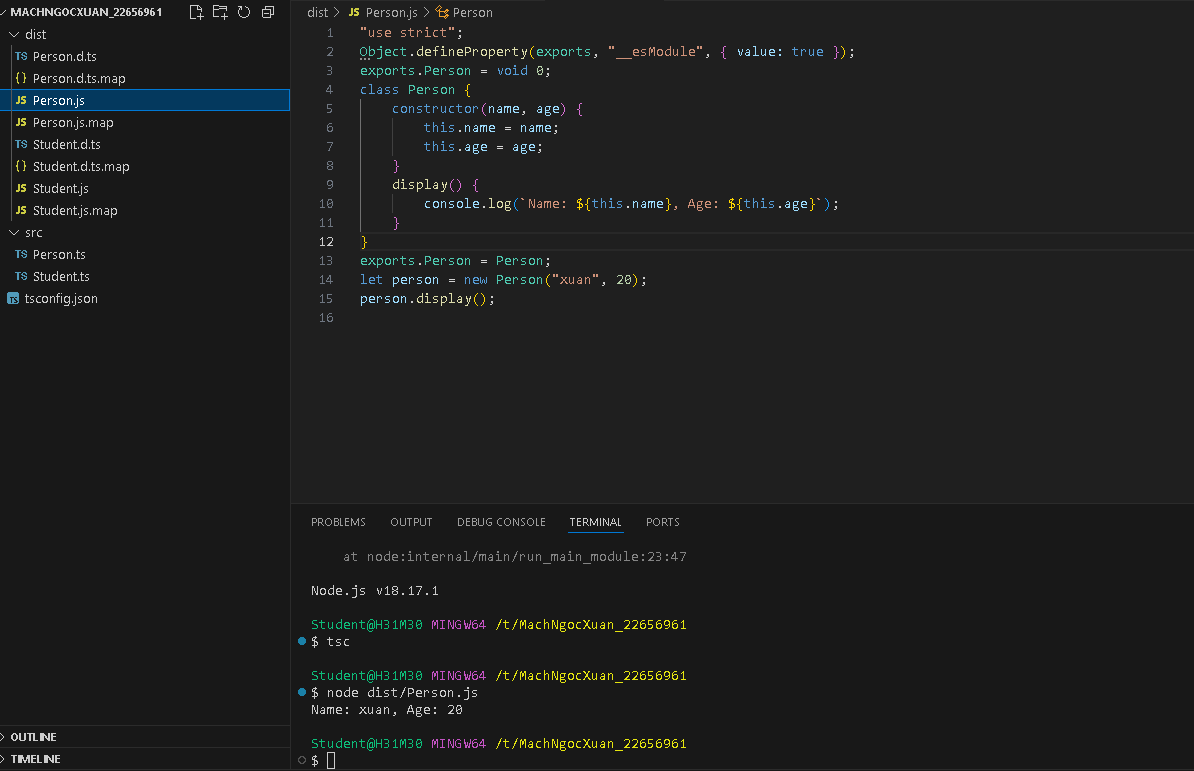
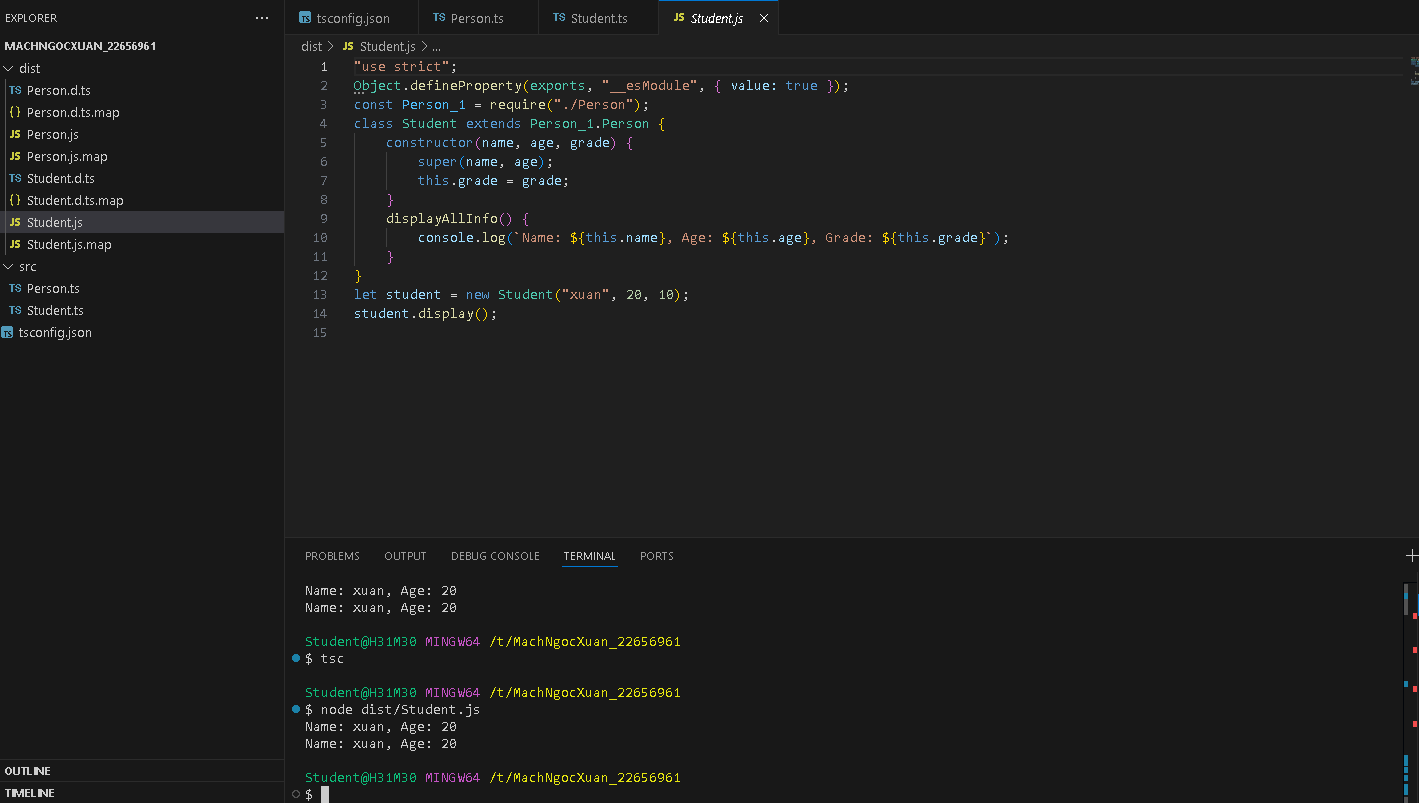
1. Create a class Person with attributes name and age. Write a method to display this information.

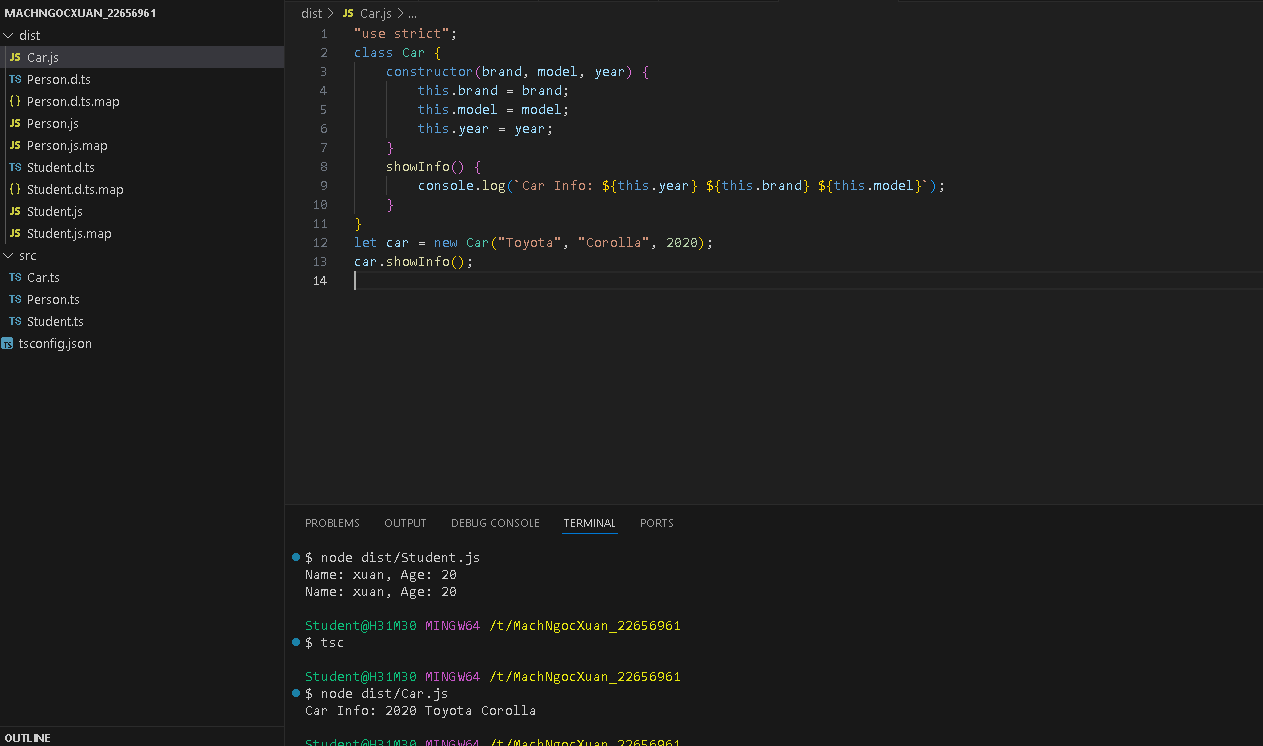


2. Write a class Student extending Person with an additional attribute grade. Add a method to

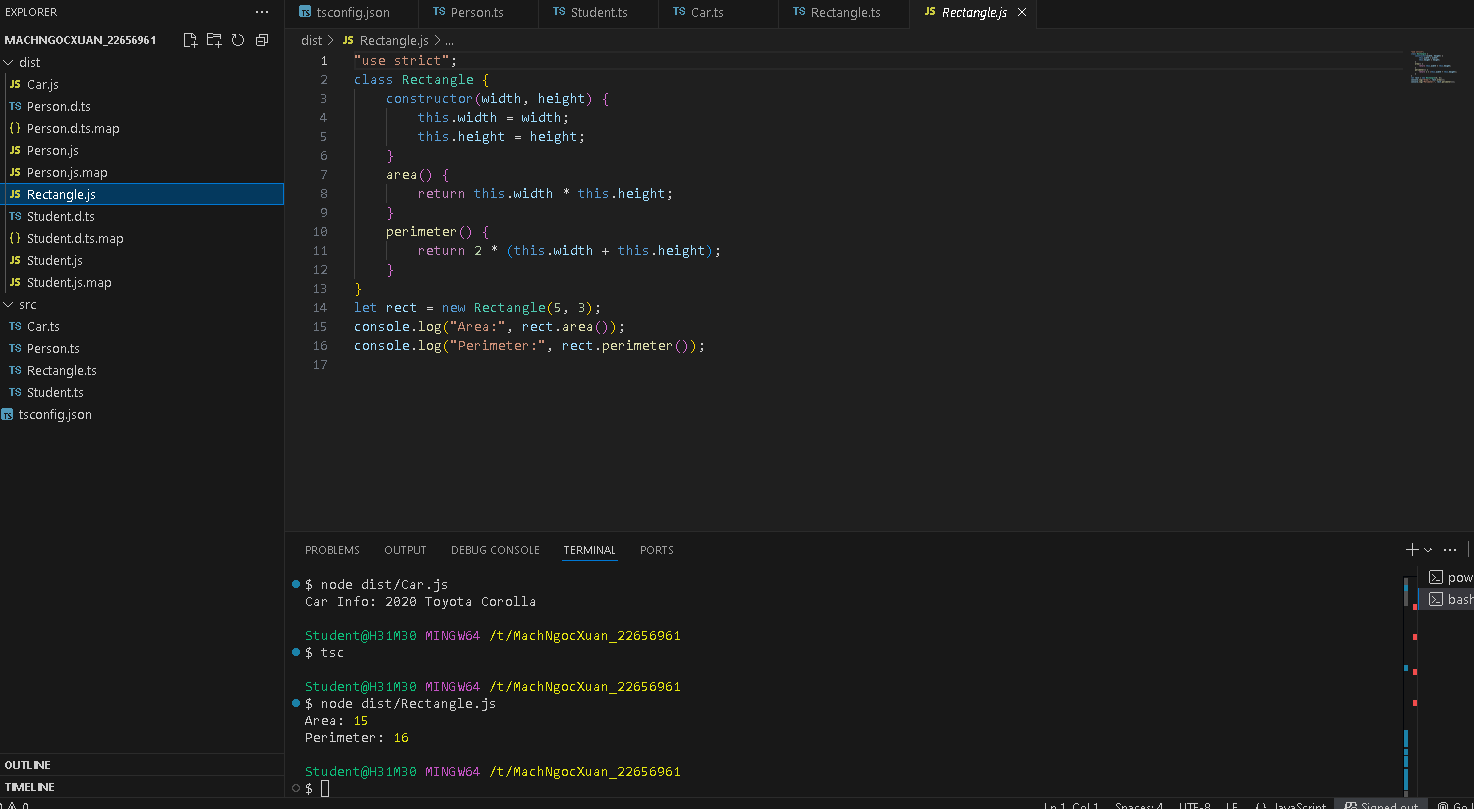
display all info.



3. Create a class Car with properties brand, model, year. Write a method to show car info.



4. Create a class Rectangle with width and height. Write a method to calculate area and perimeter.



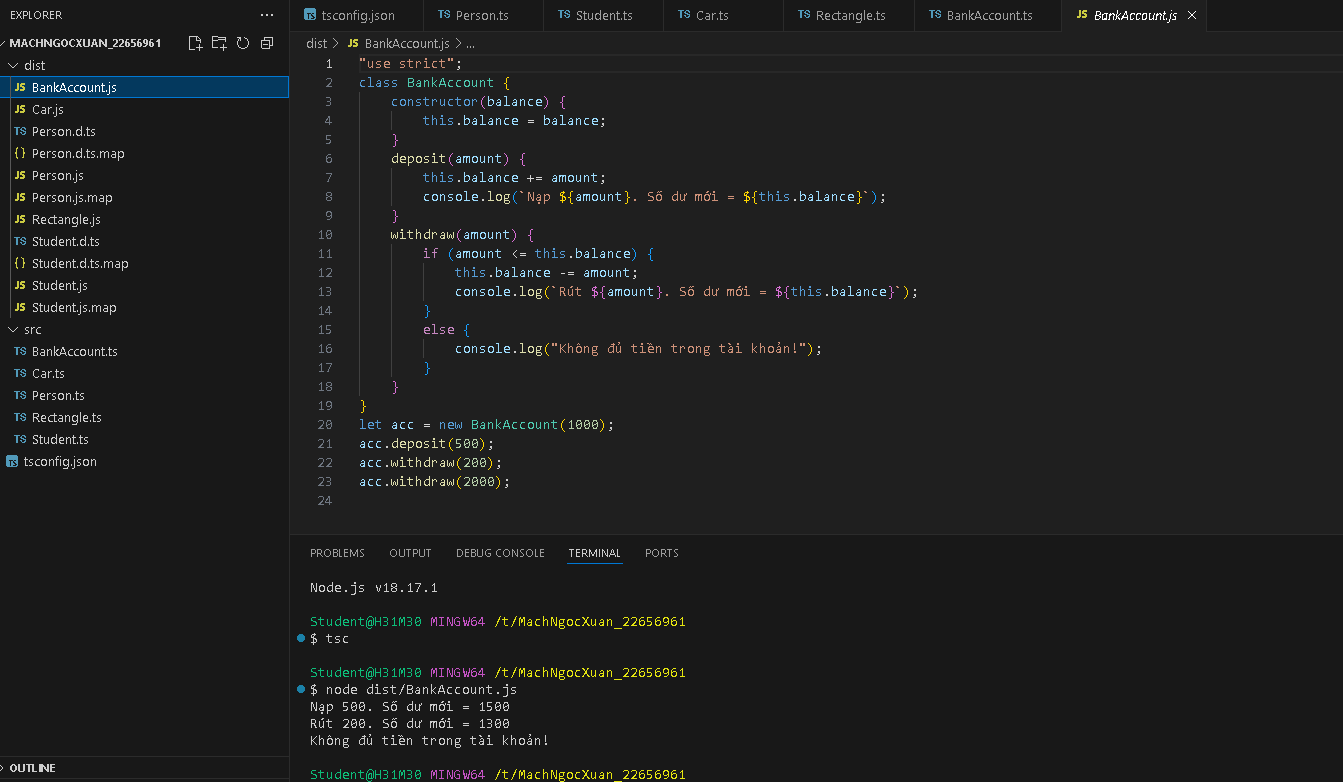
 width và height là thuộc tính (chiều rộng, chiều cao).

 constructor(width, height) là hàm khởi tạo

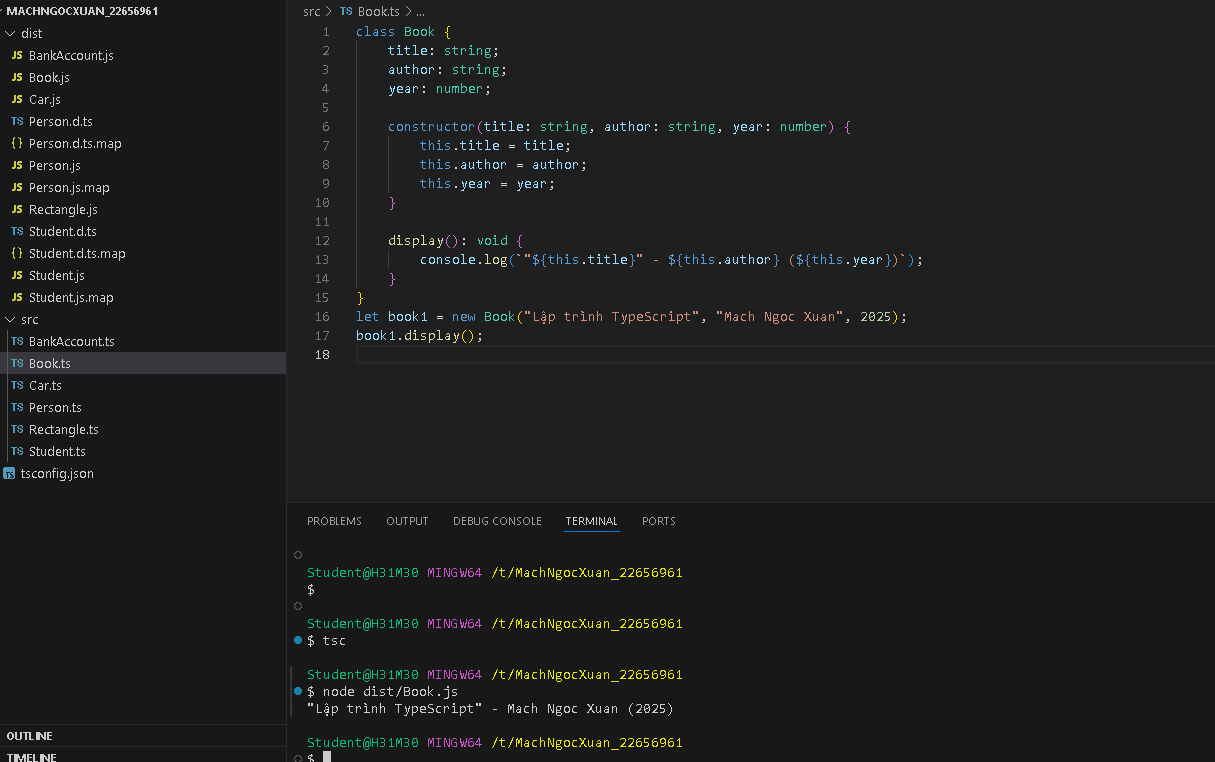
 area() là phương thức để tính diện tích: width×height

 perimeter() là phương thức để tính chu vi: Chu vi=2×(width+height)

5. Create a class BankAccount with balance. Add methods deposit() and withdraw().



6. Create a class Book with attributes title, author, year.

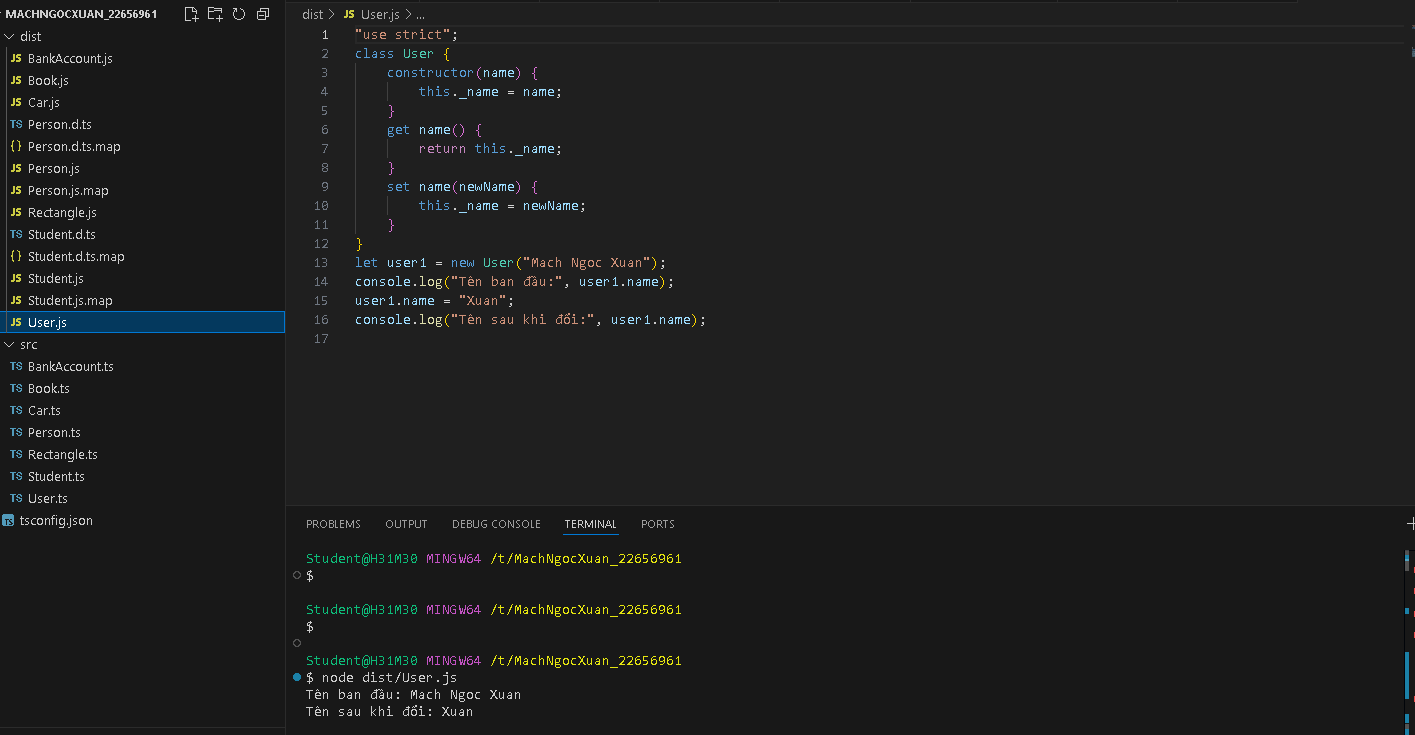


**- title, author, year** là thuộc tính của sách (tiêu đề, tác giả, năm xuất bản).

**- constructor(title, author, year)** là **hàm khởi tạo**, dùng để gán giá trị ban đầu cho các thuộc tính khi tạo sách mới.

**- display()** là **phương thức** in ra thông tin sách dưới dạng:

7. Write a class User with private property name and getter/setter.



 **private \_name**: thuộc tính riêng, chỉ dùng trong class.

 **constructor(name)**: hàm khởi tạo, gán tên ban đầu cho user.

 **get name()**: dùng để lấy giá trị \_name (đọc).

 **set name(newName)**: dùng để thay đổi giá trị \_name (ghi).

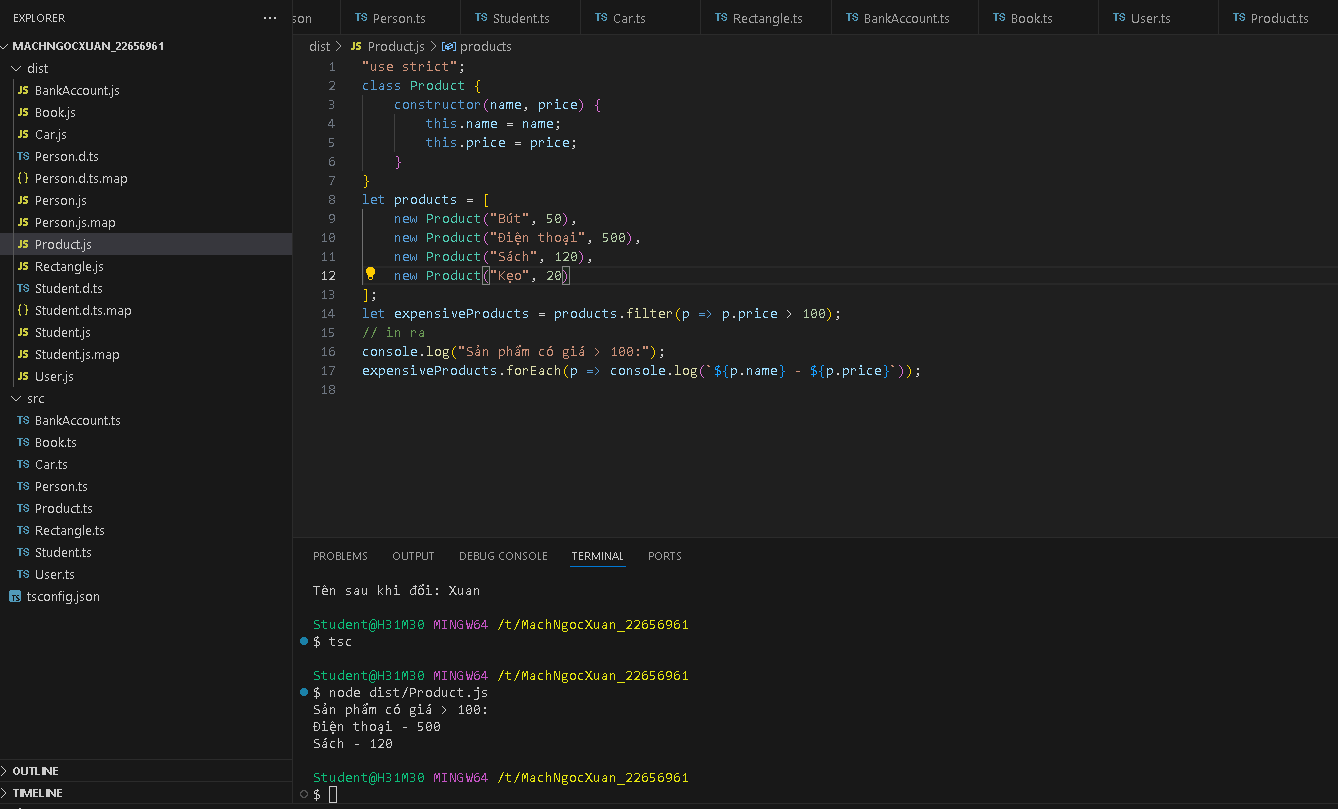
 **new User("Mach Ngoc Xuan")**: tạo user mới với tên ban đầu.

 **console.log(user1.name)**: in ra tên (dùng getter).

 **user1.name = "Nguyen Van A"**: đổi tên bằng setter.

8. Create a Product class with name, price. Create an array of products and filter products with

price > 100.



 **class Product { name; price; }**: lớp Product với 2 thuộc tính name (tên) và price (giá).

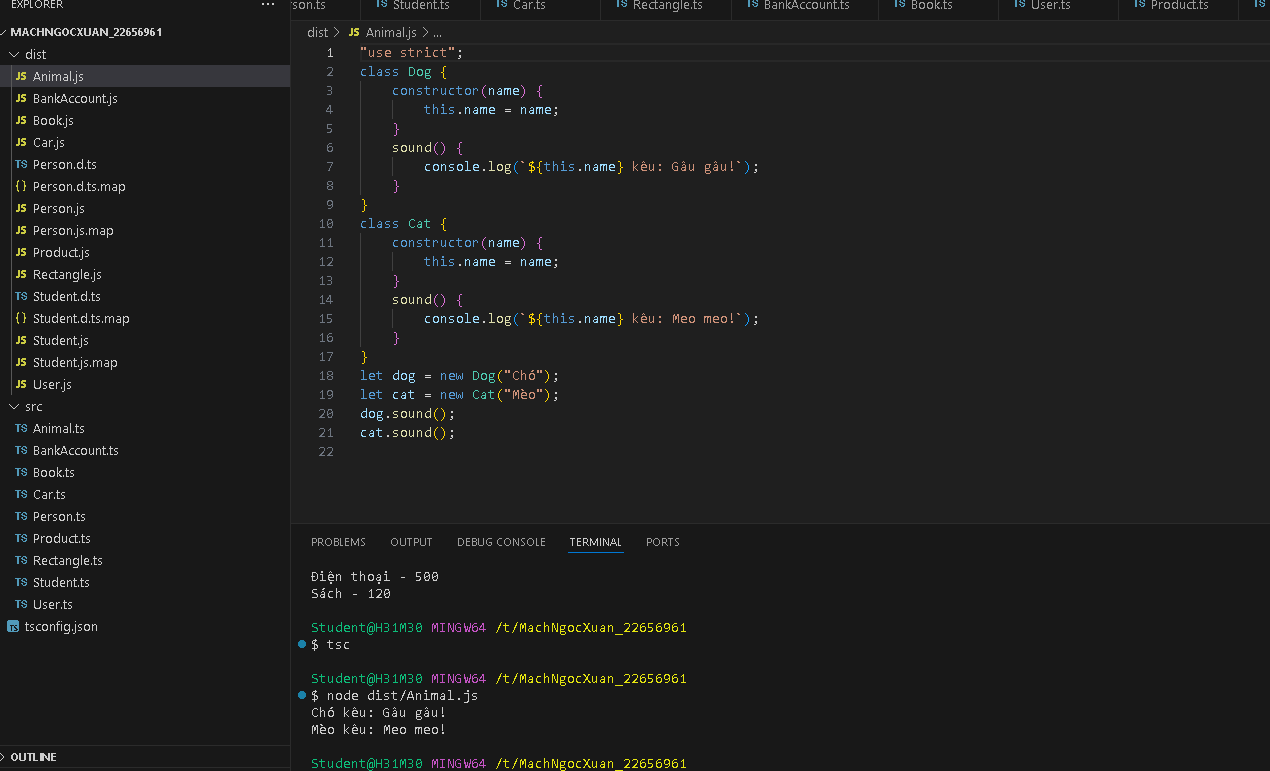
 **constructor(name, price)**: hàm khởi tạo để gán tên và giá khi tạo sản phẩm mới.

 **let products = [ ... ]**: tạo một mảng gồm nhiều sản phẩm.

 **filter(p => p.price > 100)**: lọc ra những sản phẩm có giá lớn hơn 100.

 **forEach(...)**: duyệt qua danh sách sản phẩm đắt và in ra tên + giá.

9. Define an interface Animal with name and method sound().



 **interface Animal**: định nghĩa một khuôn mẫu có name và phương thức sound().

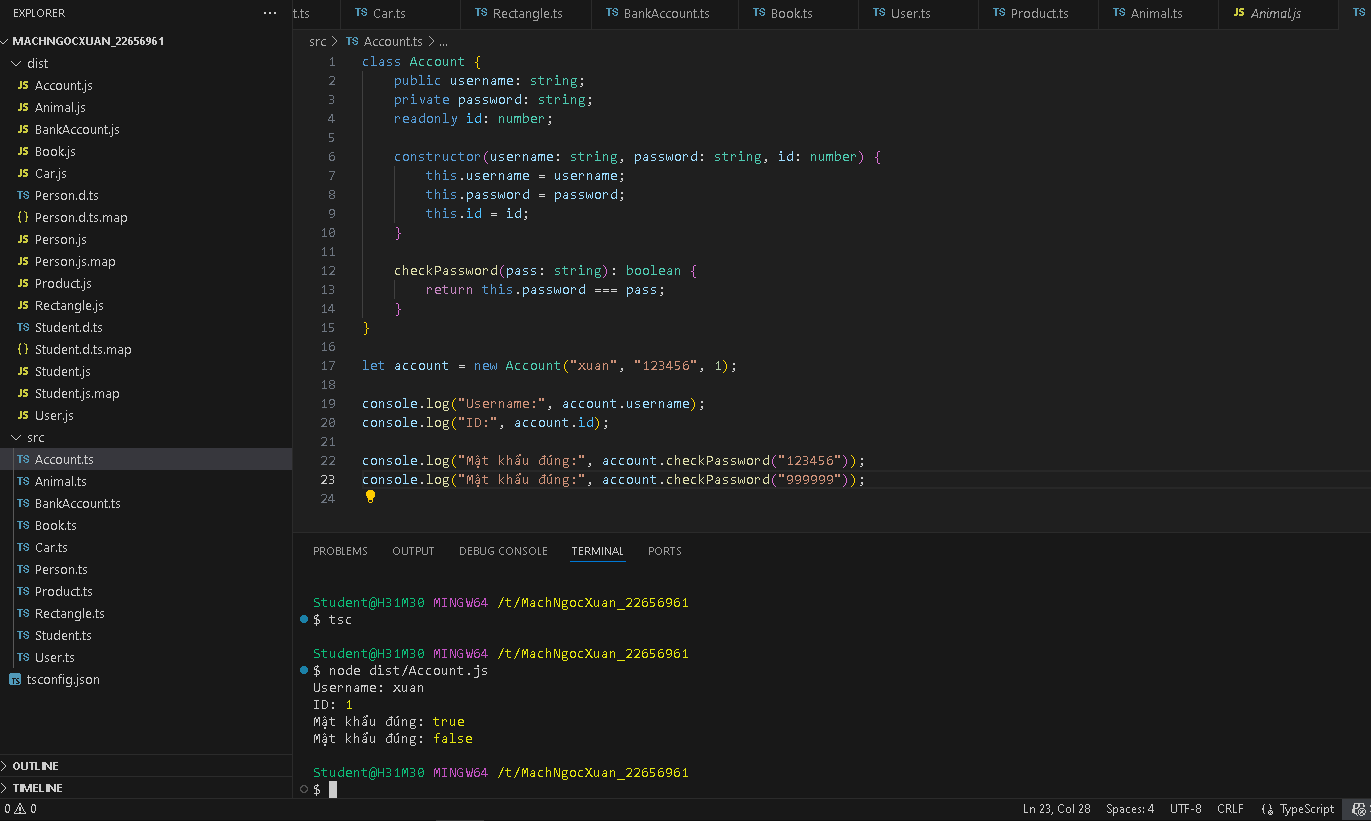
 **class Dog implements Animal**: lớp Dog bắt buộc phải có name và sound() vì nó triển khai từ Animal.

 **class Cat implements Animal**: tương tự với mèo.

 **dog.sound()**: in ra tiếng kêu của chó.

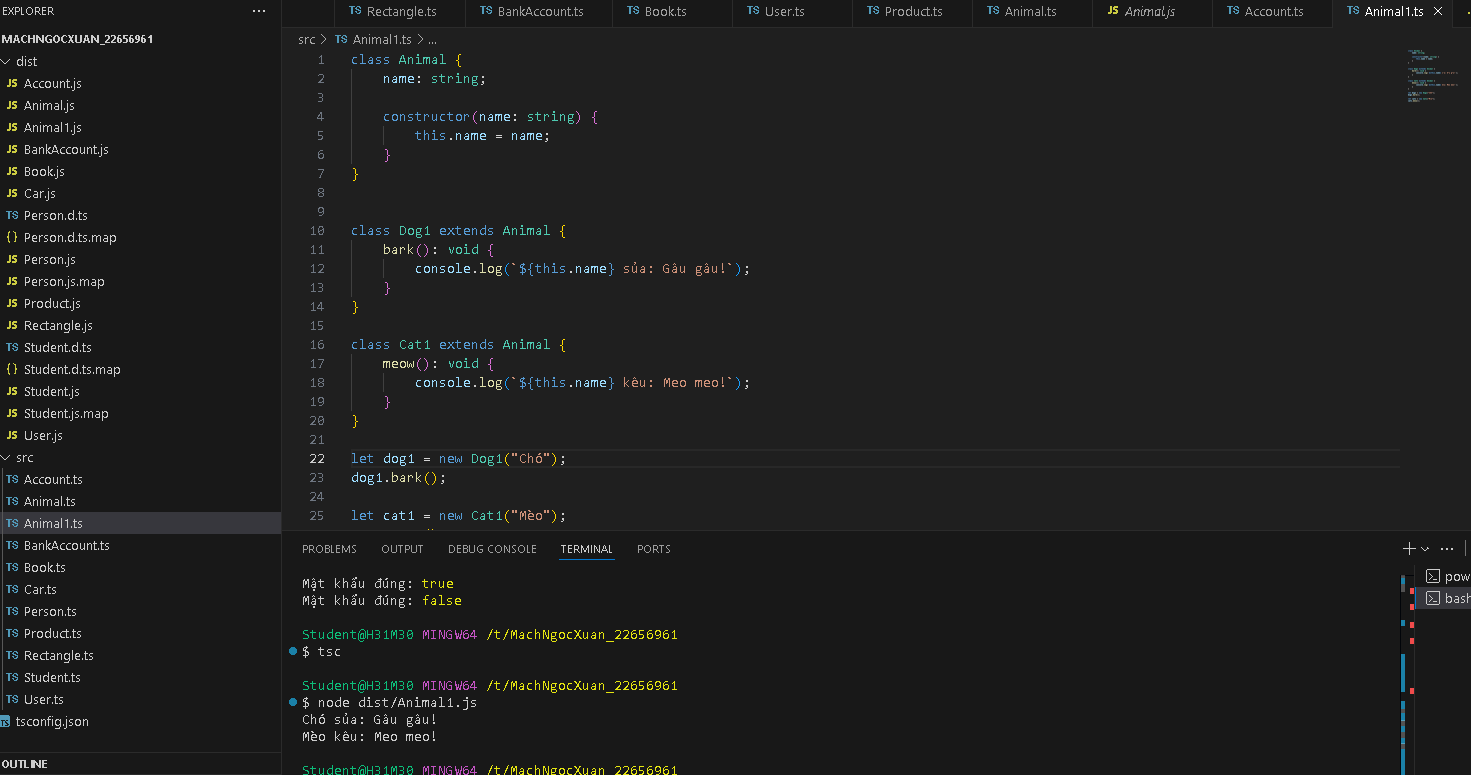
 **cat.sound()**: in ra tiếng kêu của mèo.

10. Create a class Account with public, private and readonly fields.



* **public username** → có thể truy cập từ ngoài class (acc.username).
* **private password** → không thể truy cập trực tiếp từ ngoài (acc.password ).
* **readonly id** → chỉ được gán 1 lần trong constructor, sau đó không thể thay đổi.
* **checkPassword()** → phương thức để kiểm tra mật khẩu (vì password private nên phải dùng hàm này).

11. Create a base class Animal. Extend Dog and Cat classes with methods bark() and meow().



 **class Animal1**: lớp cha, có thuộc tính name.

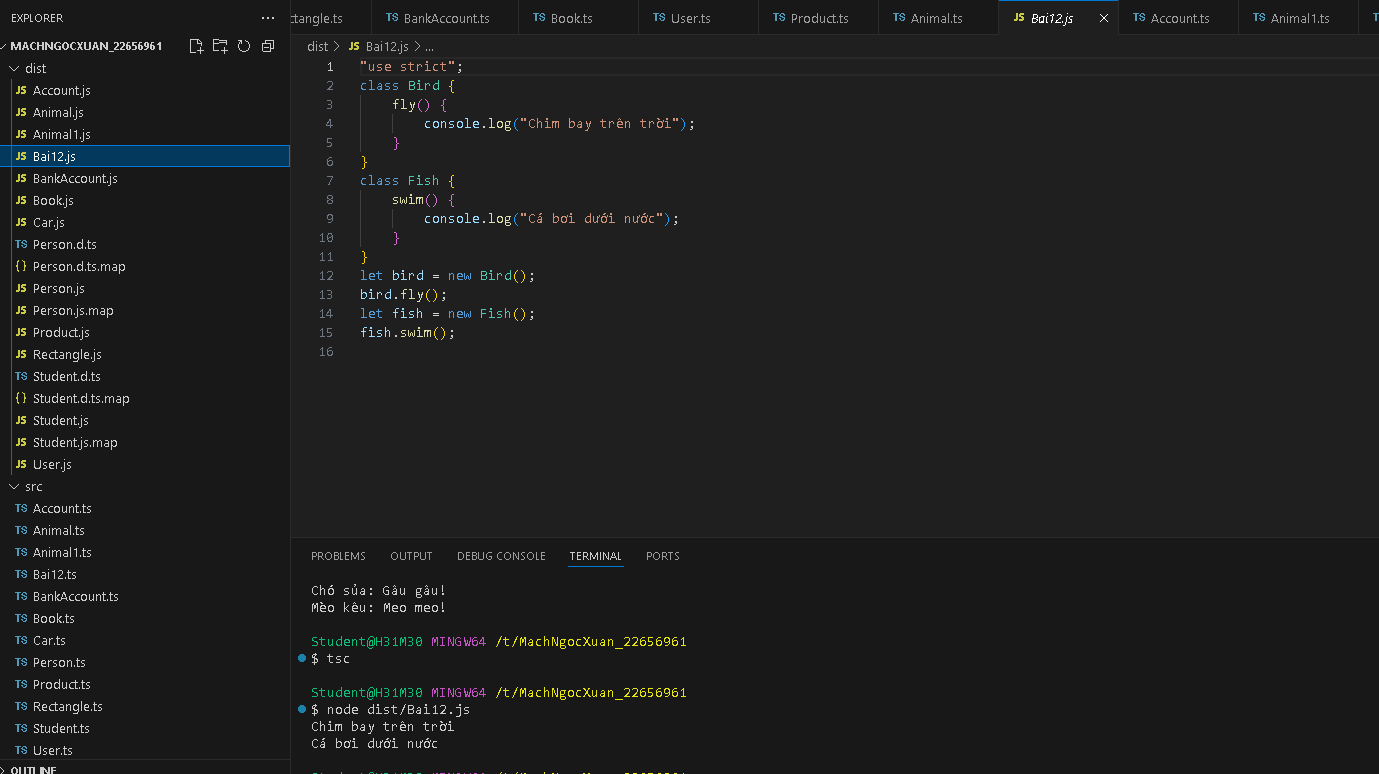
 **class Dog1 extends Animal1**: lớp con kế thừa Animal, thêm phương thức bark().

 **class Cat1 extends Animal1**: lớp con kế thừa Animal, thêm phương thức meow().

 **new Dog1("Chó")**: tạo đối tượng chó, gọi bark() in ra tiếng sủa.

 **new Cat1("Mèo")**: tạo đối tượng mèo, gọi meow() in ra tiếng kêu.

12. Define interfaces Flyable and Swimmable. Implement them in Bird and Fish classes.



 **interface Flyable**: định nghĩa hành vi có thể bay (fly()).

 **interface Swimmable**: định nghĩa hành vi có thể bơi (swim()).

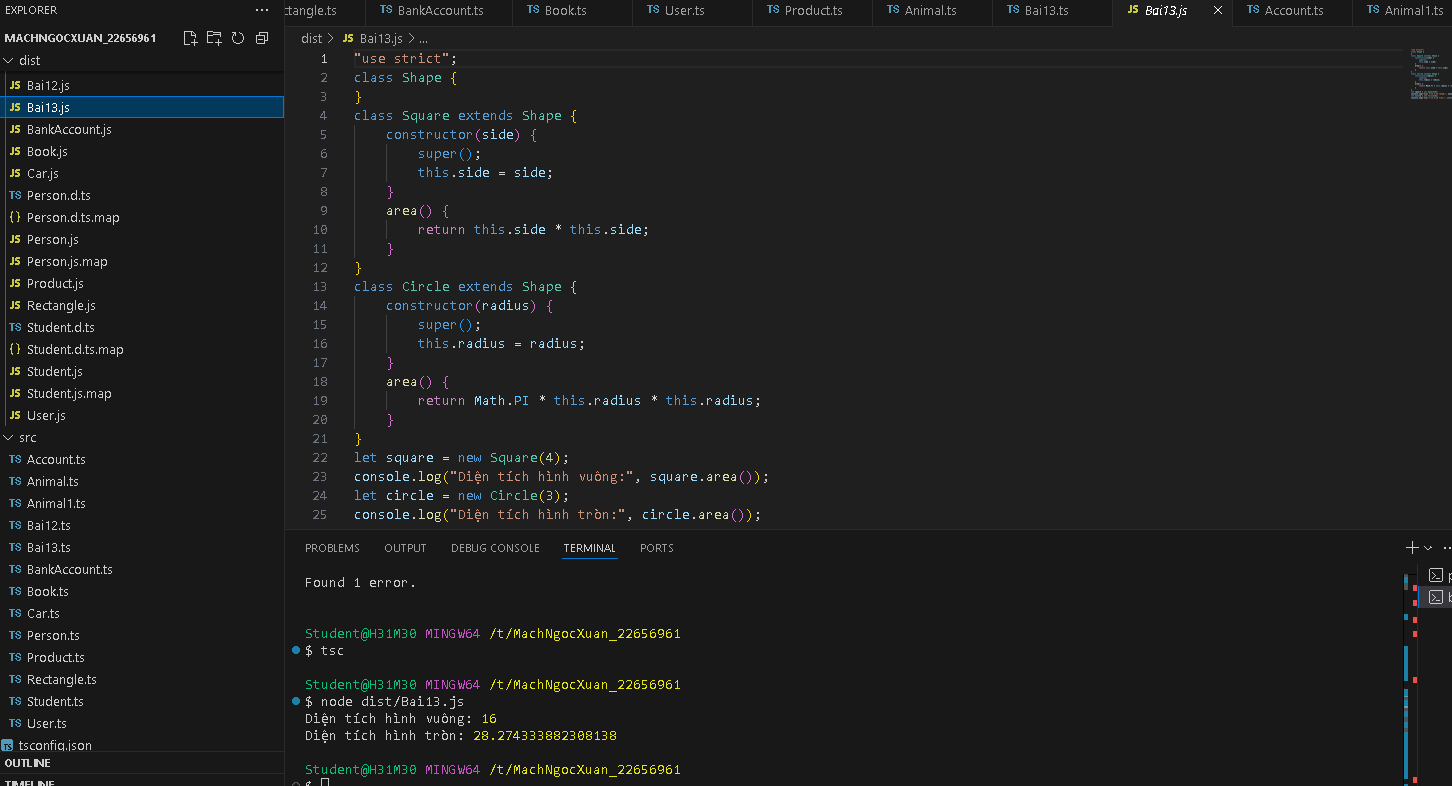
 **class Bird implements Flyable**: chim phải cài đặt phương thức fly().

 **class Fish implements Swimmable**: cá phải cài đặt phương thức swim().

 **bird.fly()**: in ra "Chim bay trên trời ".

 **fish.swim()**: in ra "Cá bơi dưới nước ".

13. Create an abstract class Shape with method area(). Implement Square and Circle.



 **abstract class Shape**: lớp trừu tượng, không thể tạo trực tiếp; có phương thức area() chưa được cài đặt.

 **class Square extends Shape**: hình vuông kế thừa Shape, cài đặt area() = cạnh × cạnh.

 **class Circle extends Shape**: hình tròn kế thừa Shape, cài đặt area() = π × r².

 **new Square(4)**: tạo hình vuông cạnh 4 → diện tích 16.

 **new Circle(3)**: tạo hình tròn bán kính 3 → diện tích ≈ 28.27.

14. Create a base class Employee. Extend Manager and Developer with specific methods.

15. Create a Library class that can store Book and User objects. Add method to add books.

16. Create a generic class Box that can store any type of value.

17. Write a singleton Logger class that logs messages to console.

18. Create a static class MathUtil with methods add(), subtract(), multiply(), divide().

19. Demonstrate method overriding using polymorphism with Animal and subclasses.

20. Write a Vehicle interface and implement it in Car and Bike classes.

21. Create a generic Repository class with methods add(), getAll().

22. Create a class Stack with push, pop, peek, isEmpty methods.

23. Create an interface Payment with method pay(amount). Implement CashPayment and

CardPayment.

24. Create an abstract class Appliance with method turnOn(). Implement Fan and AirConditioner.

25. Create a class Shape with a static method describe().

26. Create a class Order with list of products. Add method to calculate total price.

27. Create a class Teacher that extends Person. Add subject attribute and introduce method.

28. Create a class Animal with protected method makeSound(). Extend Dog and Cat to override it.

29. Create an interface Movable with method move(). Implement it in Car and Robot.

30. Create a class School with list of Students and Teachers. Add method to display info.