
Newton
Quem se prepara, não para.

Programação Orienta a Objetos

3º período

Professora: Michelle Hanne

Sumário

- ✓ Classes Abstratas e Interface

Abstração

Newton
Quem se prepara, não para.

- Em uma hierarquia, quanto mais alta a classe, mais abstrata é sua definição.
  - A classe Animal apresenta o método locomover(), mas ela não tem como implementar este método pois não sabe o tipo de animal que está tratando.
- Java permite definir métodos sem implementá-los!
- Métodos Abstratos:
  - Não possui corpo (implementação).
  - Apresenta apenas a definição seguida de ";"
  - Apresenta o modificador *abstract*.

```
public abstract class Animal {
  public int peso;
  public abstract void locomover();
}
```

  - Se uma classe apresentar pelo menos um método abstrato, ela deve ser declarada como *abstract*.

Abstração

Newton
Quem se prepara, não para.

- Classes abstratas não podem ser instanciadas!
  - São geralmente utilizadas como superclasses (são estendidas).
  - Uma subclasse de uma classe abstrata pode ser instanciada se ela sobrepor todos os métodos abstratos e fornecer implementação para cada um deles.
  - Se a subclasse não implementar *todos* os métodos abstratos da superclasse, ela também terá que ser abstrata.

```
public abstract class FiguraGeometrica {
  public abstract double area();
  public abstract double perimetro();
}

public class Retangulo extends FiguraGeometrica {
  protected double w, h;
  public Retangulo() { this(0.0,0.0); }
  public Retangulo(double l, double a) { w = l; h = a; }
  public double area() { return w*h; }
  public double perimetro() { return 2*w*h; }
}
```

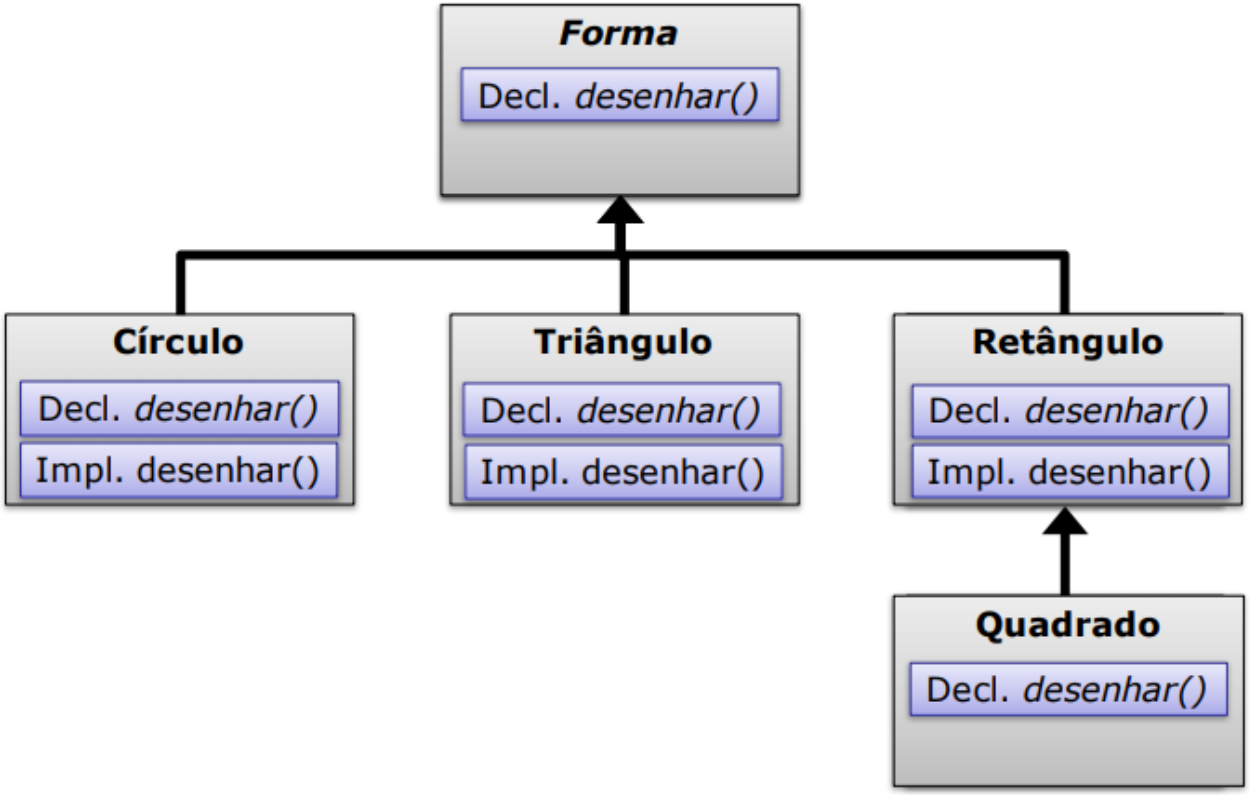
Abstração

Newton
Quem se prepara, não para.

- Não permitem criação de instâncias (objetos):
- Um método abstrato não possui implementação, portanto não pode ser chamado.
- Para ser útil, deve ser estendida:
  - **Suas subclasses devem implementar o método ou declararem-se como abstratas.**
  - **Servem para definir interfaces e prover algumas implementações comuns.**

Abstração

Newton
Quem se prepara, não para.



Interface

Newton
Quem se prepara, não para.

- Se uma classe abstrata possui apenas métodos abstratos, é melhor usar uma *interface*.
  - pode haver herança de comportamento de mais de uma (super)classe através das interfaces.
  - Interface NÃO é Classe
  - Especifica operações sem implementá-las.
- Componentes de Interface:
  - Métodos: Todos os métodos são implicitamente públicos e abstratos.
  - Constantes: São implicitamente públicas e estáticas.
- Também não podem ser instanciadas.

Interface

```
public interface Desenho {
  public void novaCor(Color c);
  public void novaPosicao(double x, double y);
  public void desenha(DrawWindow dw);
}
```

- Código armazenado em arquivo `Desenho.java`

```
public class RetanguloDesenhavel extends Retangulo
                         implements Desenho {
  private Color c;
  private double x, y;
  public RetanguloDesenhavel(double l, double a) {
    super(a,l);
  }
  public void novaCor(Color c) { this.c = c; }
  public void novaPosicao(double x, double y) {
    this.x = x; this.y = y;
  }
  public void desenha(DrawWindow dw) {
   dw.drawRect(x, y, w, h, c);
  }
}
```

Interface

Newton
Quem se prepara, não para.

- Uma classe abstrata é pura quando:
  - Possui métodos abstratos;
  - Não possui métodos concretos;
  - Não possui atributos (não-static).
- Java oferece a palavra reservada interface:
  - Cria uma classe abstrata pura;
  - Chamaremos pelo nome de interface;

Interface

Newton
Quem se prepara, não para.

```
interface Forma {
  int x = 10;
  void desenhar();
}

interface Forma {
  public static int x = 10;
  public abstract void desenhar();
}
```

Definições equivalentes

Por que isso existe? Só pra complicar a linguagem?

Exemplos

Newton
Quem se prepara, não para.

- FormaGeometrica = Uso de Abstract
- FormaGeometricaInterface = Uso de Interface
- ContaCorrente = Uso de Abstract

Referências

SILVA, Fabricio Machado da. **Paradigmas de programação**. SAGAH, 2019. ISBN digital: 9788533500426

Barnes, David e Kölling, M. **Programação Orientada a Objetos com Java**. São Paulo: Pearson Prentice Hall, 2004.

Deitel, H. M.; Deitel, P. J. **Java - Como Programar**. 6. ed. Prentice-Hall, 2005. Capítulo 4 e 5.

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: MacGraw Hill, 2002.