

# Classes Abstratas

P. O. O.

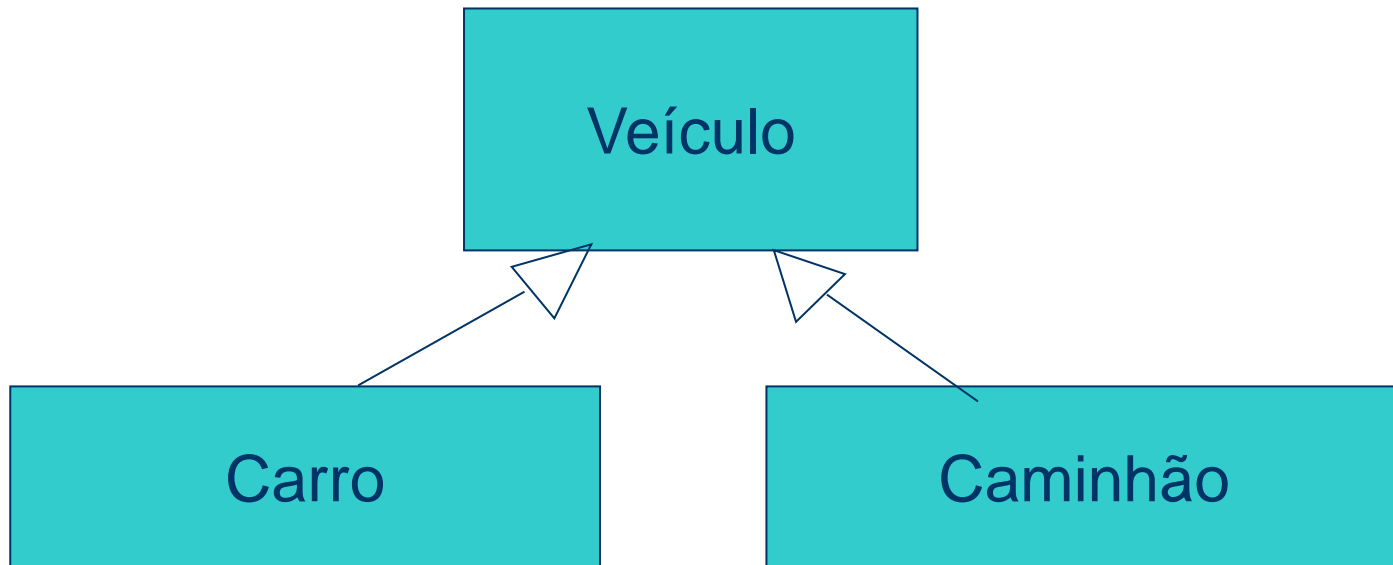
Prof. Grace

---

# Aulas Anteriores

- Herança:
  - Criação de uma nova classe a partir de uma classe existente, permitindo melhor organização e, eventualmente, reuso de código.
  - Adota um relacionamento hierárquico entre classes.
- Polimorfismo:
  - Princípio pelo qual várias classes **derivadas** de uma mesma **superclasse** podem invocar métodos que têm a **mesma identificação** (assinatura) mas **comportamentos distintos**.

# Hierarquia – Veículo

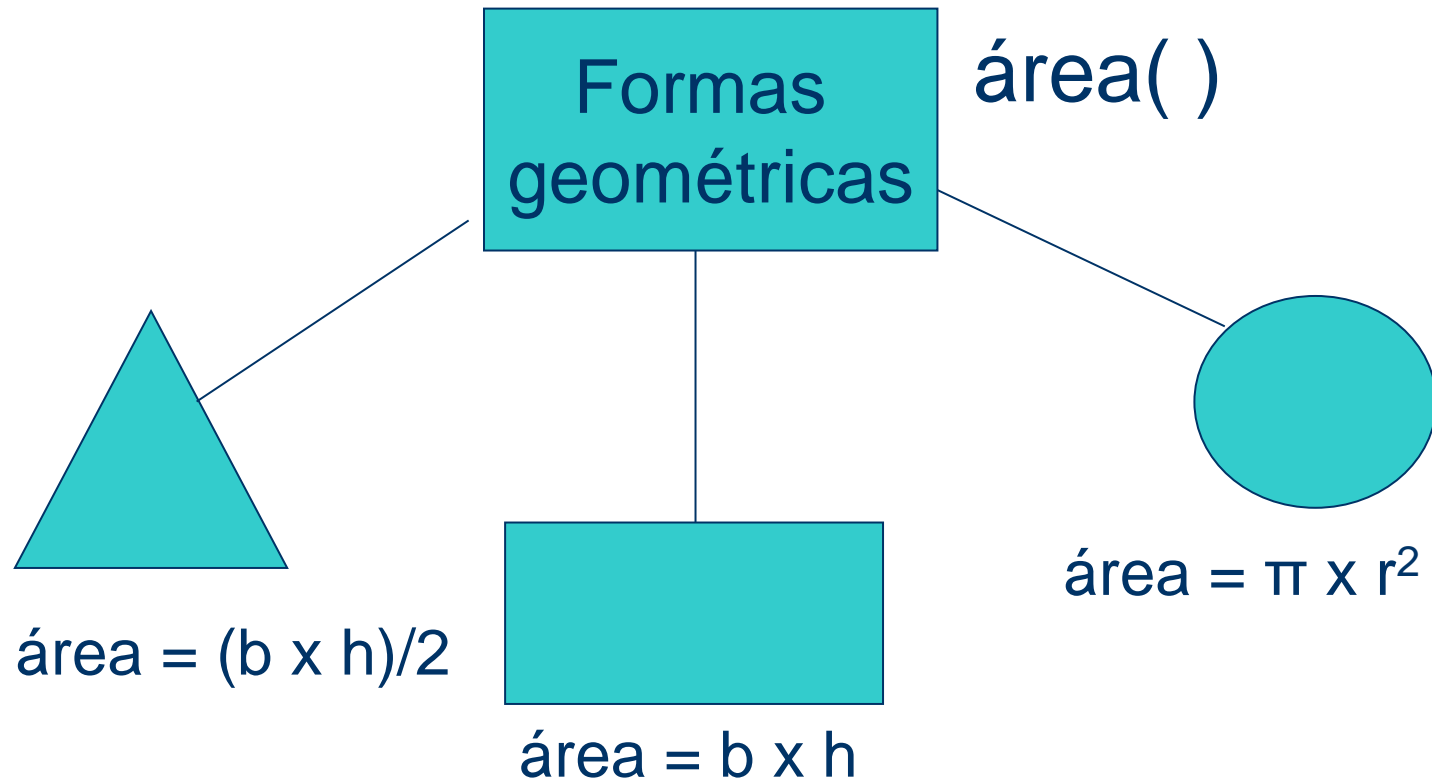


- Podemos afirmar que:
  - Todo Carro “é um” Veículo, mas nem todo Veículo é um carro.
  - Todo Caminhão “é um” Veículo.

# Classe Abstrata

- Classes demasiadamente **gerais** para criar objetos reais;
- Utilizadas como **superclasses** para subclasses concretas;
- Declara atributos e comportamentos (métodos) comuns das várias classes em uma hierarquia;
- Contém um ou mais **métodos abstratos**;
- Utiliza palavra **abstract** para declarar uma classe abstrata ou um método abstrato;
- Tentar **instanciar** um objeto de uma classe abstrata é um erro de compilação.

# Exemplo - Formas geométricas



# Classe Abstrata: Forma Geométrica

- Atributos básicos
  - Número de lados
- Métodos
  - Set/ get número de lados: métodos não abstratos
  - Área: **método abstrato**
  - Imprime(): método não abstrato
  - Precisa construtor?

# Classe FormaGeometrica

```
1 public abstract class FormaGeometrica
2 {
3     protected int lados; ← Atributo protected
4     public void setLados(int lados)
5     {
6         if (lados<0) this.lados = 0;
7         else this.lados = lados;
8     }
9     public abstract double area();
10
11     public void imprime() ← Método concreto
12     {
13         System.out.println("Numero de lados: "+ lados);
14     }
15 }
```

# Classe concreta X superclasse abstrata

- Classes abstratas contêm **um ou mais** métodos abstratos.
- No caso da forma geométrica: método **area( )** é um exemplo de método abstrato.
- Todas as subclasses **concretas** devem sobrescrever **todos** os métodos **abstratos** herdados. No caso do círculo, devemos sobrescrever o método **area( )**.



# Classe Circulo com Método Area( )

```
1 import java.lang.Math; ← Usa classe Math
2 public class Circulo extends FormaGeometrica
3 {
4     private double raio;
5     public void setRaio(double raio)
6     {
7         if (raio<0) this.raio = 0;
8         else this.raio = raio;
9     }
10
11     public Circulo (double raio)
12     {
13         setRaio(raio);
14     }
15
16     public double area() ← Implementa Area( )
17     {
18         return (Math.PI * Math.pow(raio, 2));
19     }
20 }
```

Implemente

# Classe TesteFormas

```
public class TesteFormas
{
    public static void main (String args[])
    {
        FormaGeometrica f = new FormaGeometrica();
        f.imprime();

        Circulo c = new Circulo(5);
        c.imprime();
    }
}
```

**Funciona?**

# Alterações

- Classes abstratas não podem ser instanciadas.
  - Corrigir código do teste
- O que o método **imprime()**, quando invocado a partir de um objeto circulo deve imprimir na tela?
- Como imprimir raio junto ao número de lados?
- Altero em Circulo ou FormaGeometrica?
- Redefinir método `imprime( )` na classe circulo!

# Classe Circulo - Método imprime( )

```
1 import java.lang.Math;
2 public class Circulo extends FormaGeometrica
3 {
4     private double raio;
5     public void setRaio(double raio)
6     {
7         if (raio<0) this.raio = 0;
8         else this.raio = raio;
9     }
10    public Circulo (double raio)
11    {
12        setRaio(raio);
13    }
14    public double area()
15    {
16        return (Math.PI * Math.pow(raio, 2));
17    }
18    public void imprime() ← sobrescreve imprime( )
19    {
20        System.out.println("Dados do Circulo");
21        super.imprime();
22        System.out.println("Raio: "+ raio);
23    }
24 }
```

## Mais alterações...

- Como imprimir área da forma geométrica instanciada? No caso, a área do círculo...
  - Altero **método imprime( )** na classe Círculo?
  - Ou na FormaGeometrica?
- Redefinir método imprime( ) incluindo área.

# Alterando a superclasse

```
1 public abstract class FormaGeometrica
2 {
3     protected int lados;
4
5     public void setLados(int lados)
6     {
7         if (lados<0) this.lados = 0;
8         else this.lados = lados;
9     }
10    public abstract double area();
11
12    public void imprime()
13    {
14        System.out.println("Numero de lados: " + lados);
15        System.out.printf("Area: %.2f\n", area());
16    }
17 }
```

## Exemplo 02 – Subclasse Retangulo

- Atributos básicos
  - Base
  - Altura
- Métodos
  - Sets/ Gets
  - Construtor (base, altura)
  - area( )
  - imprime( )

# Classe Retangulo

```
1 public class Retangulo extends FormaGeometrica
2 {
3     private double base, altura;
4     public Retangulo(double base, double altura)
5     {
6         super.setLados(4);
7         setBase(base);
8         setAltura(altura);
9     }
10
11     public void setBase(double base)
12     {
13         if (base<0) this.base = 0;
14         else this.base = base;
15     }
16
17     public void setAltura(double altura)
18     {
19         if (altura<0) this.altura = 0;
20         else this.altura = altura;
21     }
```



## Classe Retangulo (cont.)

```
23     public double area()
24     {
25         return base * altura;
26     }
27
28     public void imprime()
29     {
30         System.out.println("Dados do Retangulo");
31         super.imprime();
32         System.out.println("Base: " + base);
33         System.out.println("Altura: " + altura);
34     }
35 }
```

# Teste Formas

```
1 public class TesteFormas
2 {
3     public static void main(String args[])
4     {
5         Circulo c = new Circulo(5);
6         c.imprime();
7         System.out.println();
8
9         Retangulo r = new Retangulo(3, 4);
10        r.imprime();
11    }
12 }
```

## Atividade 21 – Classe abstrata

- Codifique a classe Triângulo
- Crie uma classe de teste com as 3 formas geométricas (Circulo, Retangulo e Triangulo) usando polimorfismo.

# Envio de Atividades

- As atividades desta aula devem ser enviadas pelo Teams