

Introdução à orientação a objeto

Classes e Objetos

Orientação a objeto

Imagine o seguinte cenário: Você foi incumbido de programar o novo placar automático do estádio do seu bairro

Não existem muitas restrições, só deve mostrar quais são os times e quantos gols cada um fez

Orientaçã o a objeto



Pensando em linguagem C você logo implementa o código



Passando alguns anos mudam o placar e o seu código não funciona mais



Se você mudar o código isso pode fazer com que ele não funcione mais em um placar antigo



É importante manter o legado e só especificar seu código para modelos de placares diferentes

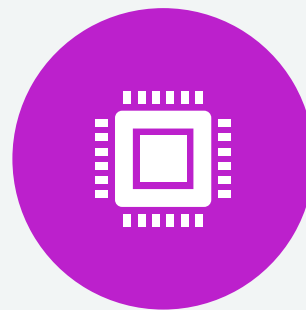
Orientação a objeto



PARADIGMA DE LINGUAGEM DE
PROGRAMAÇÃO



TEM COMO ESTRUTURAS
BÁSICAS AS CLASSES



NASCEU COMO UMA FORMA DE
ALIVIAR A MODELAGEM DE
PROBLEMAS REAIS EM MEIOS
DIGITAIS



APLICA CONCEITOS
BIOLÓGICOS

Classe



Pode ser considerado um tipo abstrato de dados (TAD)



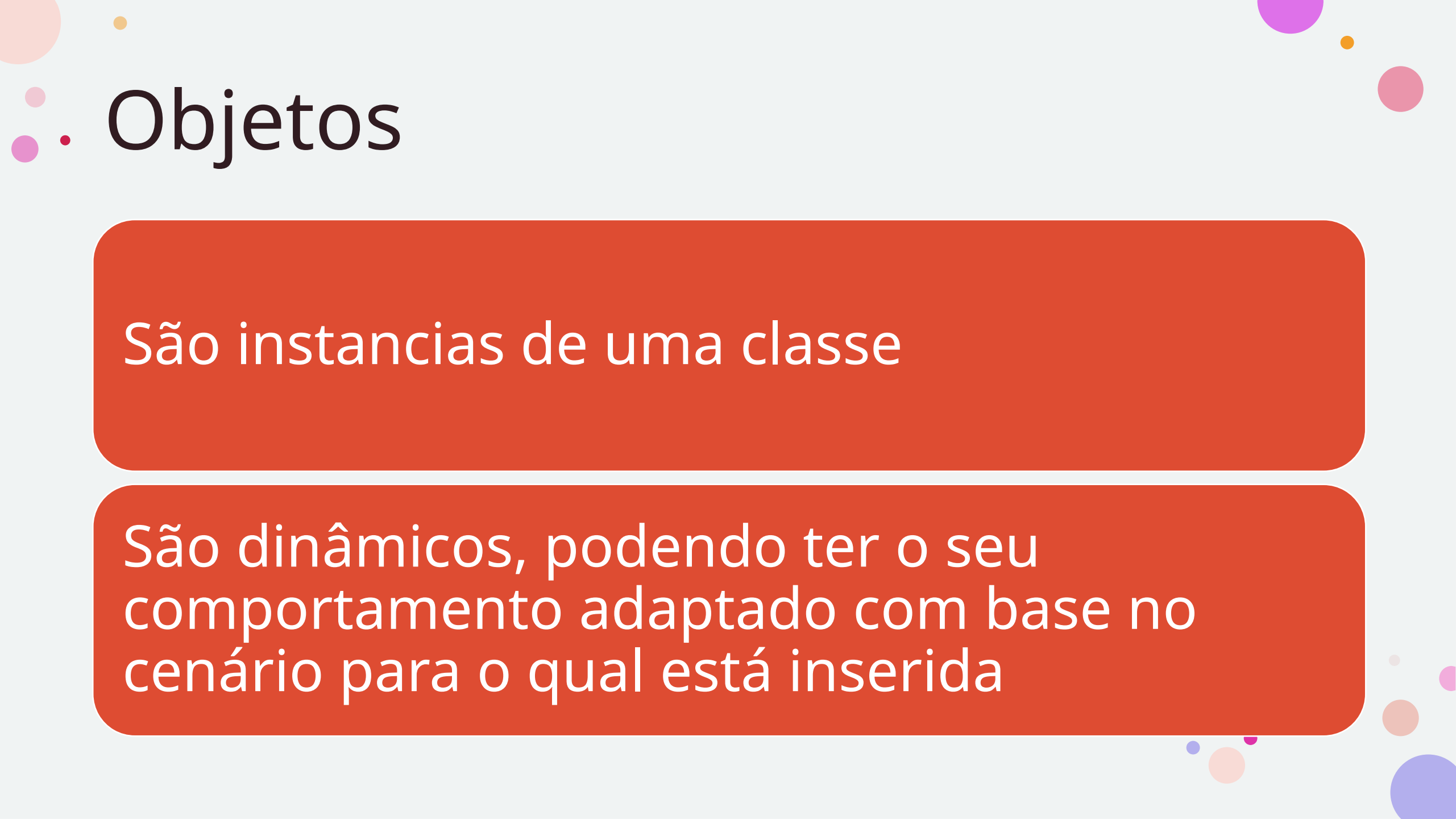
É definida por suas características

Atributos



E comportamentos

Métodos



Objetos

São instancias de uma classe

São dinâmicos, podendo ter o seu comportamento adaptado com base no cenário para o qual está inserida

Atributos



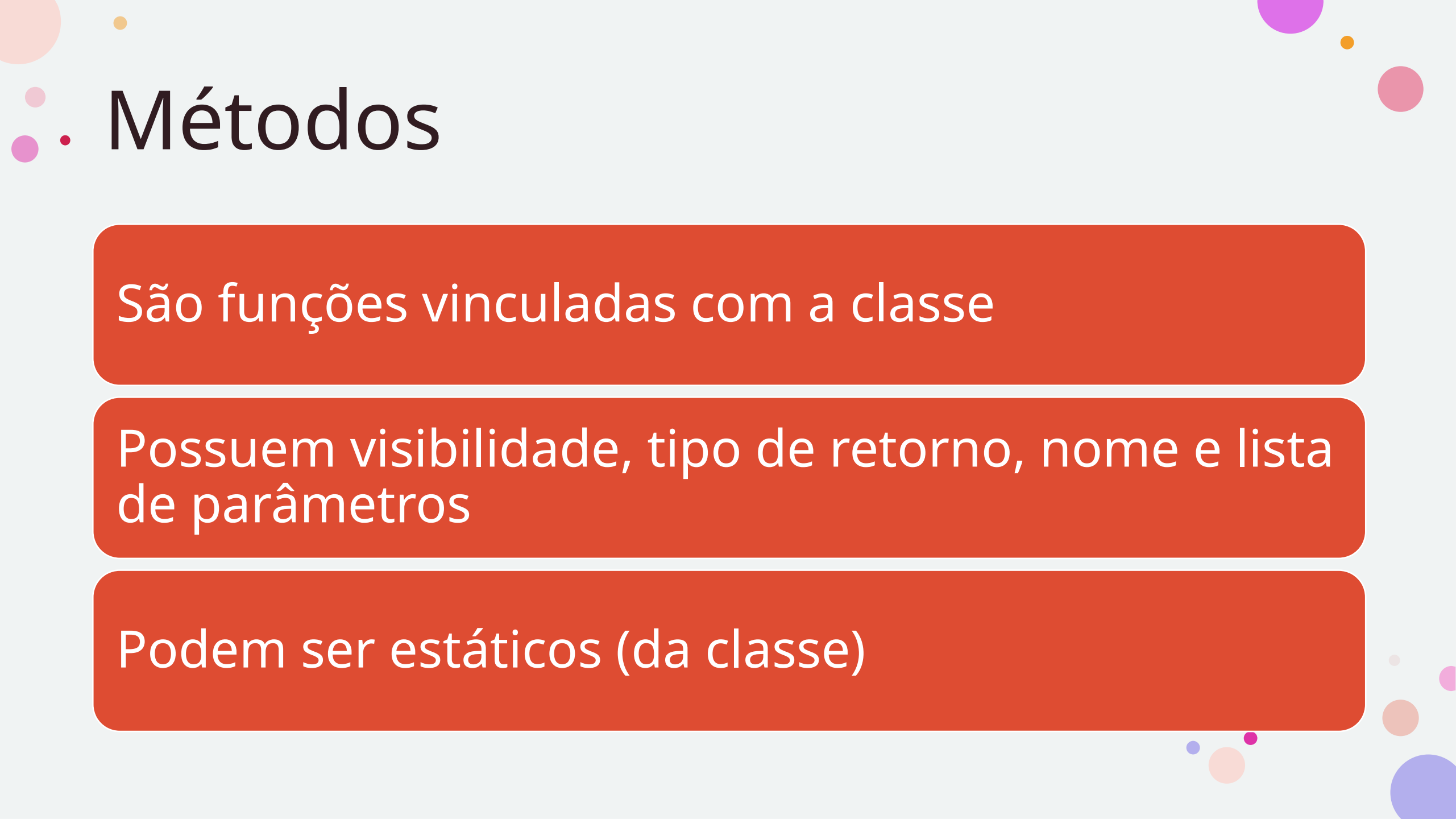
São os itens de dados que descrevem quais características uma classe pode ter



Possuem visibilidade, tipo e nome



Pode ser estáticos (atributos da classe)



Métodos

São funções vinculadas com a classe

Possuem visibilidade, tipo de retorno, nome e lista de parâmetros

Podem ser estáticos (da classe)

The background features several decorative elements: a pink circle with a blue flower-like pattern in the top-left; a pink circle with a radial line pattern in the top-right; a pink circle with a wavy line pattern in the bottom-left; and a purple circle with a dashed line pattern in the bottom-right. There are also several small solid circles in various colors (pink, orange, purple, grey) scattered around the main shapes.

Implementando o placar

Vamos usar o Eclipse uma IDE para Linguagem Java

Display

```
1 package placar;
2
3 public class Display {
4
5     public static void main(String[] args) {
6         System.out.println("INICIALIZANDO O PLACAR!");
7     }
8
9 }
```

Display

```
1 package placar;
2
3 public class Display {
4     public static void main(String[] args) {
5         System.out.println("INICIALIZANDO O PLACAR!");
6     }
7 }
8
9 }
```

Define o pacote

Define a visibilidade da classe

Nome da classe

Define que esse item vai ser uma classe

Define um método

Display

Define a visibilidade do método
(public, private ou protected)

Define que o tipo de retorno é
void (sem retorno)

Define que é
estático

Nome do
método

Lista de
parâmetros

```
5 public static void main(String[] args) {  
6     System.out.println("INICIALIZANDO O PLACAR!");  
7 }
```

Chamando o método estático println

Método main

Todo código
Java precisa de
um método
main

É o início do
código

Análogo a
função main
na linguagem
C

Criando a classe do placar

```
public class PlacarBasico {  
  
    private String timeA;  
    private String timeB;  
    private int golTimeA;  
    private int golTimeB;  
  
    public PlacarBasico(String timea, String timeb) {  
        this.timeA = timea;  
        this.timeB = timeb;  
        this.golTimeA = 0;  
        this.golTimeB = 0;  
    }  
  
    public void mostraPlacar() {  
        System.out.printf("%s X %s\n", this.timeA, this.timeB);  
        System.out.printf("%03d = %03d", this.golTimeA, this.golTimeB);  
    }  
}
```

Criando a classe do placar

```
public class PlacarBasico {  
  
    private String timeA;  
    private String timeB;  
    private int golTimeA;  
    private int golTimeB;  
  
    public PlacarBasico(String timea, String timeb) {  
        this.timeA = timea;  
        this.timeB = timeb;  
        this.golTimeA = 0;  
        this.golTimeB = 0;  
    }  
  
    public void mostraPlacar() {  
        System.out.printf("%s X %s\n", this.timeA, this.timeB);  
        System.out.printf("%03d = %03d", this.golTimeA, this.golTimeB);  
    }  
}
```

Criando atributos

Criando a classe do placar

Define a visibilidade do atributo
(public, private ou protected)

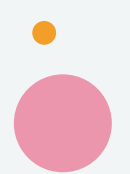
Define o
tipo

Define o nome do
atributo

```
private String timeA;  
private String timeB;  
private int golTimeA;  
private int golTimeB;
```





Atributos



Normalmente, em Java, os atributos são declarados como privados, não podendo ser acessado fora do objeto

Isso é feito para facilitar o encapsulamento



Criando a classe do placar

```
public class PlacarBasico {  
  
    private String timeA;  
    private String timeB;  
    private int golTimeA;  
    private int golTimeB;  
  
    public PlacarBasico(String timea, String timeb) {  
        this.timeA = timea;  
        this.timeB = timeb;  
        this.golTimeA = 0;  
        this.golTimeB = 0;  
    }  
  
    public void mostraPlacar() {  
        System.out.printf("%s X %s\n", this.timeA, this.timeB);  
        System.out.printf("%03d = %03d", this.golTimeA, this.golTimeB);  
    }  
}
```

Criando métodos

Criando a classe do placar

Método construtor

```
public PlacarBasico(String timea, String timeb) {  
    this.timeA = timea;  
    this.timeB = timeb;  
    this.golTimeA = 0;  
    this.golTimeB = 0;  
}
```

This faz referência ao próprio objeto

```
public void mostraPlacar() {  
    System.out.printf("%s X %s\n", this.timeA, this.timeB);  
    System.out.printf("%03d = %03d", this.golTimeA, this.golTimeB);  
}
```

Método construtor

Método especial
que é chamado
quando a classe
constrói um
objeto

Não possui tipo
de retorno

Deve ter o
mesmo nome
que a classe

Projeto principal

```
package placar;

public class Display {

    public static void main(String[] args) {
        PlacarBasico pl = new PlacarBasico("MAC", "NOR");
        pl.mostraPlacar();
    }
}
```

Projeto principal

```
package placar;  
  
public class Display {  
  
    public static void main(String[] args) {  
        PlacarBasico pl = new PlacarBasico("MAC", "NOR");  
        pl.mostraPlacar();  
    }  
}
```

Cria um novo objeto da classe PlacarBasico

Chama o método mostraPlacar da objeto pl

```
1 package placarTOP;
2
3 public class PlacarBasico {
4
5     private String nomeTimeA;
6     private String nomeTimeB;
7     private int pontuacaoA;
8     private int pontuacaoB;
9
10    public PlacarBasico(String ta, String tb) {
11        this.nomeTimeA = ta;
12        this.nomeTimeB = tb;
13    }
14
15    public void mostrarPlacar() {
16        System.out.printf("%s X %s\n", this.nomeTimeA, this.nomeTimeB);
17        System.out.printf("%03d - %03d", this.pontuacaoA, this.pontuacaoB);
18        this.pontuacaoA = 0;
19        this.pontuacaoB = 0;
20    }
21
22    public void pontuar(int time) {
23        if (time == 0) {
24            this.pontuacaoA += 1;
25        } else {
26            this.pontuacaoB += 1;
27        }
28    }
29
30 }
```

Entrada de dados

```
public static void main(String[] args) {  
  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Digite o nome do time A:");  
    String nomeA = sc.nextLine();  
    System.out.println("Digite o nome do time B:");  
    String nomeB = sc.nextLine();  
    sc.close();  
  
    PlacarBasico pb = new PlacarBasico(nomeA, nomeB);  
    pb.pontuar(0);  
    pb.pontuar(0);  
    pb.pontuar(0);  
    pb.mostrarPlacar();  
}
```


Entrada de dados

```
public static void main(String[] args) {  
  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Digite o nome do time A:");  
    String nomeA = sc.nextLine();  
    System.out.println("Digite o nome do time B:");  
    String nomeB = sc.nextLine();  
    sc.close();  
  
    PlacarBasico pb = new PlacarBasico(nomeA, nomeB);  
    pb.pontuar(0);  
    pb.pontuar(0);  
    pb.pontuar(0);  
    pb.mostrarPlacar();  
}
```

Cria um objeto da classe Scanner, passa o System.in como parâmetro

Lê a próxima linha e armazena na String nomeB