

```
this.addData(segment);  
//Set up tooltip events on the chart  
11 (this.options.showTooltips){  
    helpers.bindEvents(this, this.options.tooltipEvents, function(evt) {  
        var activeSegments = (evt.type !== 'mouseout') ? this.segments :  
        helpers.each(this.segments, function(segment){  
            segment.restore(["fillColor"]);  
        });  
        helpers.each(activeSegments, function(activeSegment){  
            activeSegment.fillColor = activeSegment.highlightColor;  
        });  
        this.showTooltip(activeSegments);  
    });  
});
```

# Introdução à Orientação à Objetos

DS - 2º modulo  
Profa. Rosangela Freire

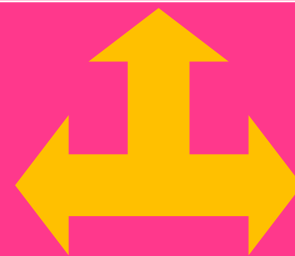
(Slides - Profa. Vanessa Ferraz)



São modelos, padrões ou estilos de programação, suportados por linguagens que agrupam características em comum.

## Paradigmas de Programação

Estruturado,  
procedural,  
procedimental



Orientado à  
objetos

# Paradigma procedimental

Conjuntos de instruções organizados em blocos para executar tarefas (instruções).

Todos os programas são reduzidos a apenas três estruturas: sequência, iteração, decisão.

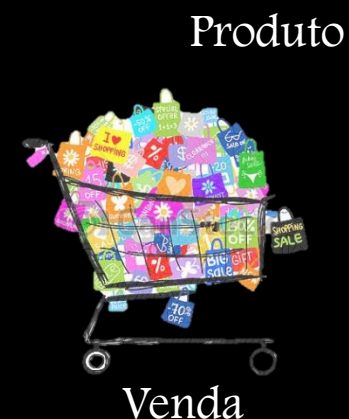
Uso de sub-rotinas e os dados são globais, para todas as sub-rotinas.

Maior dificuldade na manutenção do código.

Programas centrados no conceito de um estado e ações que manipulam o estado.

# Paradigma orientado à objetos

Representação de cada elemento do mundo real em um objeto.



Representação do sistema se aproxima do mundo real.

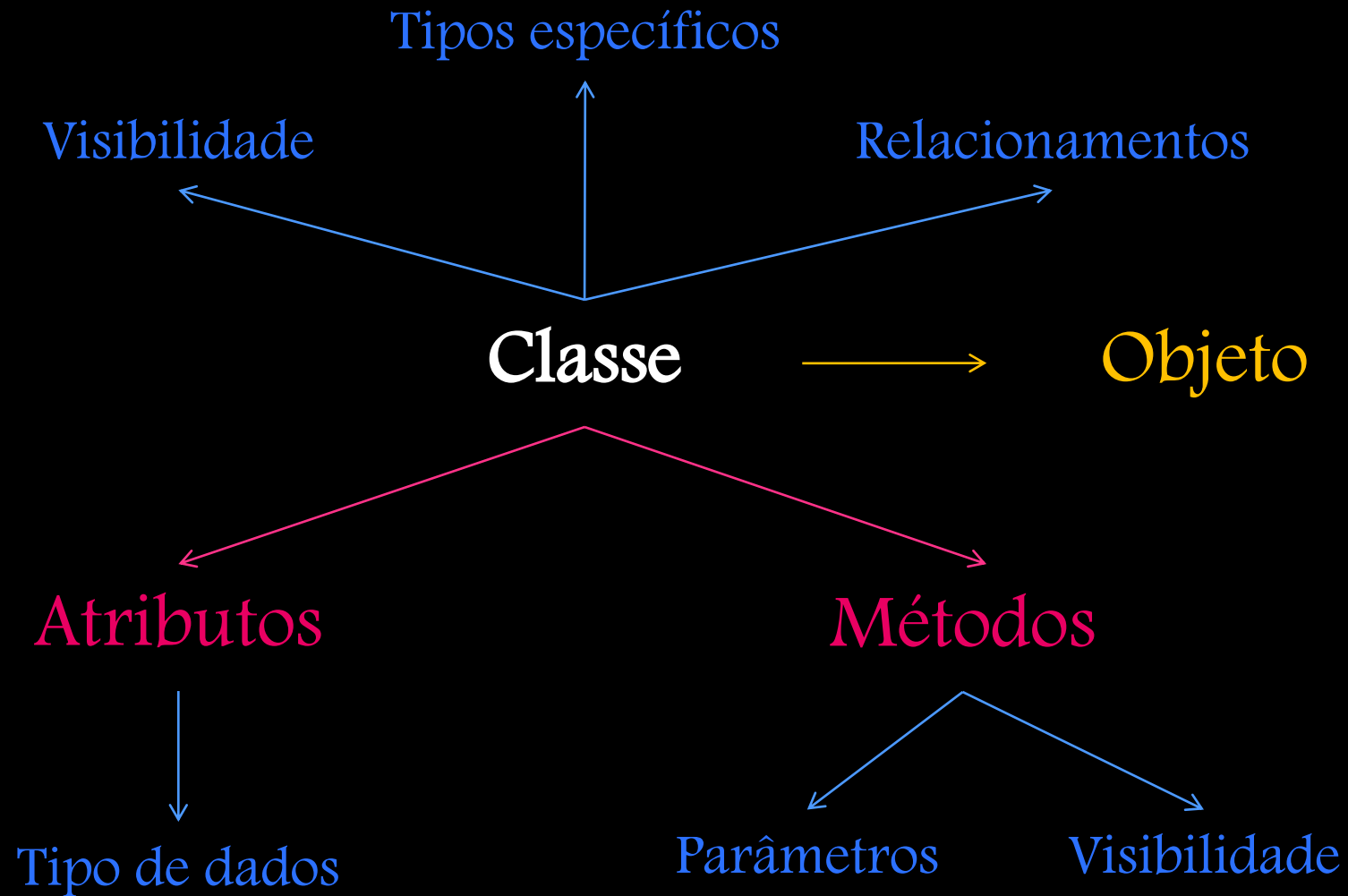
Ganhos no desenvolvimento, leitura e manutenção do código com o reaproveitamento de código fonte.

Fornece mapeamento direto entre o mundo real e as unidades de organização utilizadas no projeto.

Possui objetos com propriedades e operações que podem ser executadas.

Paradigma  
orientado à  
objetos

# Quando programo, encontro...



# classe

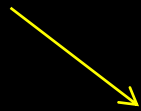
É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

É uma forma de definir um tipo de dados em uma linguagem orientada à objetos.

visibilidade

nome da classe:

SEMPRE COM A INICIAL MAIÚSCULA



```
public class Principal {
```

```
//seu código estará aqui
```

```
}
```



# objeto

É a instância de uma classe.

É a entidade capaz de reter um estado e oferecer uma série de comportamentos para examinar ou afetar o seu estado.

classe  $\neq$  objeto

?

Um **prédio** é construído a partir de uma **planta**.

Um **bolo** é feito a partir de uma **receita**.

!

A classe é o modelo com as características e comportamentos para criação dos objetos, cada um com as suas informações e exercendo os seus comportamentos.



# Você já criou classes sabia?

```
import java.util.Scanner;
public class Principal {
    public static void main(String[] args) {
        Scanner leitor = new Scanner(System.in);
        String nome, cpf;
        int codigo;

        System.out.println("Informe o nome do Cliente:");
        nome = leitor.next();
        System.out.println("Informe o cpf do Cliente:");
        cpf = leitor.next();
        System.out.println("Informe o código do Cliente:");
        codigo = leitor.nextInt();
        System.out.println("Cliente cadastrado com sucesso!!");
    }
}
```

# Como encontrar as classes?

Elas estão presentes no contexto do projeto do sistema.



Quais seriam as classes de um sistema para uma escola?

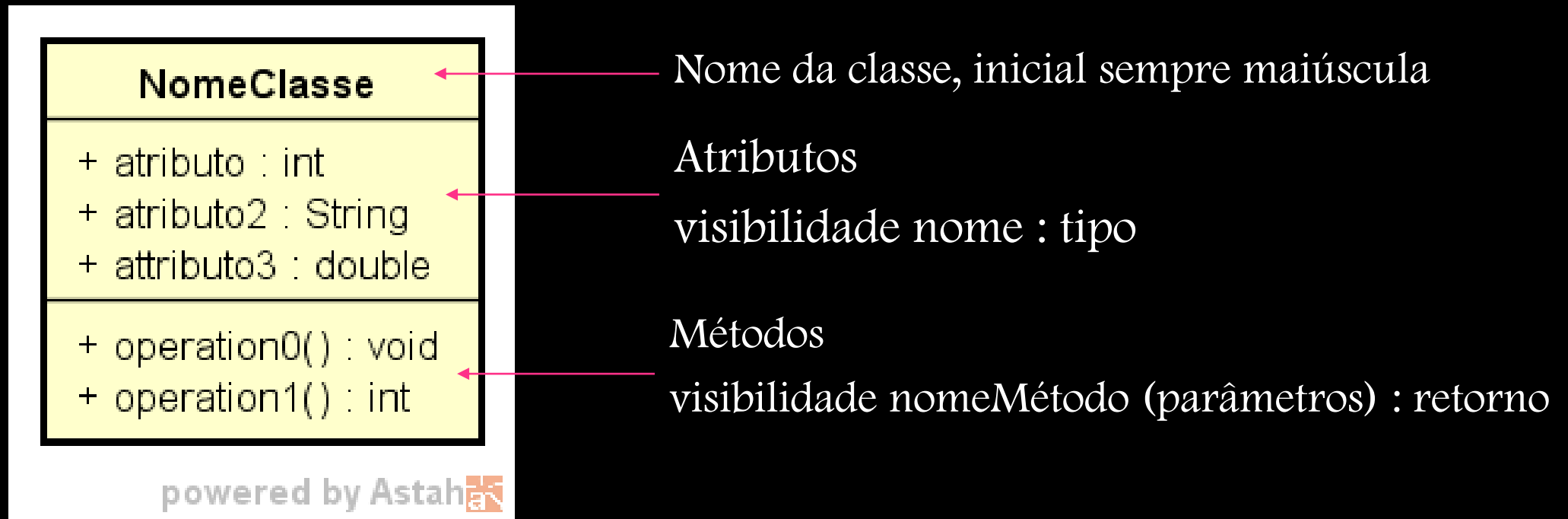
E para um sistema de vendas de uma loja de calçados?



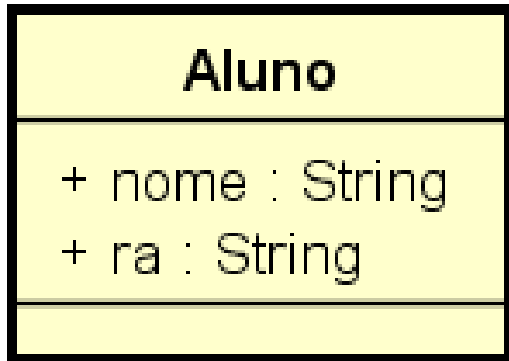
# UML

A UML é utilizada para projetar o desenvolvimento de sistemas orientados à objetos.

Ela possui 13 diagramas, um deles é o diagrama de classes, que mostram as classes e os seus relacionamentos para desenvolvermos os sistemas.

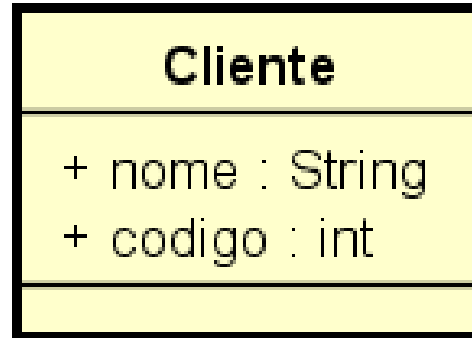


# Alguns exemplos



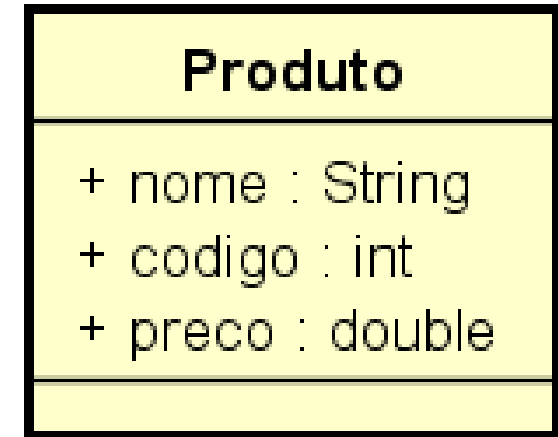
powered by Astah

```
public class Aluno{
    String nome;
    String ra;
}
```



powered by Astah

```
public class Cliente{
    String nome;
    int codigo;
}
```



powered by Astah

```
public class Produto{
    String nome;
    int codigo;
    double preco;
}
```