Programação Orientada a Objetos

Composição de Classes

Prof. Rui César das Neves, Prof. José Cordeiro

Departamento de Sistemas e Informática
Escola Superior de Tecnologia de Setúbal – Instituto Politécnico de Setúbal
2014/2015

Sumário

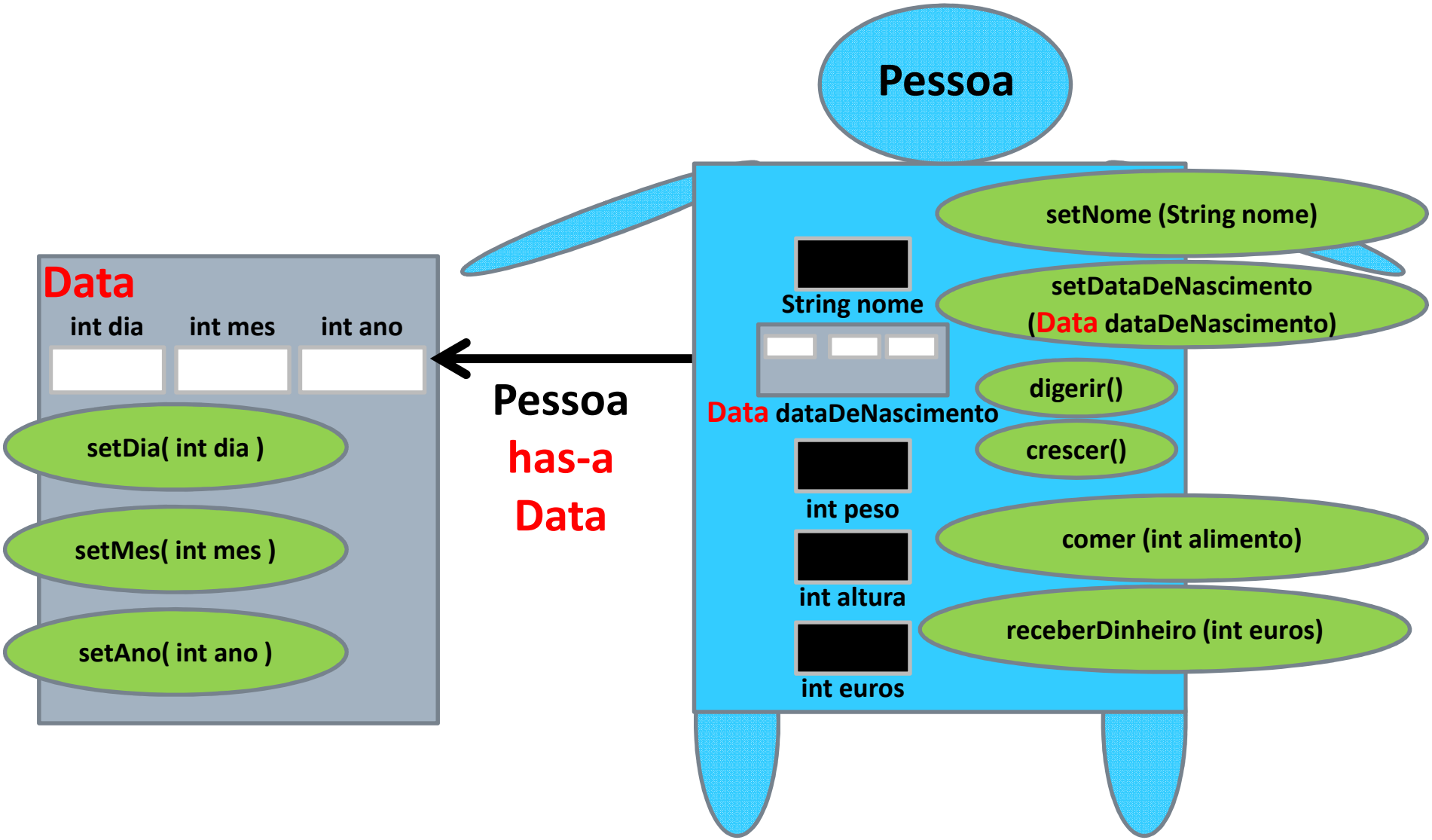
- Composição de classes – “has-a”
  - Uma classe tem como atributo(s) objeto(s) de outra(s) classe(s)

Composição de Classes (has-a)

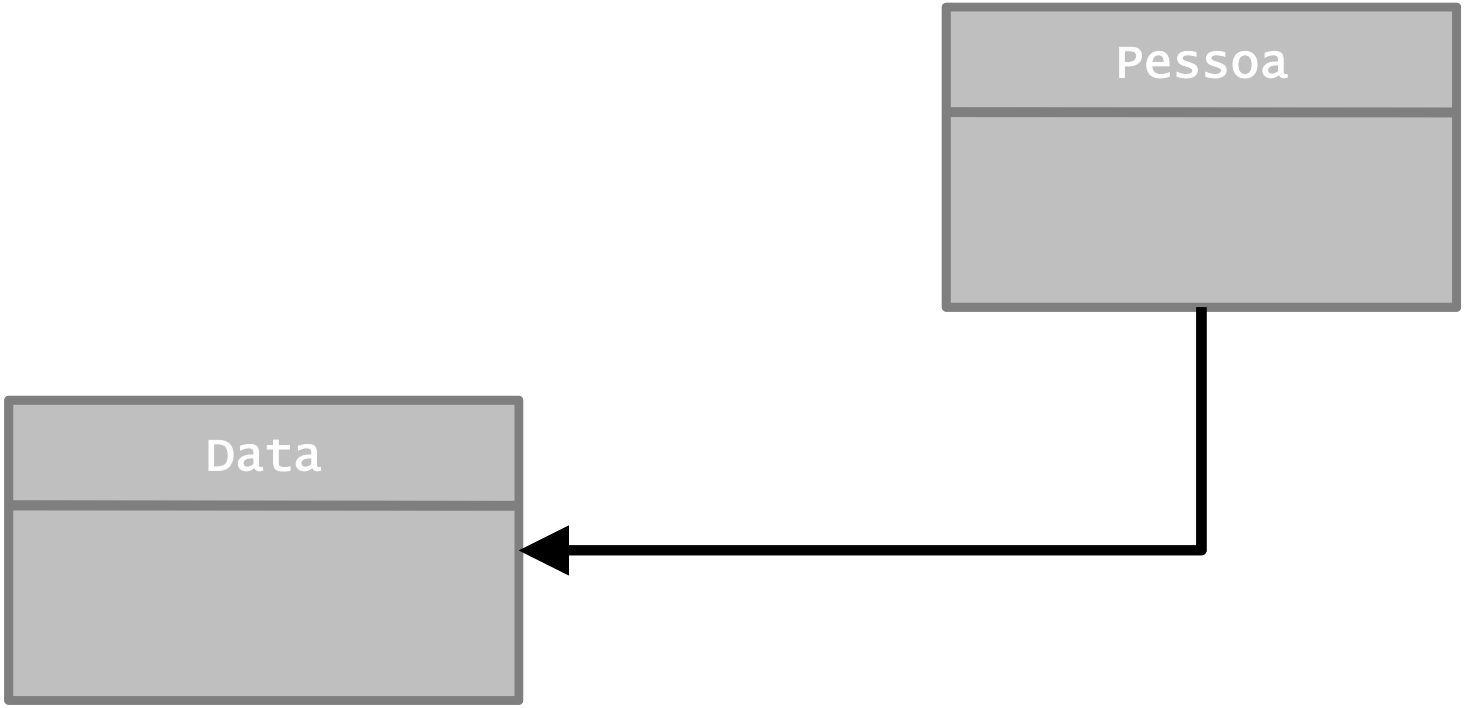
- **A Composição** de classes consiste em incluir como atributos de uma classe, objetos de outras classes anteriormente definidas.

- O mecanismo de composição estabelece uma relação de inclusão entre classes:
  - Se a classe A contém (*has a*) objectos da classe B (dado que alguns dos seus atributos são objetos da classe B)…
  - Diz-se que a Classe A é composta por B.
  - Na relação, dir-se-á que B é parte da definição de A (*part-of*).

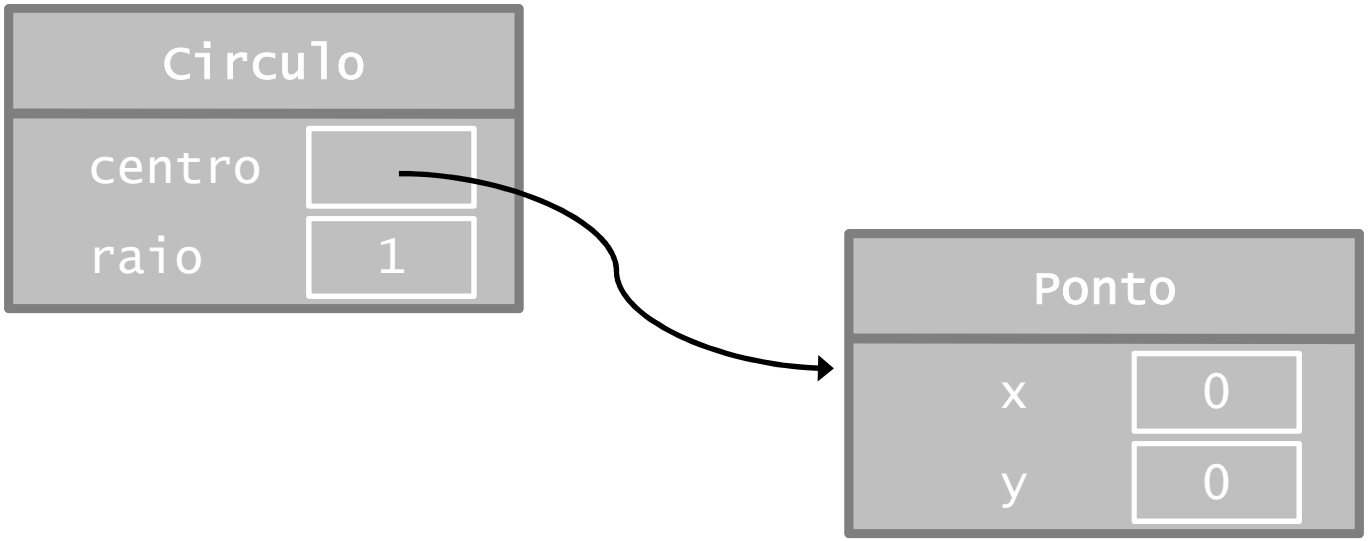
Composição de Classes (has-a)



Composição de Classes (has-a)



Composição de Classes (has-a)



Composição de Classes (has-a)

```
public class Ponto {
    private int x;
    private int y;

    public Ponto ()  { this(0,0); }          //Construtor por omissão
    public Ponto ( int x, int y ){           //Construtor com inicialização de atributos
        this.x = x; this.y = y;
    }
    public int getX (){                      //seletor de x
        return x;
    }
    public void setX( int x){                //modificador de x
        this.x = x;
    }
    public int getY (){                      //seletor de y
        return y;
    }
    public void setY ( int y ){              //modificador de x
        this.y = y;
    }
    public void setXY ( int x, int y ){      //modificador de x e y em simultâneo
        this.x = x; this.y = y;
    }
}
```

Composição de Classes (has-a)

```
public class Circulo {

    private static final double PI = 3.14159;                    //Constante
    private int raio;
    private Ponto centro;                                        //Circulo has-a Ponto

    //Não há construtor vazio: é "difícil" definir um valor, por omissão, para o raio

    public Circulo(int raio) {
        this(raio, 0, 0);
    }

    public Circulo(int raio, Ponto centro) {
        this.raio = raio;
        this.centro = new Ponto(centro.getX(), centro.getY());   //Circulo has-a Ponto
                                                                 //Porque não a atribuição directa?
    }

    public Circulo(int raio, int x, int y) {
        this.raio = raio;
        this.centro = new Ponto(x, y);                           //Circulo has-a Ponto
    }
```

Composição de Classes (has-a)

```
    public int getRaio() {
        return raio;
    }

    public void setRaio(int raio) {
        this.raio = raio;
    }

    public void setCentro(Ponto centro) {
        this.centro.setX(centro.getX());
        this.centro.setY(centro.getY());            //Porque não a atribuição directa?
    }

    public Ponto getCentro() {
        return new Ponto(centro.getX(), centro.getY());            //Porquê a cópia?
    }

    public double getArea() {
        return PI * raio * raio;
    }

    public double getPerimetro() {
        return 2 * PI * raio;
    }
}
```

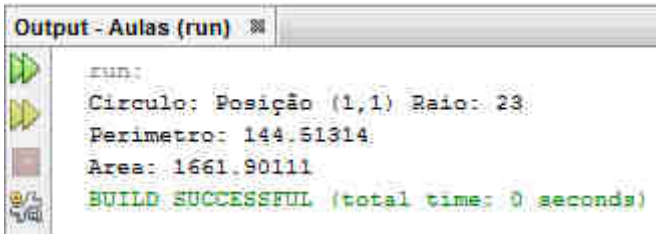
Composição de Classes (has-a)

```
public class Programa {
   public static void main ( String[] args ) {

      Circulo circulo1 = new Circulo(23, 1, 1);

      System.out.println ( "Circulo: Posição (" + circulo1.getCentro().getX()
                  + "," + circulo1.getCentro().getY() + ") " + "Raio: " +
                  circulo1.getRaio() );

      System.out.println ( "Perimetro: " + circulo1.getPerimetro() +
                           "\nArea: " + circulo1.getArea()  );
   }
}
```



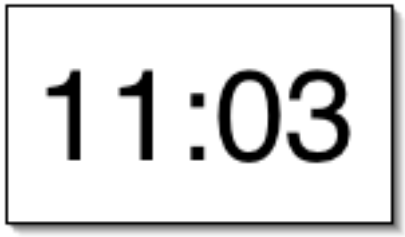
Resumindo

- Composição de classes – “has-a”
  - Atributos de uma classe são objetos de outras classes
  - Reutilização de código no seu melhor!

Composição de Classes - Exemplo

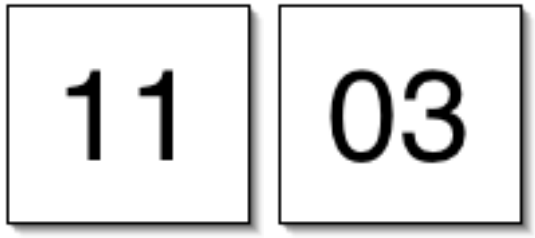
Um relógio digital

Modularização



Um mostrador de 4 dígitos?

Ou dois mostradores de 2 dígitos?



Composição de Classes - Exemplo

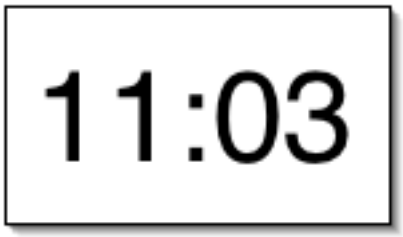
Implementação de MostradorNumero


```
public class MostradorNumero
{
    private int limite;
    private int valor;

    Construtor e métodos omitidos...
}
```

Composição de Classes - Exemplo

Implementação de MostradorRelogio




```
public class MostradorRelogio
{
    private MostradorNumero horas;
    private MostradorNumero minutos;

    Construtor e métodos omitidos...
}
```

Composição de Classes - Exemplo

Implementação de MostradorNumero

```
public class MostradorNumero
{
    private int limite;
    private int valor;

    public MostradorNumero(int limiteMaximo)
    {
        limite = limiteMaximo;
        valor = 0;
    }
```

Composição de Classes - Exemplo

Implementação de MostradorNumero

```
public int getValor()
{
    return valor;
}

public void incrementar()
{
    valor = (valor + 1) % limite;
}
```

Composição de Classes - Exemplo

Implementação de MostradorNumero

```
//@Override
//public String toString()
public String obterValorMostrador()
{
    if (valor < 10) {
        return "0" + valor;
    }
    else {
        return "" + valor;
    }
}
```

Implementação de MostradorRelogio

```
public class MostradorRelogio
{
    private MostradorNumero horas;
    private MostradorNumero minutos;

    public MostradorRelogio()
    {
        horas = new MostradorNumero(24);
        minutos = new MostradorNumero(60);
    }
```

Composição de Classes - Exemplo

Implementação de MostradorRelogio

```
public void incrementar()
{
    minutos.incrementar();
    if(minutos.getValor() == 0) {
        // “deu a volta”!
        horas.incrementar();
    }
}
```

Composição de Classes - Exemplo

Implementação de MostradorRelogio

```
//@Override
//public String toString()
public String obterValorMostrador()
{
    return horas.obterValorMostrador() +
           ":" +
           minutos.obterValorMostrador();
}
```

Leitura Complementar

- Composição
  - Páginas 93 - 108
- Capítulo 3
  - Páginas 65 a 112