Implementation of Abstract Class using Access Modifiers

Aim

To implement an abstract class using `public`, `private`, and `protected` access modifiers and demonstrate inheritance and method overriding in Java.

Problem Statement

Create an abstract class named **Person** for a university management system. The class must contain the following instance variables:

- `personId` (type `int`)
- `name` (type `String`)
- `role` (type `String`)

The variables `personId` and `name` should be declared as **private**, and the variable `role` should be declared as **protected**. Include a constructor that initializes the `personId` and `name`. Provide **public getter and setter methods** for all variables, but ensure that `role` can only be modified inside subclasses using **protected access**.

Declare an abstract method named **calculateMonthlyAllowance()** that returns a `double` value.

Create three subclasses named **StudentPerson**, **StaffPerson**, and **ResearchPerson** that extend the `Person` class. Each subclass must implement the `calculateMonthlyAllowance()` method with different logic:

- StudentPerson → 2000
- StaffPerson → 8000
- ResearchPerson → 12000

Each subclass should set its own `role` value using the protected variable from the parent class.

Write a test application named **PersonDemo** that creates an array of at least six `Person` objects containing different combinations of `StudentPerson`, `StaffPerson`, and `ResearchPerson`. Display the `personId`, `name`, `role`, and monthly allowance for each object using the appropriate methods.