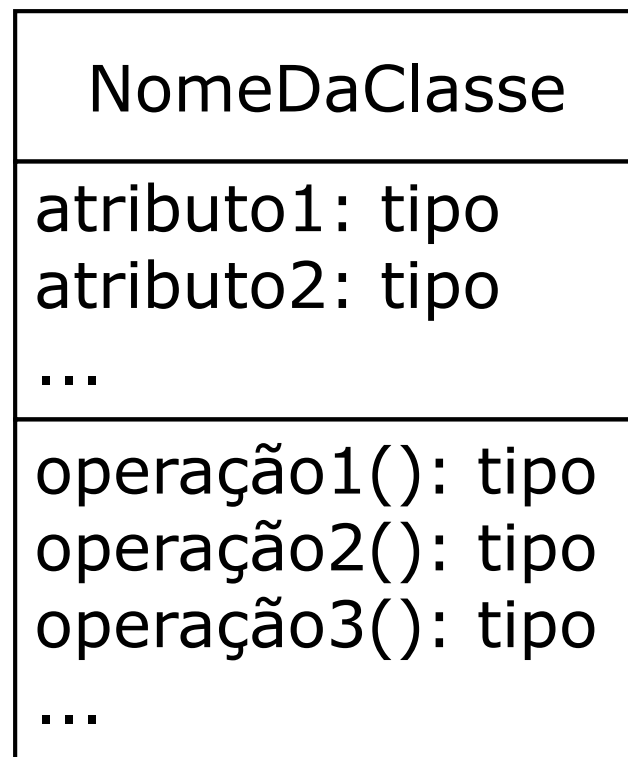


Seção de nome da classe

Seção de atributos

Seção de operações

Classes: Representação em UML

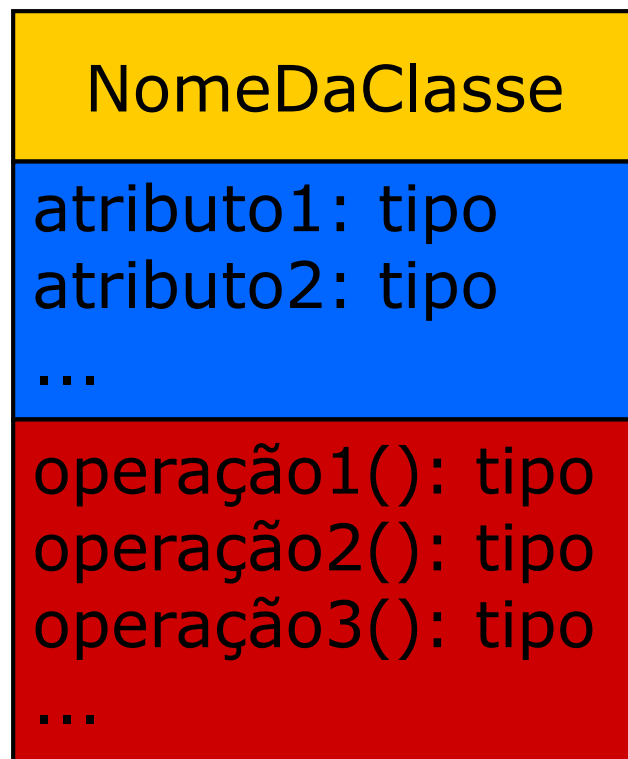


←..... Seção de nome
da classe

←..... Seção de atributos

←..... Seção de operações

Classes: Representação em Java



```
public class NomeDaClasse {  
    tipo atributo1;  
    tipo atributo2;  
    ...  
    tipo método1() {  
        // lógica...  
    }  
    tipo método2() {  
        // lógica...  
    }  
    ...  
}
```

Classes: Representação em Java

- Um método é a implementação de uma operação.

Gato

nome: String
idade: int
peso: float

correr()
comer()
fazerAniversário()
miar()

```
void fazerAniversario() {  
    System.out.println  
        ("Parabéns!!");  
    idade++;  
}
```

Classes: Representação em Java

Arquivo: Gato.java

```
public class Gato {  
    String nome;  
    int idade;  
    float peso;  
  
    void correr() {  
        System.out.println(nome + " está correndo!");  
        peso--;  
        System.out.println("Peso: " + peso);  
    }  
    void comer() {  
        System.out.println(nome + " está comendo!");  
        peso = peso + 2;  
        System.out.println("Peso: " + peso);  
    }  
}
```

Classes: Representação em Java

Arquivo: Gato.java

```
public class Gato {  
    String nome;  
    int idade;  
    float peso;  
  
    void fazerAniversario() {  
        System.out.println("Parabens pra você, ...!");  
        idade++;  
        System.out.println("Idade: " + idade);  
    }  
} // classe Gato
```

Instanciação e Instância

- **Instanciação** é o processo que permite um objeto ser criado a partir de uma classe.
- Quando instanciamos ("criamos") um objeto de uma classe, o objeto passa a existir na memória.
- E quando o objeto é instanciado podemos começar a usá-lo.
- Dizemos que um objeto é uma **instância** de uma classe.
- Exemplo: os objetos Tom, Frajola e Garfield são **instâncias** da classe **Gato**.

Instanciação e Instância

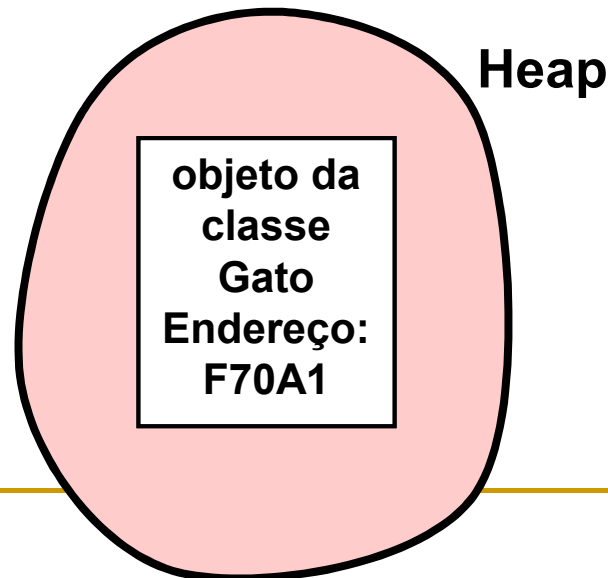
- Vamos instanciar um objeto gato da classe Gato:

```
public class InstanciandoUmGato {  
    public static void main(String[] args) {  
  
        new Gato();  
  
    }  
}
```

Instanciação e Instância

- O operador **new** na instrução
new Gato();

instancia um objeto da classe Gato em uma região de memória chamada **Heap**.



Instanciação e Instância

- **Problema:**

`new Gato();`

cria o objeto, mas não temos como acessá-lo, referenciá-lo!

- Sem acessar o objeto, não podemos utilizá-lo.

- **Solução:**

- ☐ usar uma variável para referenciar o objeto assim...

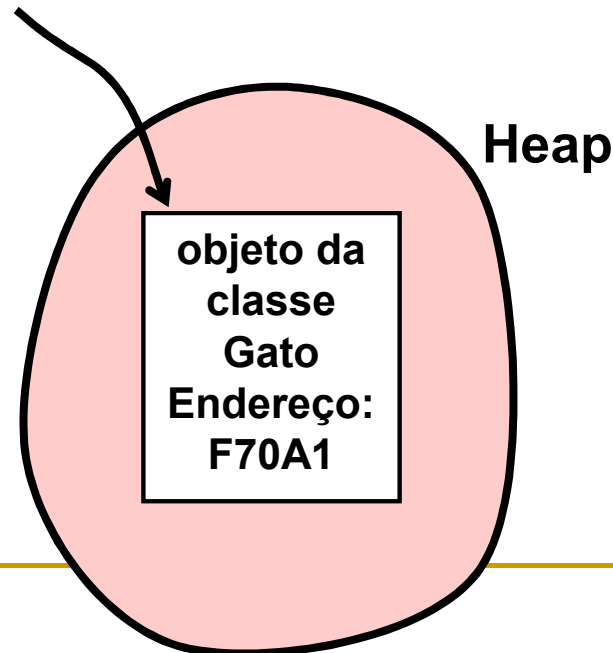
Instanciação e Instância

```
public class InstanciandoUmGato {  
    public static void main(String[] args) {  
  
        Gato frajola;  
        frajola = new Gato();  
  
    }  
}
```

Instanciação e Instância

- A variável **frajola** está referenciando o objeto.
- Agora, temos acesso ao objeto, e podemos utilizá-lo.

frajola



Instanciação e Instância

- Como podemos utilizar um objeto?
 - Atribuindo valores para seus atributos (atribuir um estado para o objeto).
 - Invocando suas operações, visando algum objetivo.

Instanciando e usando objetos

Arquivo: GatoPrincipal1.java

```
public class GatoPrincipal1 {  
  
    public static void main(String[] args) {  
        // Instanciação do objeto frajola da classe Gato.  
        Gato frajola = new Gato();  
  
        // Atribuindo dados ao gato frajola  
        frajola.nome = "Frajola";  
        frajola.idade = 2;  
        frajola.peso = 2.3f; // "f" de float.  
  
        .....  
    }  
}
```

Instanciando e usando objetos

Arquivo: GatoPrincipal1.java

```
public class GatoPrincipal1 {  
  
    public static void main(String[] args) {  
        .....  
  
        // Mostrando o estado do gato  
        System.out.println("Nome:" + frajola.nome);  
        System.out.println("Idade:" + frajola.idade);  
        System.out.println("Peso:" + frajola.peso);  
  
    } // fim main  
}
```

Instanciando e usando objetos

Arquivo: GatoPrincipal2.java

```
public class GatoPrincipal2 {  
    public static void main(String[] args) {  
        // Instanciando dois gatos
```

```
Gato fracjola = new Gato();
```

```
Gato garfield = new Gato();
```

```
// Atribuindo dados aos gatos
```

```
fracjola.nome = "Fracjola";
```

```
fracjola.idade = 2;
```

```
fracjola.peso = 2.3f; // "f" de float.
```

```
garfield.nome = "Garfield ";
```

```
garfield.idade = 1;
```

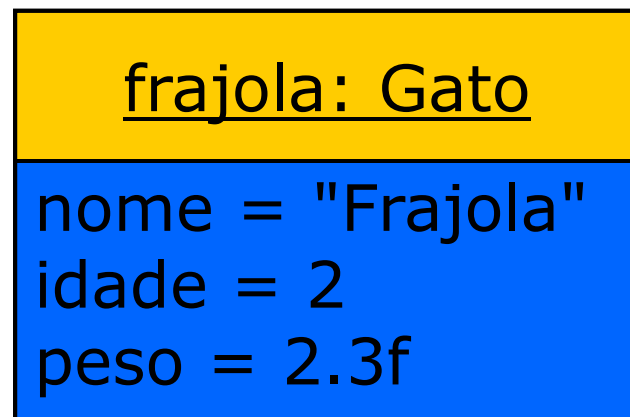
```
garfield.peso = 5.7f; // "f" de float.
```

Instanciando e usando objetos

Arquivo: GatoPrincipal2.java

```
public class GatoPrincipal2 {  
    public static void main(String[] args) {  
        ...  
        // Mostrando os estados dos gatos  
        System.out.println("Nome:" + frajola.nome);  
        System.out.println("Idade:" + frajola.idade);  
        System.out.println("Peso:" + frajola.peso);  
  
        System.out.println("Nome:" + garfield.nome);  
        System.out.println("Idade:" + garfield.idade);  
        System.out.println("Peso:" + garfield.peso);  
  
    } // fim main  
}
```

Objetos: Representação em UML

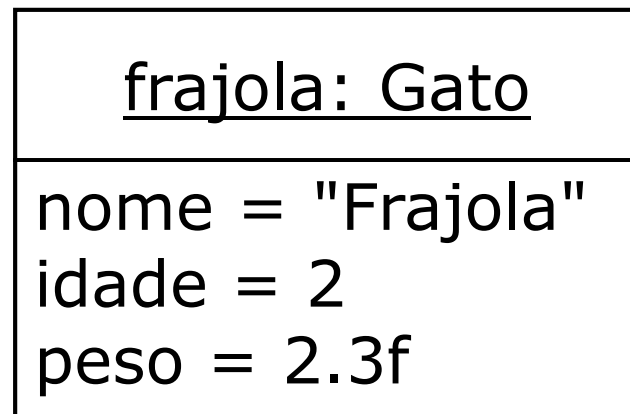


Nome do objeto
e da classe

Estado do objeto
frajola

Mas, UML não é colorido, assim!!

Objetos: Representação em UML



Nome do objeto
e da classe

Estado do objeto
frajola

Envio de Mensagem

- **Envio de mensagem** é quando invocamos uma operação de um objeto.
- Nada mais é do que chamar um método de um objeto.
- Exemplo:
 - envio da mensagem "frajola, coma":
frajola.comer();
 - envio da mensagem "frajola, corra":
frajola.correr();
 - envio da mensagem "frajola, faça aniversário":
frajola.fazerAniversario();

Envio de mensagens

Arquivo: GatoPrincipal3.java

```
public class GatoPrincipal3 {  
    public static void main(String[] args) {  
        // Instanciação...  
        Gato frajola = new Gato();  
  
        // Atribuindo dados ao gato frajola  
        frajola.nome = "Frajola";  
        frajola.idade = 2;  
        frajola.peso = 2.3f; // "f" de float.  
  
        .....  
    }  
}
```

Envio de mensagens

Arquivo: GatoPrincipal3.java

```
public class GatoPrincipal3 {  
    public static void main(String[] args) {  
.....  
        frajola.fazerAniversario();  
        frajola.correr();  
        frajola.correr();  
        frajola.comer();  
        frajola.fazerAniversario();  
    } // fim main  
}
```

Envio de Mensagem

- No programa **GatoPrincipal3** vemos

```
    frajola.nome = "Frajola";  
    frajola.fazerAniversario();
```

- Em todos os casos, temos:

```
        objeto.atributo  
        objeto.operação( parâmetros )  
        objeto.operação() . operação() . operação()
```

- Um membro (atributo ou operação) de um objeto é sempre referenciado em conjunto com o objeto ao qual pertence.

Receita para programar OO*

1. Descubra os **objetos** do problema.
2. Identifique as **classes** às quais pertencem os objetos.
3. Implemente as classes utilizando uma linguagem de programação OO.
4. Em seu programa, **instancie** objetos a partir das classes.
5. Use os objetos para resolver o problema inicial.
6. Venda seu programa e fique milionário!

Exercícios

Exercícios

Q1. Considere os seguintes objetos abaixo. Identifique a classe destes objetos, três de seus atributos e operações, e represente tudo em UML:

1. Volkswagen Fusca, Toyota Corolla, Honda Civic, Fiat Idea, e Ford Mondeo.
2. Objetos que marcam horas e minutos fabricados por Casio, Citizen, Rolex, Dumont, Seiko, etc.

Exercícios

Q2. Considere os seguintes objetos abaixo. Identifique a classe destes objetos, três de seus atributos e operações, e represente tudo em UML:

3. Objetos que aceitam e tocam mídias DVDs permitindo que possamos assistir aos filmes gravados nestas mídias.
4. Um poema de Drummond digitado no MSWord, uma carta para a tia Doquinha digitada no MSWord, a lista de compras do meu rancho digitada no MSWord, e várias páginas de um livro digitadas no MSWord.

Exercícios

Q3. Considere as classes **ContaBancária**, **Dado**, e **DocumentoMSWord**. Para cada classe, escreva em Java como você instanciaria um objeto e enviaria algumas mensagens para este objeto.

```
public class ContaBancariaTeste {  
    public static void main(String[] args) {  
        // Escreva o código aqui...  
    }  
}
```

Exercícios

Q4. Verifique se as afirmativas são verdadeiras ou falsas:

- a) Uma classe descreve os atributos e as operações (métodos) de um conjunto homogêneo de objetos.
- b) Na linguagem de modelagem UML, uma classe é representada por uma caixa com três partes: na parte superior escrevemos o nome da classe; na parte central escrevemos a lista dos atributos da classe; e na parte inferior da caixa escrevemos a lista das operações.

(continua)

Exercícios

Q4. Verifique se as afirmativas são verdadeiras ou falsas:

- c) Imagine uma classe chamada **Cachorro**. *Raça, cor, peso, nome, latir e dormir* são exemplos de atributos possíveis para a classe **Cachorro**.
- d) Em **$X \ y = \text{new } X();$**
a variável **y** é uma instância da classe **X**.
- e) Suponha que **x** seja um objeto da classe **X** e que **m()** seja um método de **X**. Para chamar o método **m()** devemos fazer **x.m()**;

(continua)

Exercícios

Q4. Verifique se as afirmativas são verdadeiras ou falsas:

- f) “Atributos” e “Métodos” são dois nomes diferentes que significam uma mesma coisa.
- g) Uma classe é uma instância de um objeto.
- h) A principal função de um atributo é armazenar os dados de um objeto; a de um método é armazenar as instâncias de uma classe.
- i) As características de um objeto (ou classe) podem ser chamadas de *operações* ou *métodos*.

(continua)

Exercícios

Q4. Verifique se as afirmativas são verdadeiras ou falsas:

- j) Sempre que uma mensagem é enviada para um objeto, o seu estado se altera.
- k) Duas instâncias quaisquer de uma classe podem ter estados iguais, e serem objetos distintos.
- l) **$K \ p = \text{new } K;$**
é um exemplo de instanciação de um objeto denominado **p** da classe **K** .

Exercícios

Q5. Considere o seguinte fragmento de código Java:

```
public class Cachorro {  
    double peso;  
    int idade;  
    void correr() {  
        if (peso > 1.0) {  
            peso--;  
        }  
    }  
    void comer() { peso++; }  
    void fazerAniversario() {  
        idade++;  
        comer(); // Parabéns!  
        comer(); // Parabéns!  
    }  
} // Cachorro
```

Exercícios

Q5. Considere o seguinte fragmento de código Java:

```
class CachorroTeste {  
    public static void main( String[] args ) {  
        Cachorro c = new Cachorro();  
        c.peso = /* escolha um número de 9.0 a 15.0 */;  
        c.idade = 5;  
        for(int i = 0; i < 3; i++) {  
            c.fazerAniversario();  
            c.correr();  
        }  
        c.comer();  
    }  
}
```

Exercícios

Q5. Após a execução do método `main()` de **CachorroTeste**, qual será o valor:

- a) Do atributo `idade`?
- b) Do atributo `peso`?

Exercícios

Q6. Escreva um programa em Java que atribua um peso alto para um cachorro, digamos 200 Kg, e faça o cachorro fazer exercícios até pesar 5 Kg.

Bibliografia

- Fowler, Martin. UML Essencial. 2o. ed. – Porto Alegre: Bookman, 2000.
- Deitel, Harvey. Java - Como Programar. 6a. ed. Porto Alegre: Bookman, 2006.
- Eckel, Bruce. Thinking in Java. 3a. edição. Prentice-Hall, 2001.