G..Geetha

2320030487-S1

Simple Inheritance:

* Define a base class called Vehicle with properties like brand and year, and methods like startEngine() and stopEngine().Define a derived class called Car that extends Vehicle and adds properties like model and methods like honk().Create objects of both classes and demonstrate the use of their properties and methods.

class Vehicle {

String brand;

int year;

public Vehicle(String brand, int year) {

this.brand = brand;

this.year = year;

}

public void startEngine() {

System.out.println(brand + " engine started.");

}

public void stopEngine() {

System.out.println(brand + " engine stopped.");

}

}

class Car extends Vehicle {

String model;

public Car(String brand, int year, String model) {

super(brand, year); // Calling the constructor of the base class

this.model = model;

}

public void honk() {

System.out.println(brand + " " + model + " is honking: Beep Beep!");

}

}

public class Main {

public static void main(String args[]) {

Vehicle myVehicle = new Vehicle("GenericBrand", 2020);

System.out.println("Vehicle Brand: " + myVehicle.brand);

System.out.println("Vehicle Year: " + myVehicle.year);

myVehicle.startEngine();

myVehicle.stopEngine();

Car myCar = new Car("Toyota", 2021, "Camry");

System.out.println("\nCar Brand: " + myCar.brand);

System.out.println("Car Year: " + myCar.year);

System.out.println("Car Model: " + myCar.model);

myCar.startEngine();

myCar.honk();

myCar.stopEngine();

}

}

* Method Overriding

· Define a base class called Animal with a method makeSound(). Define a derived class called Dog that overrides the makeSound() method. Create objects of both classes and demonstrate method overriding.

class Animal {

public void makeSound() {

System.out.println("Some generic animal sound");

}

}

class Dog extends Animal {

@Override

public void makeSound() {

System.out.println("Bark")

}

}

public class Main {

public static void main(String[] args) {

Animal myAnimal = new Animal();

myAnimal.makeSound();

Dog myDog = new Dog();

myDog.makeSound();

Animal anotherDog = new Dog();

anotherDog.makeSound();

}

}

* Interface-Basics

· Define an interface Shape with methods draw() and getArea().

· Create a class Circle that implements Shape and provides implementations for draw() and getArea().

· Create objects of Circle and demonstrate the use of the interface methods

interface Shape {

void draw();

double getArea();

}

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public void draw() {

System.out.println("Draw a circle with radius: " + radius);

}

public double getArea() {

return Math.PI \* radius \* radius;

}

}

public class Main {

public static void main(String[] args) {

Circle myCircle = new Circle(5.0);

myCircle.draw();

System.out.println("The area of the circle is: " + myCircle.getArea());

}

}

* Implement multiple interfaces in a single class.

interface Printable {

void print();

}

interface Drawable {

void draw();

}

class Shape implements Printable, Drawable {

private String name;

public Shape(String name) {

this.name = name;

}

public void print() {

System.out.println("Printing the shape: " + name);

}

public void draw() {

System.out.println("Drawing the shape: " + name);

}

}

public class Main {

public static void main(String[] args) {

Shape myShape = new Shape("Circle");

myShape.print();

myShape.draw();

}

}

* Define two interfaces, Printable with a method print() and Scannable with a method scan().Create a class PrinterScanner that implements both interfaces and provides implementations for print() and scan().Create an object of PrinterScanner and demonstrate the use of both interface methods.

interface Printable {

void print();

}

interface Scannable {

void scan();

}

class PrinterScanner implements Printable, Scannable {

@Override

public void print() {

System.out.println("Printing the document...");

}

public void scan() {

System.out.println("Scanning the document...");

}

}

public class Main {

public static void main(String[] args) {

PrinterScanner myDevice = new PrinterScanner();

myDevice.print();

myDevice.scan();

}

}