Overriding (Sobrescrita – “Animal”)

// Classe base

class Animal {

public void fazerSom() {

System.out.println("O animal faz um som.");

}

}

// Classe derivada

class Cachorro extends Animal {

@Override

public void fazerSom() {

System.out.println("O cachorro late: Au Au!");

}

}

// Classe derivada

class Gato extends Animal {

@Override

public void fazerSom() {

System.out.println("O gato mia: Miau!");

}

}

// Classe principal

public class Main {

public static void main(String[] args) {

Animal meuCachorro = new Cachorro();

Animal meuGato = new Gato();

meuCachorro.fazerSom(); // Saída: O cachorro late: Au Au!

meuGato.fazerSom(); // Saída: O gato mia: Miau!

}

}

Overloading (Sobrecarga – “Calculadora”)

class Calculadora {

// Método para somar dois inteiros

public int somar(int a, int b) {

return a + b;

}

// Método para somar dois números de ponto flutuante

public double somar(double a, double b) {

return a + b;

}

}

// Classe principal

public class Main {

public static void main(String[] args) {

Calculadora calc = new Calculadora();

int somaInteiros = calc.somar(5, 10);

double somaDoubles = calc.somar(5.5, 10.5);

System.out.println("Soma de inteiros: " + somaInteiros); // Saída: Soma de inteiros: 15

System.out.println("Soma de doubles: " + somaDoubles); // Saída: Soma de doubles: 16.0

}

}

Diagrama de classes - Polimorfismo

Overriding

Animal

+fazerSom()

Cachorro

+fazerSom()

Gato

+fazerSom()

Overloading

Calculadora

+somar(int, int)

+somar(double, double)

Classes abstratas

Exemplo 01 (“Veículo”)

// Classe abstrata

abstract class Veiculo {

// Método abstrato

public abstract void mover();

// Método abstrato

public abstract void parar();

}

// Classe concreta

class Carro extends Veiculo {

@Override

public void mover() {

System.out.println("O carro está se movendo.");

}

@Override

public void parar() {

System.out.println("O carro parou.");

}

}

// Classe concreta

class Bicicleta extends Veiculo {

@Override

public void mover() {

System.out.println("A bicicleta está se movendo.");

}

@Override

public void parar() {

System.out.println("A bicicleta parou.");

}

}

// Classe principal

public class Main {

public static void main(String[] args) {

Veiculo meuCarro = new Carro();

Veiculo minhaBicicleta = new Bicicleta();

meuCarro.mover(); // Saída: O carro está se movendo.

meuCarro.parar(); // Saída: O carro parou.

minhaBicicleta.mover(); // Saída: A bicicleta está se movendo.

minhaBicicleta.parar(); // Saída: A bicicleta parou.

}

}

Exemplo 02 (“Forma”)

// Classe abstrata

abstract class Forma {

// Método abstrato

public abstract double calcularArea();

}

// Classe concreta

class Circulo extends Forma {

private double raio;

public Circulo(double raio) {

this.raio = raio;

}

@Override

public double calcularArea() {

return Math.PI \* raio \* raio; // Área do círculo

}

}

// Classe concreta

class Quadrado extends Forma {

private double lado;

public Quadrado(double lado) {

this.lado = lado;

}

@Override

public double calcularArea() {

return lado \* lado; // Área do quadrado

}

}

// Classe principal

public class Main {

public static void main(String[] args) {

Forma meuCirculo = new Circulo(5);

Forma meuQuadrado = new Quadrado(4);

System.out.println("Área do círculo: " + meuCirculo.calcularArea()); // Saída: Área do círculo: 78.53981633974483

System.out.println("Área do quadrado: " + meuQuadrado.calcularArea()); // Saída: Área do quadrado: 16.0

}

}

Diagrama de classes – Classes Abstratas

|  |  |  |
| --- | --- | --- |
|  |  |  |

Exemplo 01

Veículo

+mover()

+parar()

Carro

+mover()

+parar()

Bicicleta

+mover()

+parar()

Exemplo 02

Forma

+calcularArea()

Quadrado

+lado()

+calcularArea()

Circulo

+raio()

+calcularArea()