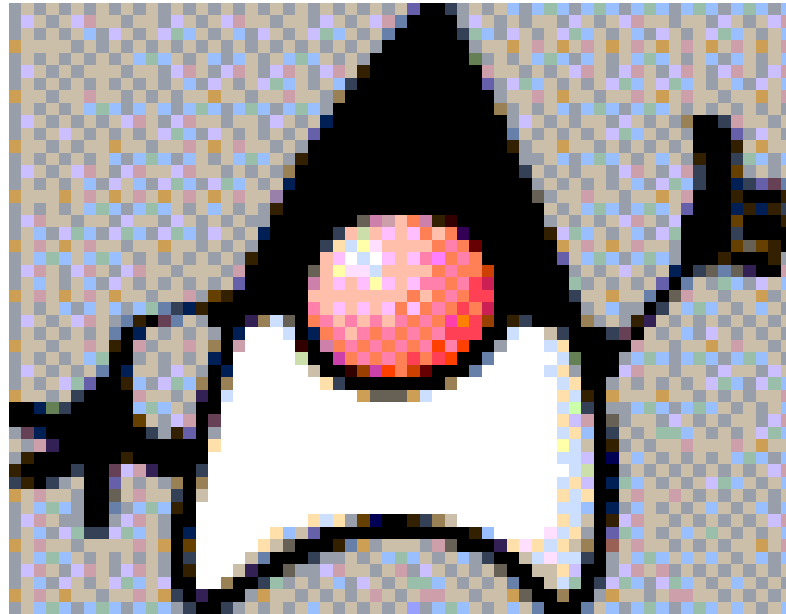


Introdução a Orientação a Objetos - Parte 1



Faculdade Dom Bosco de Porto Alegre
SisInfo & ADS - Estrutura de Dados
Prof^a Adriana Paula Zamin Scherer

Orientação a Objetos

- **Propósitos da Programação Orientada a Objetos**
 - ⇒ Prover mecanismos para visualizar a complexidade do desenvolvimento de software da mesma forma que visualizamos a complexidade do mundo real;
 - ⇒ Acelerar o desenvolvimento de softwares com base na modularidade;
 - ⇒ Melhorar a qualidade do software desenvolvido.

Orientação a Objetos

Classes X Objetos :

- ⇒ Uma classe não é um objeto, uma classe é a definição de como um objeto se comportará quando este for criado.
- ⇒ As classes definem um novo tipo de dado. Se for criada uma classe Caixa, podemos a partir dela criar novos objetos do tipo Caixa.
- ⇒ Uma classe é uma construção de software que possui variáveis de instância e métodos.
- ⇒ Portanto, uma classe é o carimbo de um objeto, e o objeto é uma instância de uma classe. Como um objeto é uma instância de uma classe, estes dois termos serão utilizados de forma intercambiável.

Classes

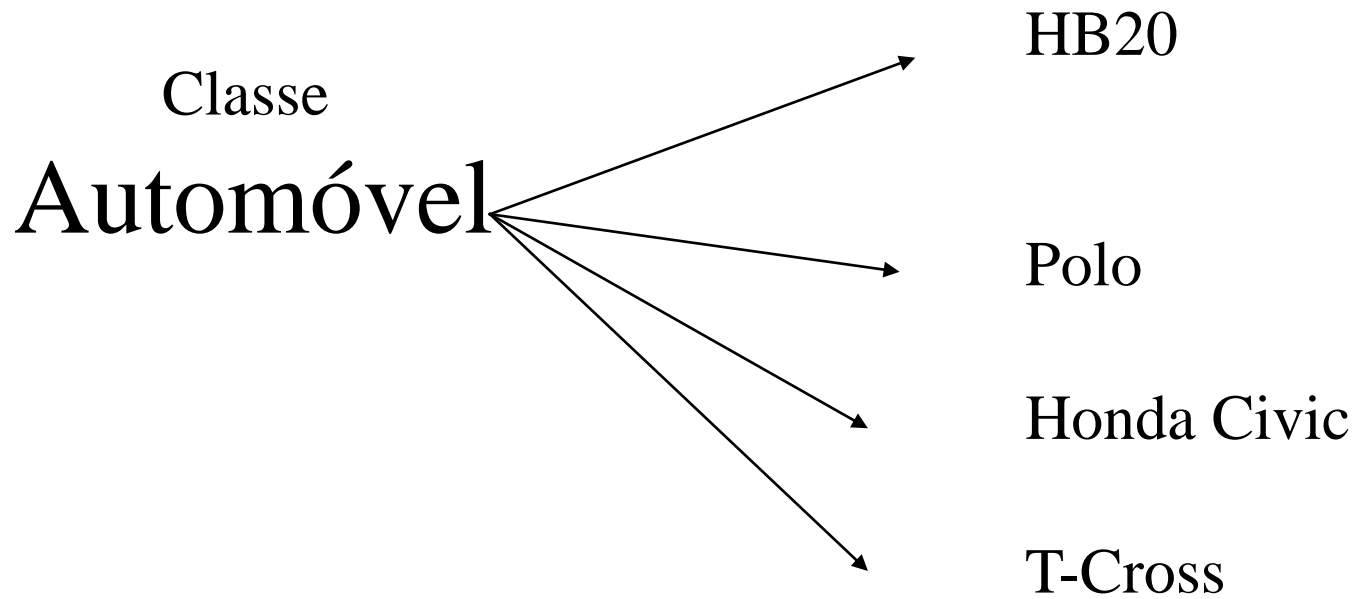
- Definem a estrutura e o comportamento de um tipo de objeto;
- Permitem a instanciação de um ou mais objetos de um mesmo tipo.

Nome
Atributos
Métodos

Classes

Visão do mundo real:

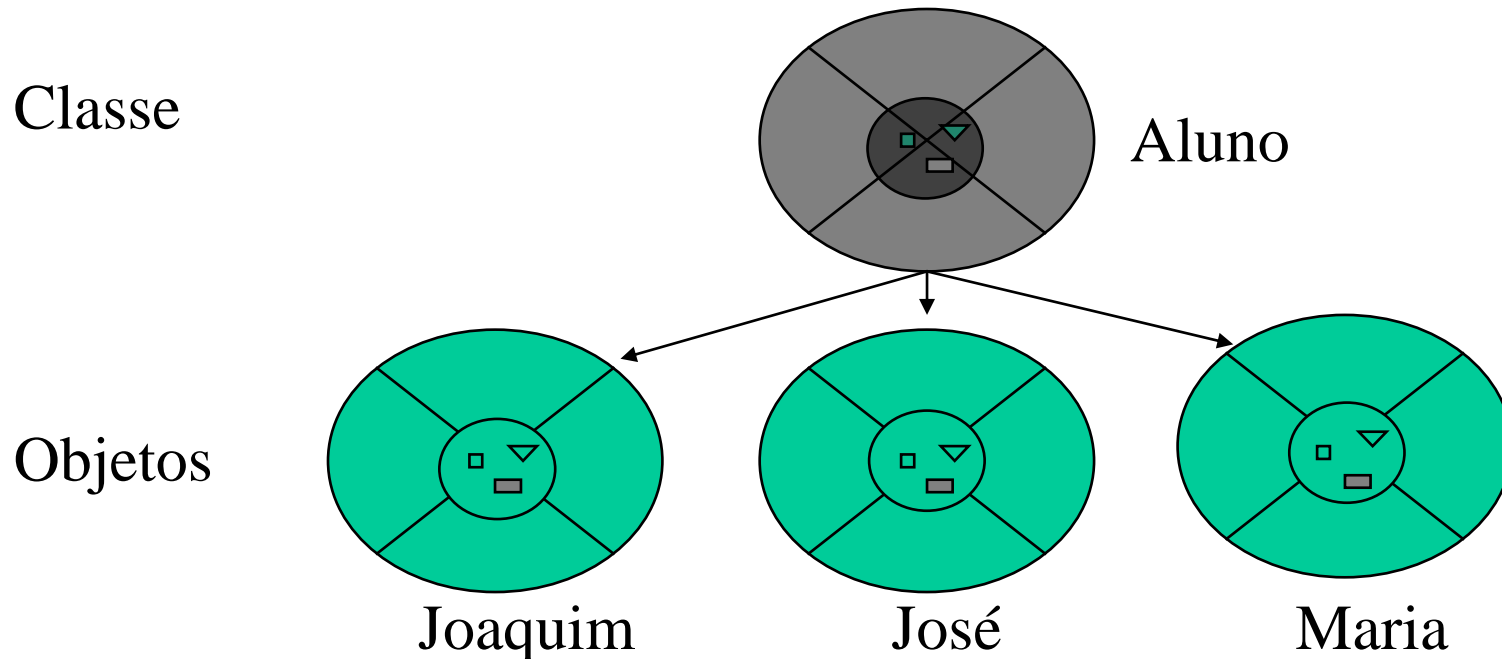
- moldes para criação de objetos;
- especificam propriedades e ações em comum a todos seus objetos.



Classes

Classes - Visão em POO:

Um protótipo que define os atributos e os métodos comuns a todos objetos de um determinado tipo e da própria classe, com a finalidade de servir como molde para a criação de objetos.



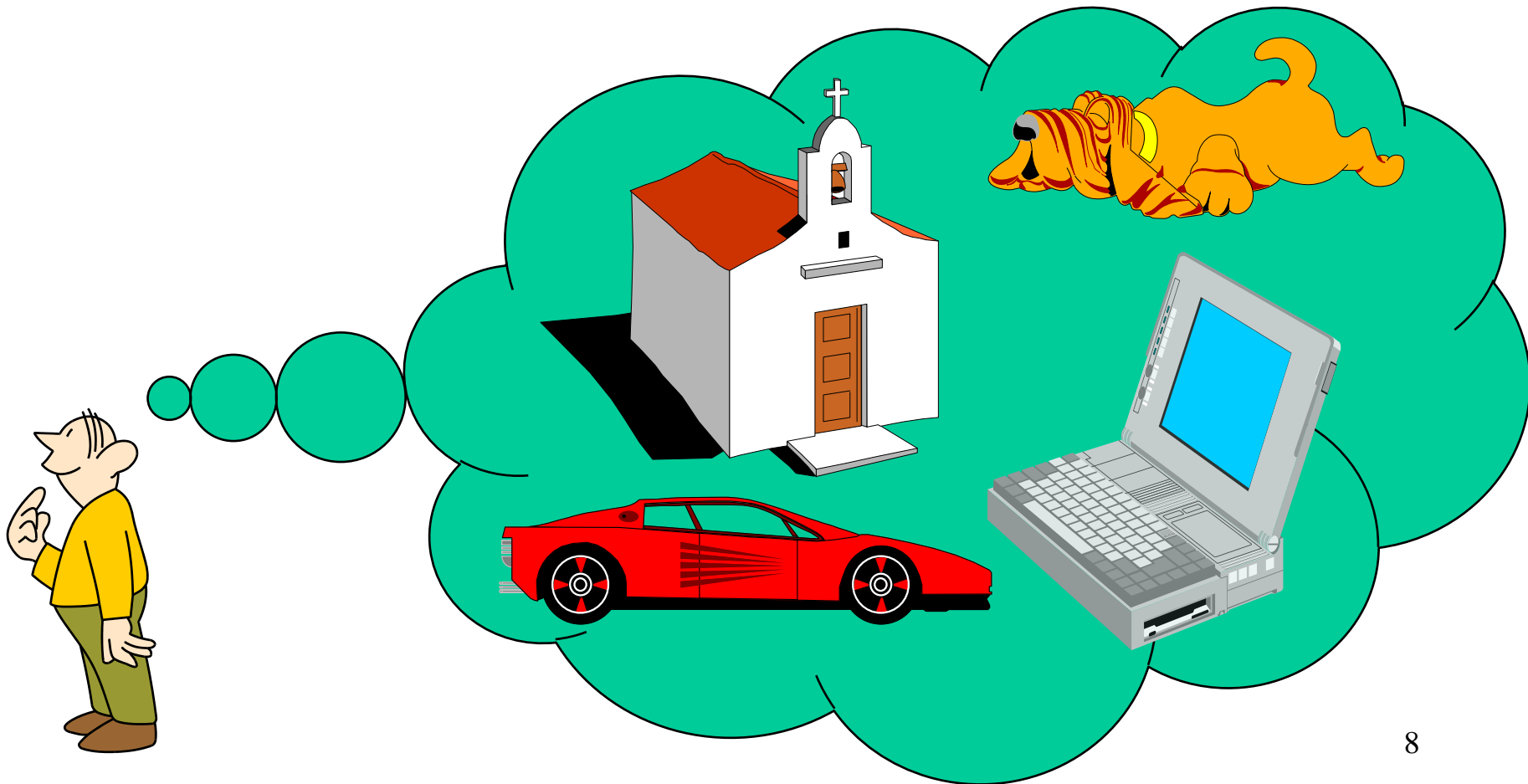
Objetos

- **Objeto** é uma representação abstrata de uma entidade do mundo real, que tem um identificador único, propriedades embutidas e a habilidade de interagir com outros objetos e consigo mesmo.
- **Estado do objeto** é um conjunto de valores que os atributos do objeto podem ter em um determinado instante do tempo.

Objetos

Visão do mundo real:

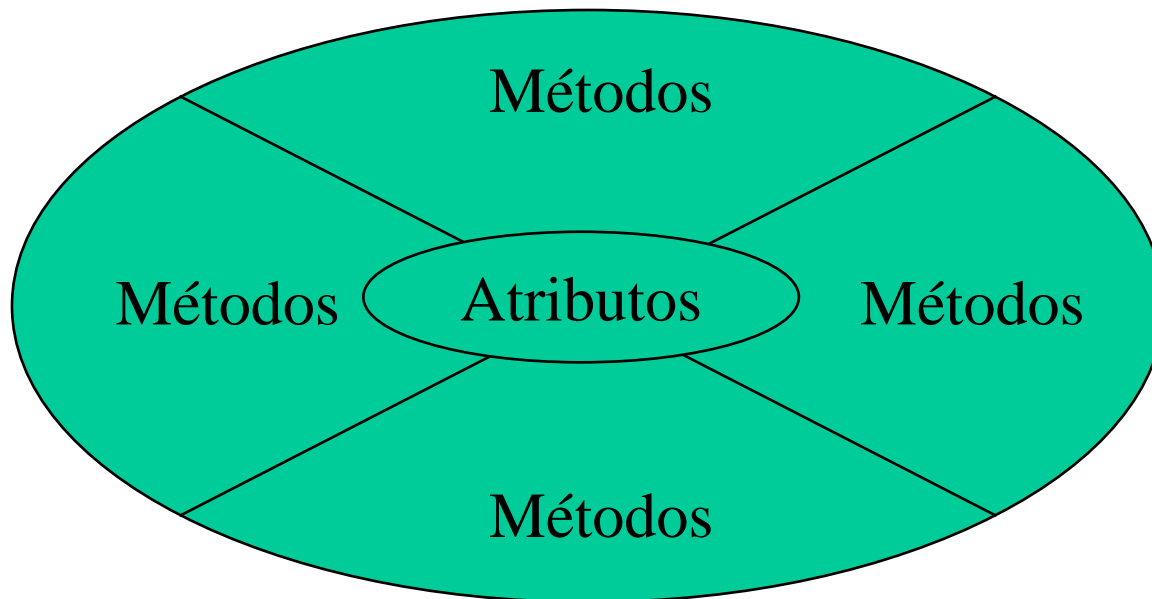
- Na nossa vida, estamos cercados por objetos.



Objetos

Visão em POO:

Um conjunto distinto de atributos e métodos que representam entidades do mundo real, implementados por software e instanciados através de classes.



Criando Classes em Java

- Para definir uma classe utiliza-se a palavra chave *class* e o nome da classe. Supondo a classe `Caixa`.

```
class Caixa  
{  
    // Definição dos atributos e métodos  
}
```

Criando Classes em Java

- Atributos: também chamados de variáveis de instância, isto é, são pertencentes a própria classe, destinadas a armazenar alguma informação e que serão utilizadas pelos objetos quando eles forem criados.
- Métodos: enquanto os atributos armazenam os dados associados aos objetos, os métodos podem realizar operações com os atributos ou ainda realizar ações com os objetos. Por meio dos métodos é possível imitar o comportamento de um objeto no mundo real.

Criando Classes em Java

- Exemplo:

```
class Caixa
{
    // atributos
    double largura;
    double altura;
    double profundidade;

    // métodos
    double volume ()
    {
        return largura * altura * profundidade;
    }
}
```

Criando Classes em Java

- No exemplo anterior percebemos que a classe Caixa não é uma classe executável, mas apenas um modelo a partir do qual podem ser criados outros objetos. Esta é a base de funcionamento de toda a linguagem Java.

Criando Objetos em Java

- Objetos são criados através da declaração `new`, seguida de um método construtor. Exemplo:

```
Caixa caixaSapato = new Caixa();
```

- O método construtor deve obrigatoriamente receber o mesmo nome da sua classe;
- Se um construtor não for declarado, é assumido um construtor *default* da linguagem Java.
- O método construtor gera uma instância do objeto em memória e o seu código é executado imediatamente após a criação do objeto provendo-lhe um estado inicial;

Criando Objetos em Java

- A declaração de uma variável referência a um objeto não aloca espaço em memória. A alocação de memória é feita somente quando o objeto é criado.
- Considere o exemplo abaixo:

```
Caixa  caixaSapato;  
caixaSapato = new Caixa();
```

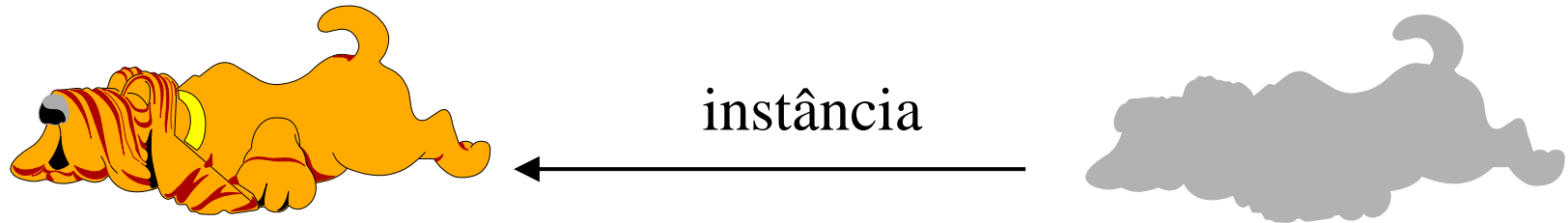
- O primeiro comando, a declaração, aloca apenas o espaço suficiente para a referência. O segundo, aloca o espaço para os atributos do objeto `caixaSapato`.
- Somente após a criação do objeto é que seus membros (atributos e métodos) podem ser referenciados.

Exemplo

```
class CaixaExemplo
{
    public static void main (String args[])
    {
        Caixa caixaSapato = new Caixa();
        caixaSapato.altura = 10;
        caixaSapato.largura = 30;
        caixaSapato.profundidade = 50;

        System.out.println( "Volume da caixa de
sapato e: " + caixaSapato.volume() );
    }
}
```


Mais sobre atributos ...



atributos do objeto

- Nome
- Cor do pêlo
- Idade

atributos da classe (estáticos)

- Número de patas

Exemplo

```
class Cachorro
{
    // Atributos da classe Cachorro
    static final int NUMERODEPATAS = 4;

    // Atributos do objeto (criados quando o objeto é instanciando)
    String nome;
    int idade;
    String corDoPelo;
}
```

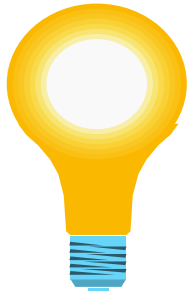
Exemplo

```
public class Canil
{
    public static void main(String args[])
    {
        Cachorro rex = new Cachorro();
        rex.nome = "Rex";
        rex.idade = 2;
        rex.corDoPelo = "Marrom";
        System.out.println("Todo cachorro tem " +
            Cachorro.NUMERODEPATAS + " patas");
        System.out.println("O cachorro "+ rex.nome +
            " tem pêlo de cor:" + rex.corDoPelo);
    }
}
```

Mais sobre métodos ...

Classe: Lâmpada

Objeto: philips60w



Métodos

- acender
- apagar
- indicarVoltagem

Referência a Membros de Objetos e Classes

Variável

< referência.variável >

rex.idade = 2;

Método

< resultado = referência.método(parâmetros) >

caixaSapato.volume();

- Os parâmetros, quando existentes, são posicionais e são tipados
- Número de parâmetros é verificado em tempo de compilação

Retornando valor em métodos

- Na definição do nome do método deve ser indicado se haverá ou não o retorno de valores.

- Com retorno de valores:

```
int quadrado (int y)
{
    return y * y;
}
```

- sem retorno de valores:

```
void setValor (int y)
{
    valor = y;
}
```

Exercícios