Problem 1: Animal Sounds

Description: Create a class hierarchy for different animals, where each animal makes a different sound. Use method overriding to achieve polymorphism.

Requirements:

1. Create a base class Animal with a method makeSound().
2. Create derived classes Dog, Cat, and Cow, each overriding the makeSound() method to produce their respective sounds.
3. Implement a main method to demonstrate calling makeSound() on different animal objects stored in an Animal array.

Problem 2: Payment System

Description: Create a class hierarchy for different payment methods, where each method processes payment differently. Use method overriding to achieve polymorphism.

Requirements:

1. Create a base class Payment with a method processPayment().
2. Create derived classes CreditCardPayment, DebitCardPayment, and PayPalPayment, each overriding the processPayment() method to process payments in their respective ways.
3. Implement a main method to demonstrate calling processPayment() on different payment objects stored in a Payment array.

Problem 3: Drawing Shapes

Description: Create a class hierarchy for different shapes, where each shape has a different way of drawing itself. Use method overriding to achieve polymorphism.

Requirements:

1. Create a base class Shape with a method draw().
2. Create derived classes Circle, Rectangle, and Triangle, each overriding the draw() method to draw the respective shape.
3. Implement a main method to demonstrate calling draw() on different shape objects stored in a Shape array.

Problem 4: Employee Roles

Description: Create a class hierarchy for different types of employees, where each type of employee has a different way of working. Use method overriding to achieve polymorphism.

Requirements:

1. Create a base class Employee with a method work().
2. Create derived classes Manager, Developer, and Intern, each overriding the work() method to describe their respective work.
3. Implement a main method to demonstrate calling work() on different employee objects stored in an Employee array.

Problem 5: Vehicle Movements

Description: Create a class hierarchy for different types of vehicles, where each vehicle moves differently. Use method overriding to achieve polymorphism.

Requirements:

1. Create a base class Vehicle with a method move().
2. Create derived classes Car, Bicycle, and Boat, each overriding the move() method to describe their respective movements.
3. Implement a main method to demonstrate calling move() on different vehicle objects stored in a Vehicle array.