

Programação Orientada a Objeto: Introdução

Professor: Adonai Estrela Medrado

Data: 22/07/2008

Programação

- ◆ Programação
 - ◆ É o ato de fazer um programa.
 - ◆ Um programa é “conjunto de ordens dadas a um computador para a realização de um determinado processo” (LANCHARRO, LOPEZ e FERNANDEZ, 1991, p. 2).
 - ◆ Envolve o uso de um código (linguagem).
 - ◆ Constitui somente **uma** das etapas no desenvolvimento de *software*.
 - ◆ Preocupação com a qualidade deve apresentar-se durante todas as etapas de desenvolvimento.

Programação Orientada a Objeto: Destques

- ◆ Programação orientada a objeto não é sinônimo de qualidade de *software* (STROUSTRUP, 1995).
- ◆ Importante no aprendizado da orientação a objeto:
 - ◆ “A aplicação impensada de técnicas que são eficazes em uma linguagem em outra conduz tipicamente a código mal estruturado, de baixo desempenho e difícil manutenção.” (STROUSTRUP, 2000, p. 18).

Programação Orientada a Objeto: Definição

- ◆ Segundo Stroustrup (1995), uma linguagem é orientada a objeto se, e somente se, suporta:
 - ◆ Abstração
 - ◆ Herança
 - ◆ Polimorfismo em tempo de execução
- ◆ Critério adicional da pureza:
 - ◆ As classes devem ser a única unidade de programação (MEYER, 1997).

Abstração – Classes e Objetos

◆ Classe

- ◆ Ponto central da orientação a objeto.
- ◆ Definição: uma classe é um **tipo abstrato de dado** equipado com implementação parcial ou total. (MEYER, 1997, p. 165).

◆ Objeto

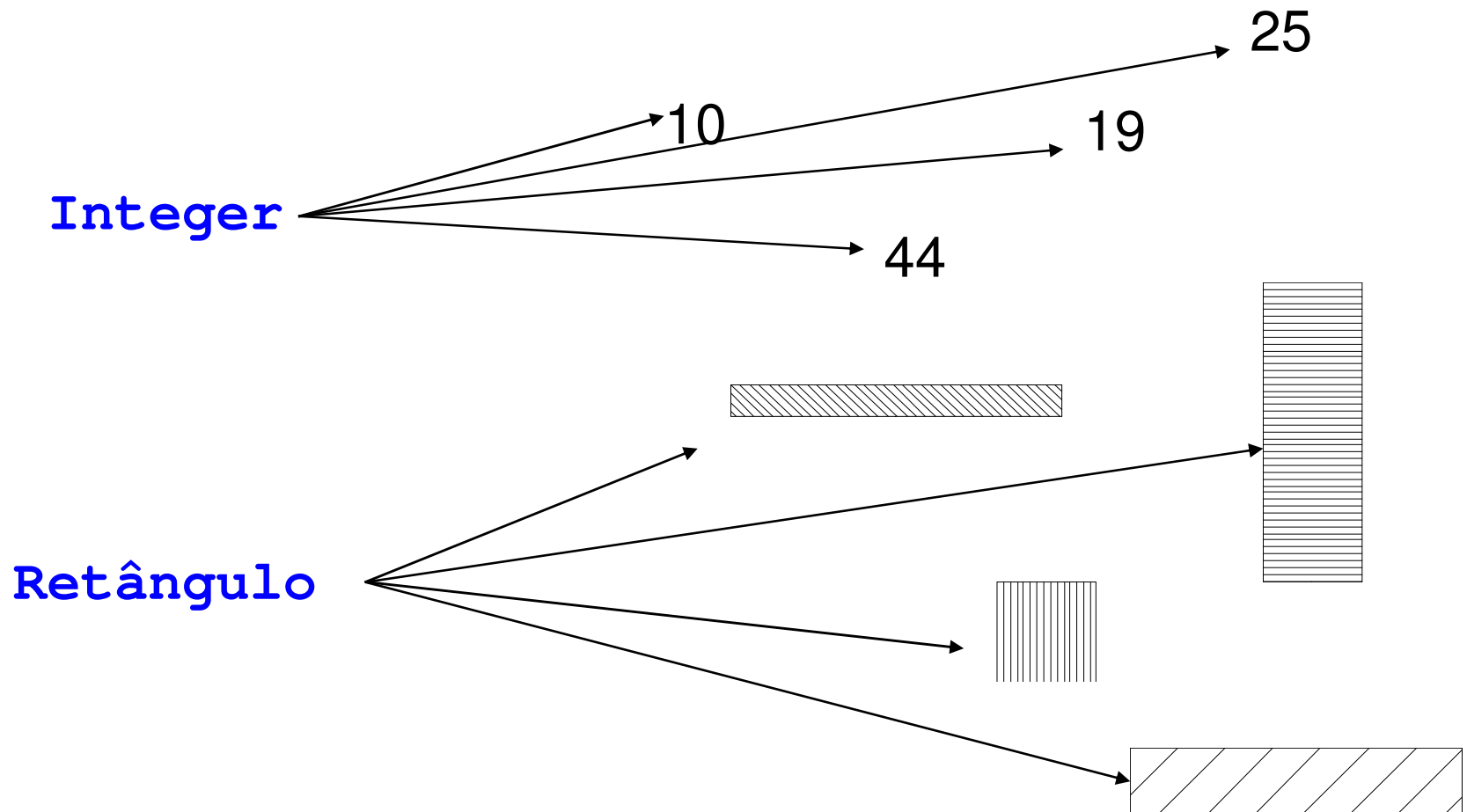
- ◆ Definição: instância **em tempo de execução** de alguma classe (MEYER, 1997, p. 218).

Abstração – Classes e Objetos

Exemplo Visual

Tipos de Dado

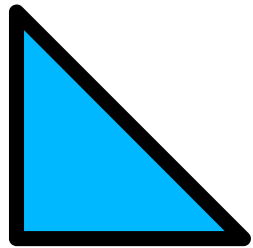
Instâncias



Herança

Exemplo Visual

- ◆ Uma classe A é descendente de B se herda direta ou indiretamente características de B. (MEYER, 2000).



Triângulo



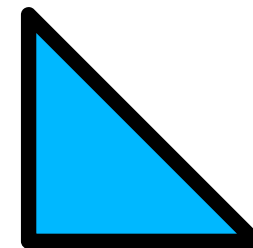
Retângulo

Compartilham
atributos de uma
Figura
Geométrica como
cor de borda,
preenchimento e
área.

Polimorfismo

Exemplo Visual

Área Triângulo
(base x altura) / 2



Solicita área da
Figura Geométrica



Descendente de
Figura Geométrica

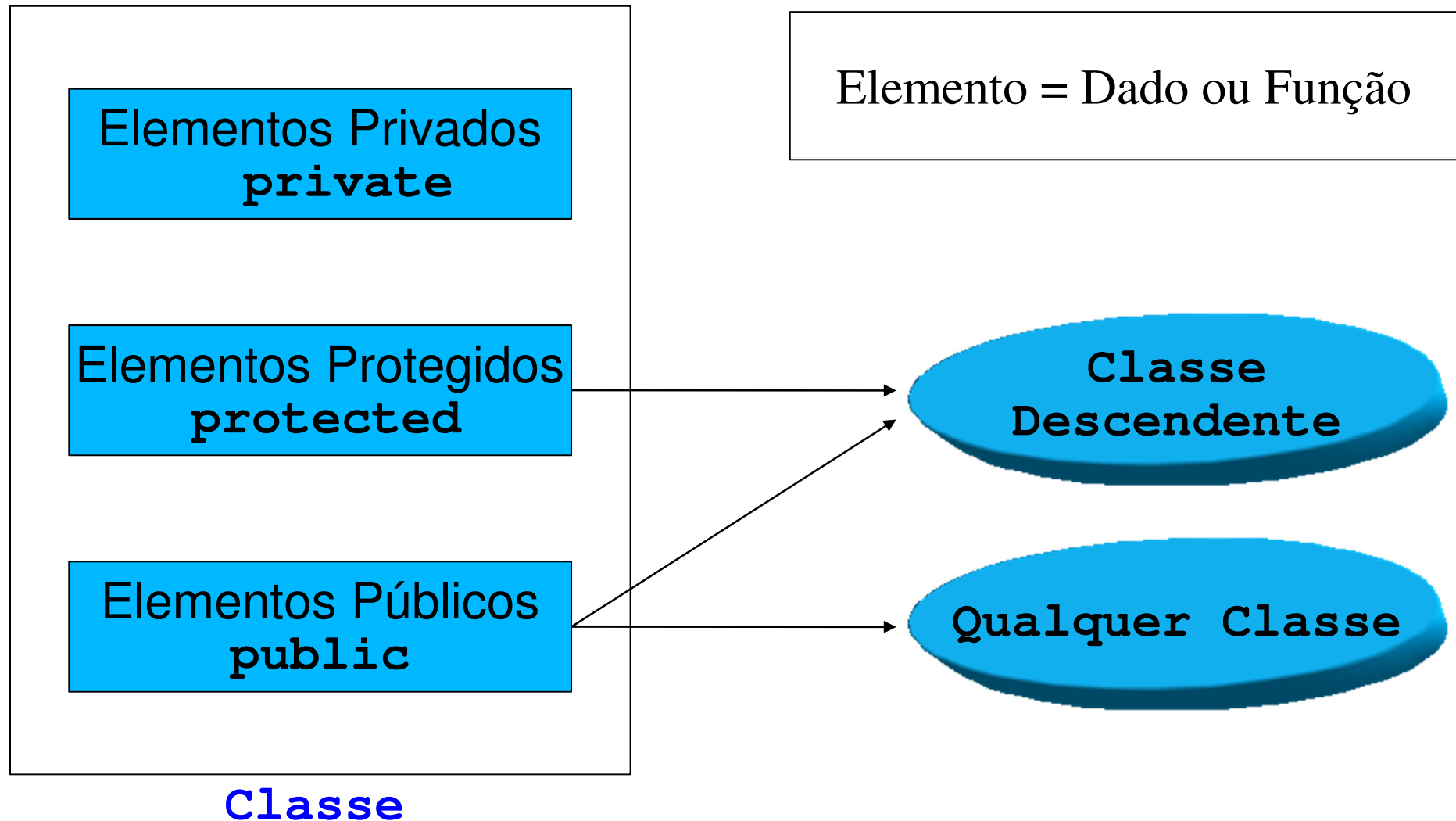


Recurso de
polimorfismo
retorna a área do
Triângulo ou do
Quadrado
conforme o caso.



Área Retângulo
altura x largura

Principais Modificadores de Acesso



```
class Triangulo extends FiguraGeometrica {
```

Campos

```
    private double altura;  
    private double base;
```

Métodos

```
    public double getAltura () {  
        return altura;  
    }
```

```
    public void setAltura (double valor) {  
        altura = valor;  
    }
```

```
    public double getBase () {  
        return base;  
    }
```

```
    public void setBase (double valor) {  
        base = valor;  
    }
```

```
    public double calcularArea() {  
        return (getAltura() * getBase()) / 2;  
    }
```

Construtor

```
    public Triangulo() {  
        setAltura(1);  
        setBase(1);  
    }
```

```
}
```

```
class Retangulo extends FiguraGeometrica {
```

Campos

```
private double altura;  
private double largura;
```

Métodos

```
public double getAltura () {  
    return altura;  
}
```

```
public void setAltura (double valor) {  
    altura = valor;  
}
```

```
public double getLargura () {  
    return largura;  
}
```

```
public void setLargura (double valor) {  
    largura = valor;  
}
```

```
public double calcularArea() {  
    return getAltura() * getLargura();  
}
```

Construtor

```
public Retangulo() {  
    setAltura(1);  
    setLargura(1);  
}
```

```
}
```

```
abstract class FiguraGeometrica {
```

Campos

```
{  
  private int corPreenchimento;  
  private int corBorda;  
}
```

Métodos

```
{  
  public int getCorPreenchimento () {  
    return corPreenchimento;  
  }  
  public void setCorPreenchimento (int valor) {  
    corPreenchimento = valor;  
  }  
  public double getBorda () {  
    return corBorda;  
  }  
  public void setBorda (int valor) {  
    corBorda = valor;  
  }  
  public abstract double calcularArea();  
}
```

Criando e Acessando Objetos

A palavra **new** é utilizada na criação um objeto.

```
FiguraGeometrica minhaFigura;  
int numeroDecisor;  
  
...  
  
if (numeroDecisor % 2 == 0)  
    minhaFigura = new Triangulo();  
else  
    minhaFigura = new Retangulo();  
  
System.out.print (  
    minhaFigura.calcularArea ());
```

Pontos Mais Importantes

- ◆ Aprender a programação orientada a objeto não tem necessariamente como consequência um aumento na qualidade do código.
- ◆ A programação é somente uma das fases do desenvolvimento de *software*.
- ◆ Classes podem definir a acessibilidade a seus elementos (**public**, **private**, **protected**).
- ◆ Para se criar um objeto utiliza-se da palavra chave **new**.
- ◆ Sugestão de pesquisa: Padrões de Projeto (*Design Patterns*).

Referências

- ◆ GAMMA, Erich e et al. **Padrões de projeto**: soluções reutilizáveis de software orientado a objeto. Porto Alegre: Bookman, 2000.
- ◆ LANCHARRO, Eduardo Alcalde, LOPEZ, Miguel Garcia e FERNANDEZ, Salvador Peñuelas **Informática básica**. São Paulo: Makron Books, 1991.
- ◆ MEYER, Bertrand. **Object-oriented software construction**. New Jersey: Prentice Hall PTR, 1997.
- ◆ STROUSTRUP, Bjarne. **Why C++ is not just an Object-Oriented Programming Language**. 1995. Disponível em: <<http://www.research.att.com/~bs/oopsla.pdf>> Acesso: 20/07/2008
- ◆ STROUSTRUP, Bjarne. **A linguagem de programação C++**. Porto Alegre: Bookman, 2000.