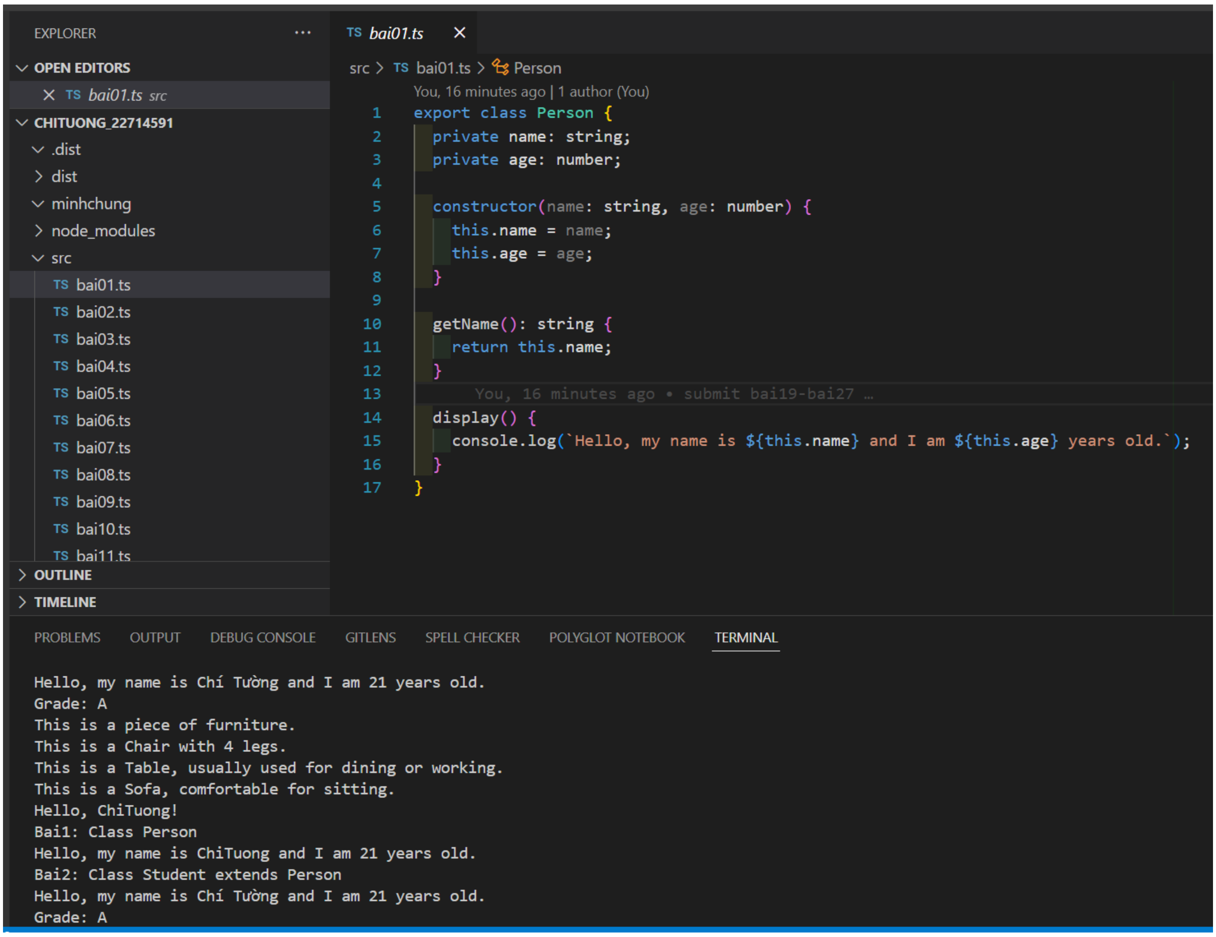
Minh Chứng BT React Native Lab01

Nguyễn Lâm Chí Tường

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

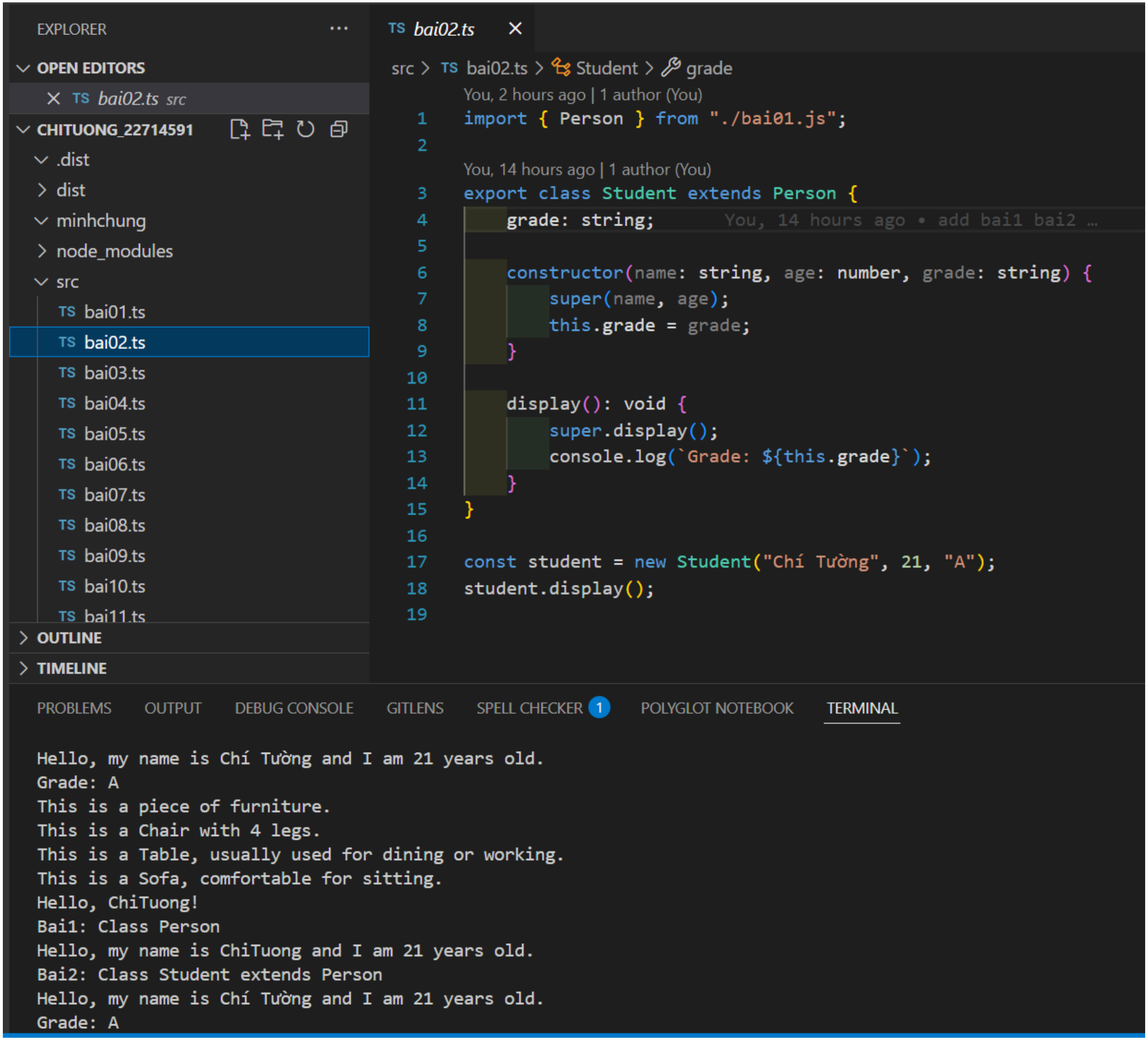


```typescript
export class Person {
  private name: string;
  private age: number;

  constructor(name: string, age: number) {
    this.name = name;
    this.age = age;
  }

  getName(): string {
    return this.name;
  }

  display() {
    console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
  }
}
```

```
Hello, my name is Chí Tường and I am 21 years old.
Grade: A
This is a piece of furniture.
This is a Chair with 4 legs.
This is a Table, usually used for dining or working.
This is a Sofa, comfortable for sitting.
Hello, ChiTuong!
Bai1: Class Person
Hello, my name is ChiTuong and I am 21 years old.
Bai2: Class Student extends Person
Hello, my name is Chí Tường and I am 21 years old.
Grade: A
```

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

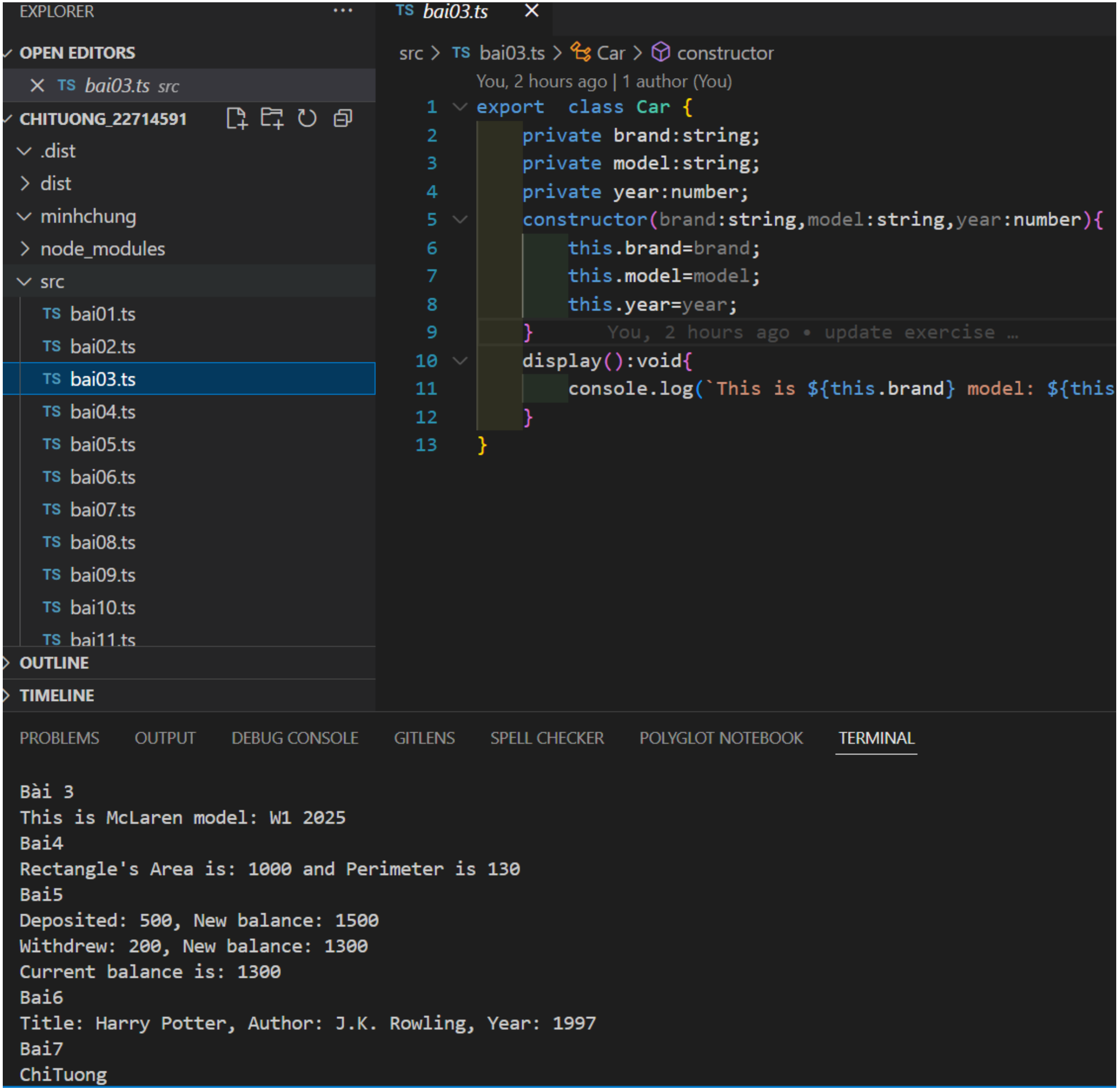
display all info.



```
You, 2 hours ago | 1 author (You)
import { Person } from "./bai01.js";

You, 14 hours ago | 1 author (You)
export class Student extends Person {
    grade: string;        You, 14 hours ago • add bai1 bai2 …

    constructor(name: string, age: number, grade: string) {
        super(name, age);
        this.grade = grade;
    }

    display(): void {
        super.display();
        console.log(`Grade: ${this.grade}`);
    }
}

const student = new Student("Chí Tường", 21, "A");
student.display();
```

```
Hello, my name is Chí Tường and I am 21 years old.
Grade: A
This is a piece of furniture.
This is a Chair with 4 legs.
This is a Table, usually used for dining or working.
This is a Sofa, comfortable for sitting.
Hello, ChiTuong!
Bai1: Class Person
Hello, my name is ChiTuong and I am 21 years old.
Bai2: Class Student extends Person
Hello, my name is Chí Tường and I am 21 years old.
Grade: A
```

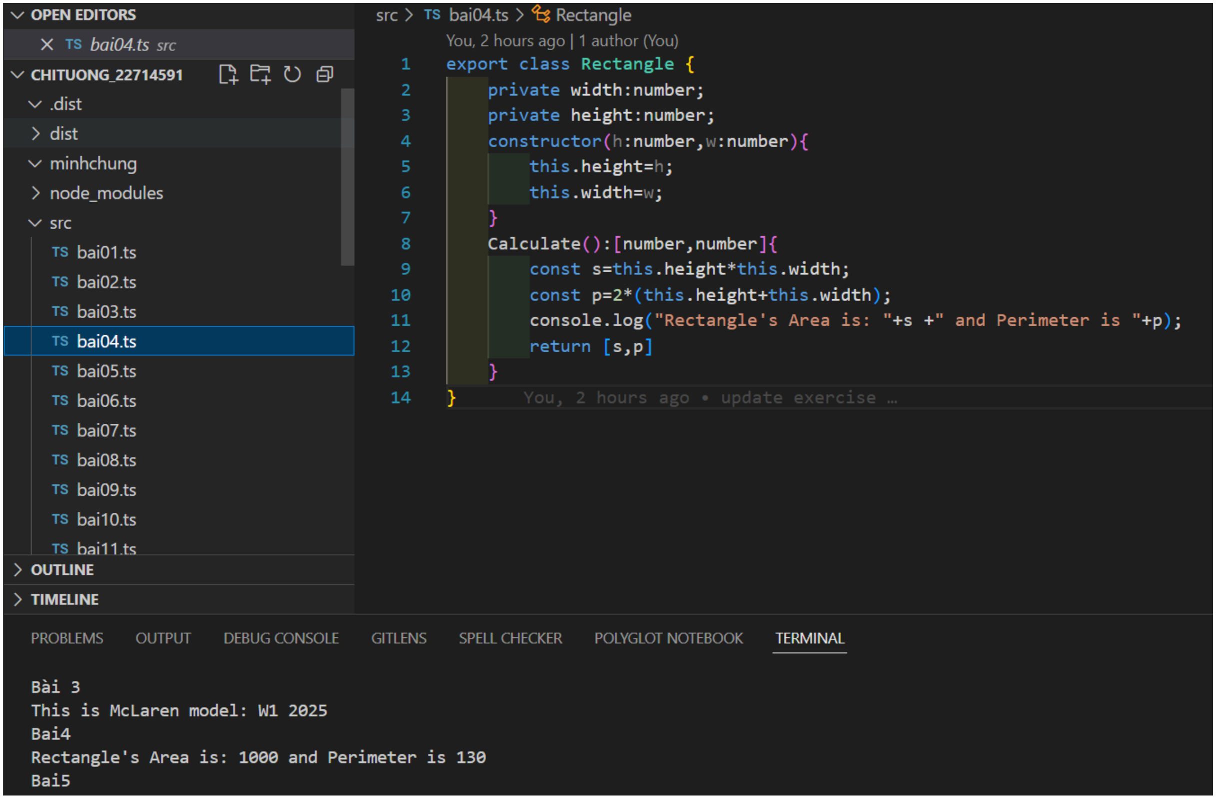
3. Create a class Car with properties brand, model, year. Write a method to show car info.



```
export  class Car {
    private brand:string;
    private model:string;
    private year:number;
    constructor(brand:string,model:string,year:number){
        this.brand=brand;
        this.model=model;
        this.year=year;
    }       You, 2 hours ago • update exercise …
    display():void{
        console.log(`This is ${this.brand} model: ${this
    }
}
```

```
Bài 3
This is McLaren model: W1 2025
Bai4
Rectangle's Area is: 1000 and Perimeter is 130
Bai5
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
Current balance is: 1300
Bai6
Title: Harry Potter, Author: J.K. Rowling, Year: 1997
Bai7
ChiTuong
```

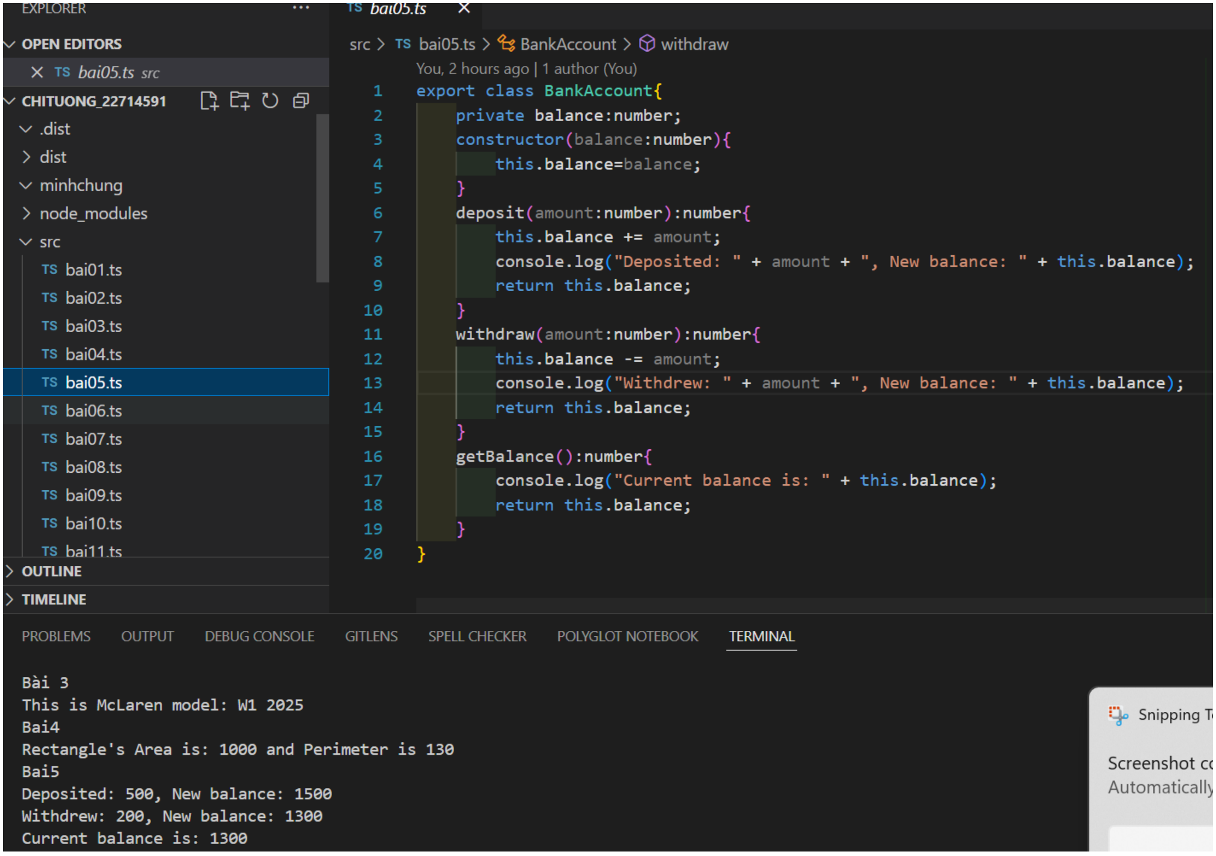
4. Create a class Rectangle with width and height. Write a method to calculate area and perimeter.



```typescript
export class Rectangle {
    private width:number;
    private height:number;
    constructor(h:number,w:number){
        this.height=h;
        this.width=w;
    }
    Calculate():[number,number]{
        const s=this.height*this.width;
        const p=2*(this.height+this.width);
        console.log("Rectangle's Area is: "+s +" and Perimeter is "+p);
        return [s,p]
    }
}
```

```
Bài 3
This is McLaren model: W1 2025
Bai4
Rectangle's Area is: 1000 and Perimeter is 130
Bai5
```

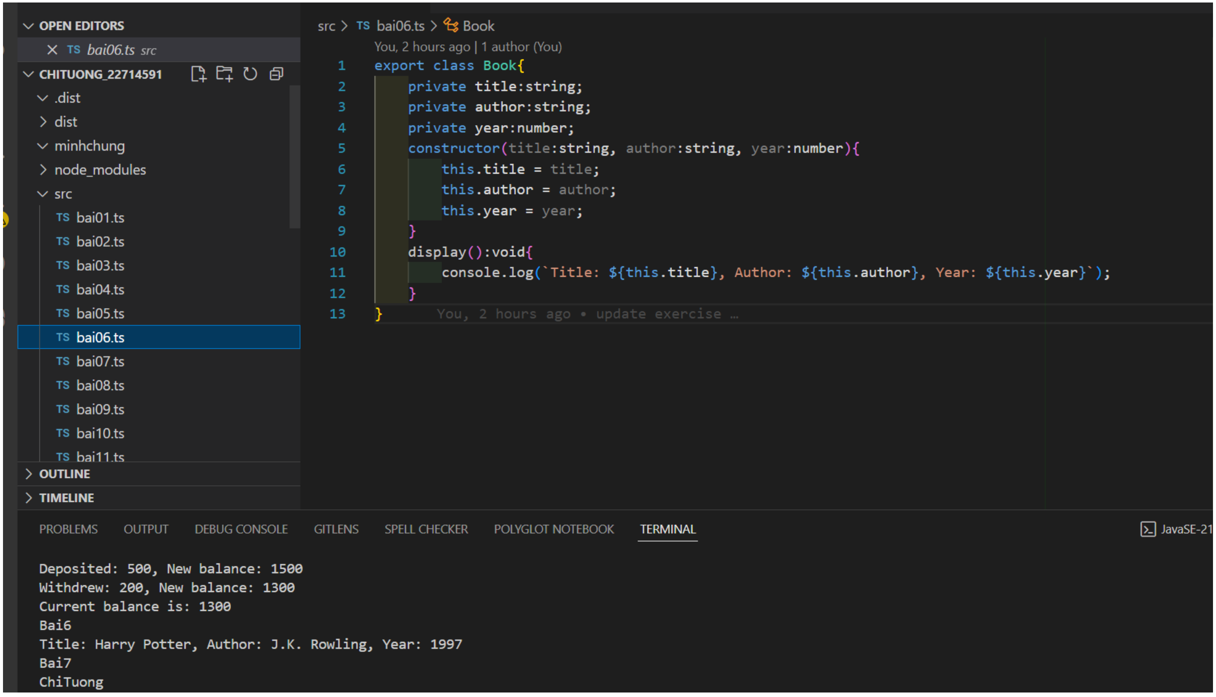
5. Create a class BankAccount with balance. Add methods deposit() and withdraw().



```typescript
export class BankAccount{
    private balance:number;
    constructor(balance:number){
        this.balance=balance;
    }
    deposit(amount:number):number{
        this.balance += amount;
        console.log("Deposited: " + amount + ", New balance: " + this.balance);
        return this.balance;
    }
    withdraw(amount:number):number{
        this.balance -= amount;
        console.log("Withdrew: " + amount + ", New balance: " + this.balance);
        return this.balance;
    }
    getBalance():number{
        console.log("Current balance is: " + this.balance);
        return this.balance;
    }
}
```

```
Bài 3
This is McLaren model: W1 2025
Bai4
Rectangle's Area is: 1000 and Perimeter is 130
Bai5
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
Current balance is: 1300
```

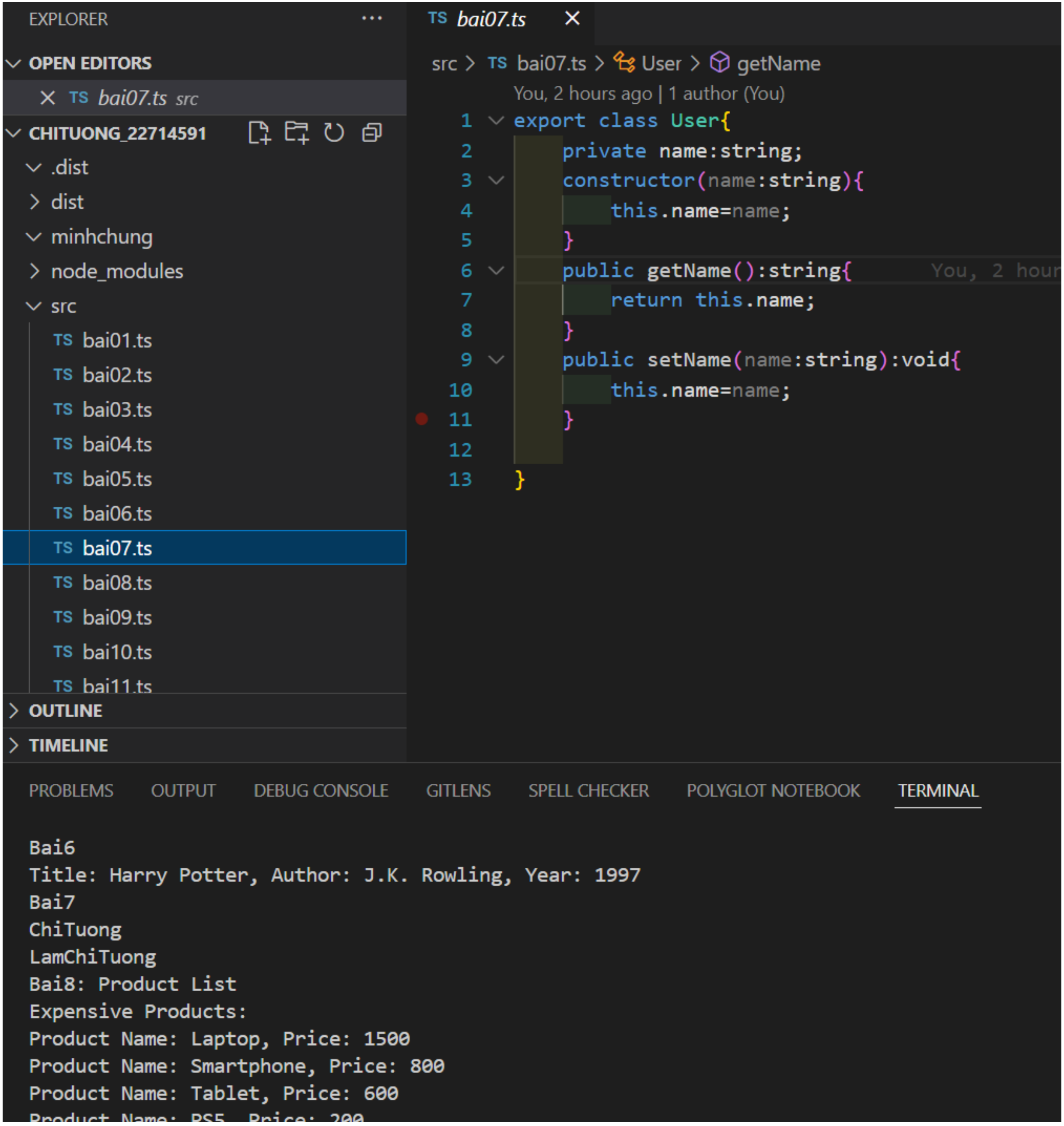
6. Create a class Book with attributes title, author, year.



```typescript
export class Book{
    private title:string;
    private author:string;
    private year:number;
    constructor(title:string, author:string, year:number){
        this.title = title;
        this.author = author;
        this.year = year;
    }
    display():void{
        console.log(`Title: ${this.title}, Author: ${this.author}, Year: ${this.year}`);
    }
}       You, 2 hours ago • update exercise …
```

```
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
Current balance is: 1300
Bai6
Title: Harry Potter, Author: J.K. Rowling, Year: 1997
Bai7
ChiTuong
```

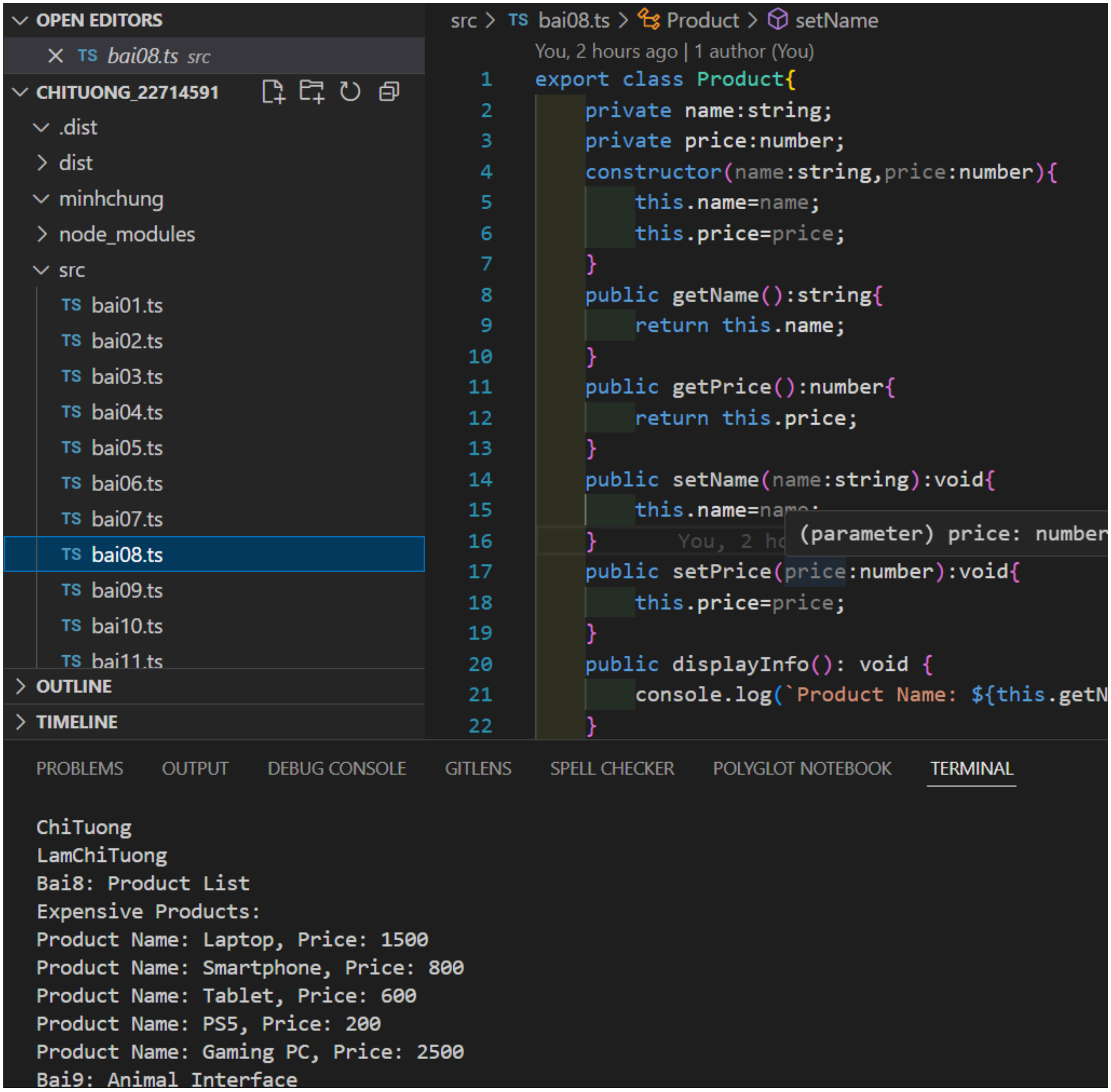
7. Write a class User with private property name and getter/setter.



```typescript
export class User{
    private name:string;
    constructor(name:string){
        this.name=name;
    }
    public getName():string{        You, 2 hour
        return this.name;
    }
    public setName(name:string):void{
        this.name=name;
    }

}
```

```
Bai6
Title: Harry Potter, Author: J.K. Rowling, Year: 1997
Bai7
ChiTuong
LamChiTuong
Bai8: Product List
Expensive Products:
Product Name: Laptop, Price: 1500
Product Name: Smartphone, Price: 800
Product Name: Tablet, Price: 600
Product Name: PS5, Price: 200
```

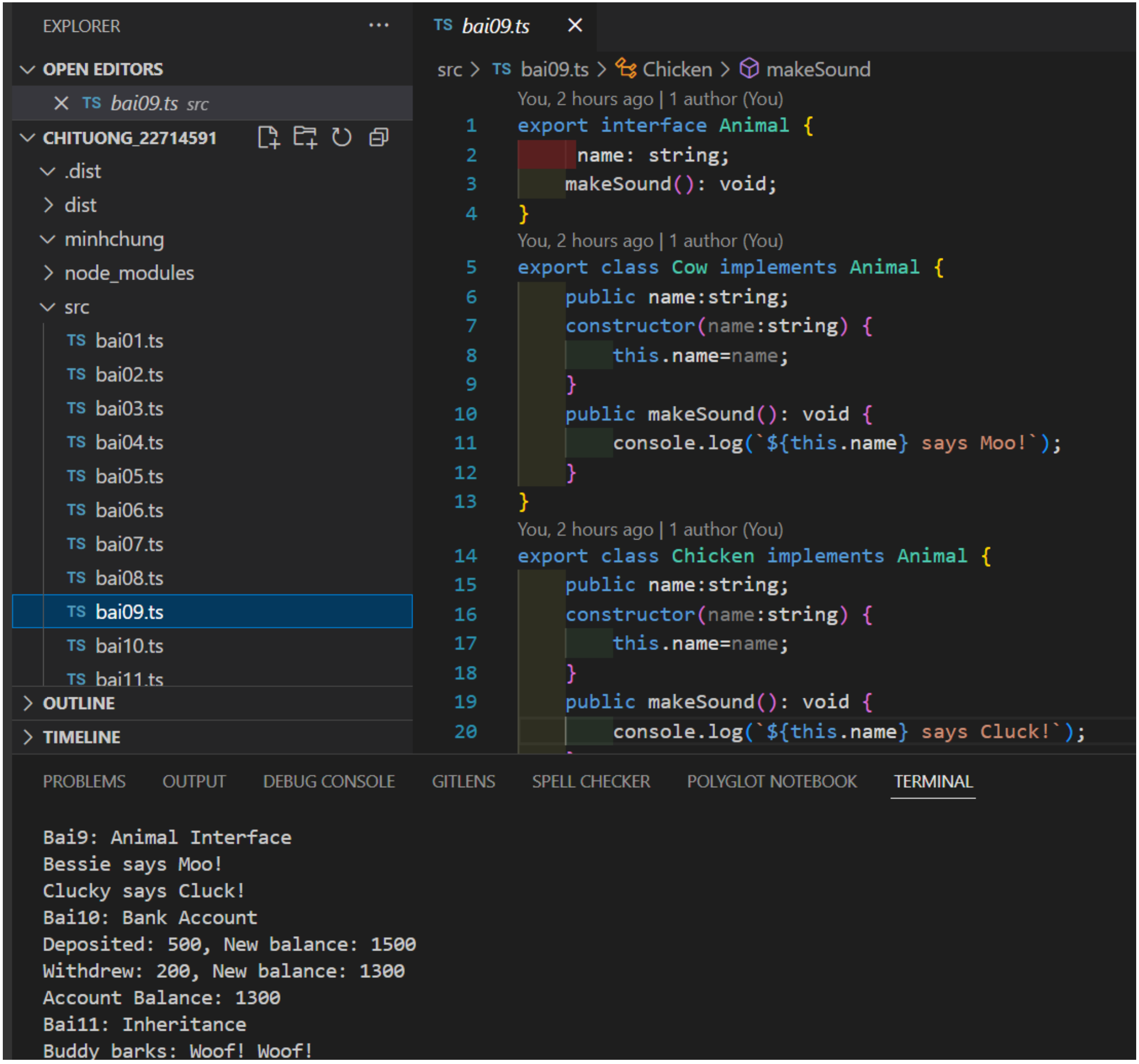
8. Create a Product class with name, price. Create an array of products and filter products with price > 100.



```
export class Product{
    private name:string;
    private price:number;
    constructor(name:string,price:number){
        this.name=name;
        this.price=price;
    }
    public getName():string{
        return this.name;
    }
    public getPrice():number{
        return this.price;
    }
    public setName(name:string):void{
        this.name=name;
    }
    public setPrice(price:number):void{
        this.price=price;
    }
    public displayInfo(): void {
        console.log(`Product Name: ${this.getN
    }
```

```
ChiTuong
LamChiTuong
Bai8: Product List
Expensive Products:
Product Name: Laptop, Price: 1500
Product Name: Smartphone, Price: 800
Product Name: Tablet, Price: 600
Product Name: PS5, Price: 200
Product Name: Gaming PC, Price: 2500
Bai9: Animal Interface
```

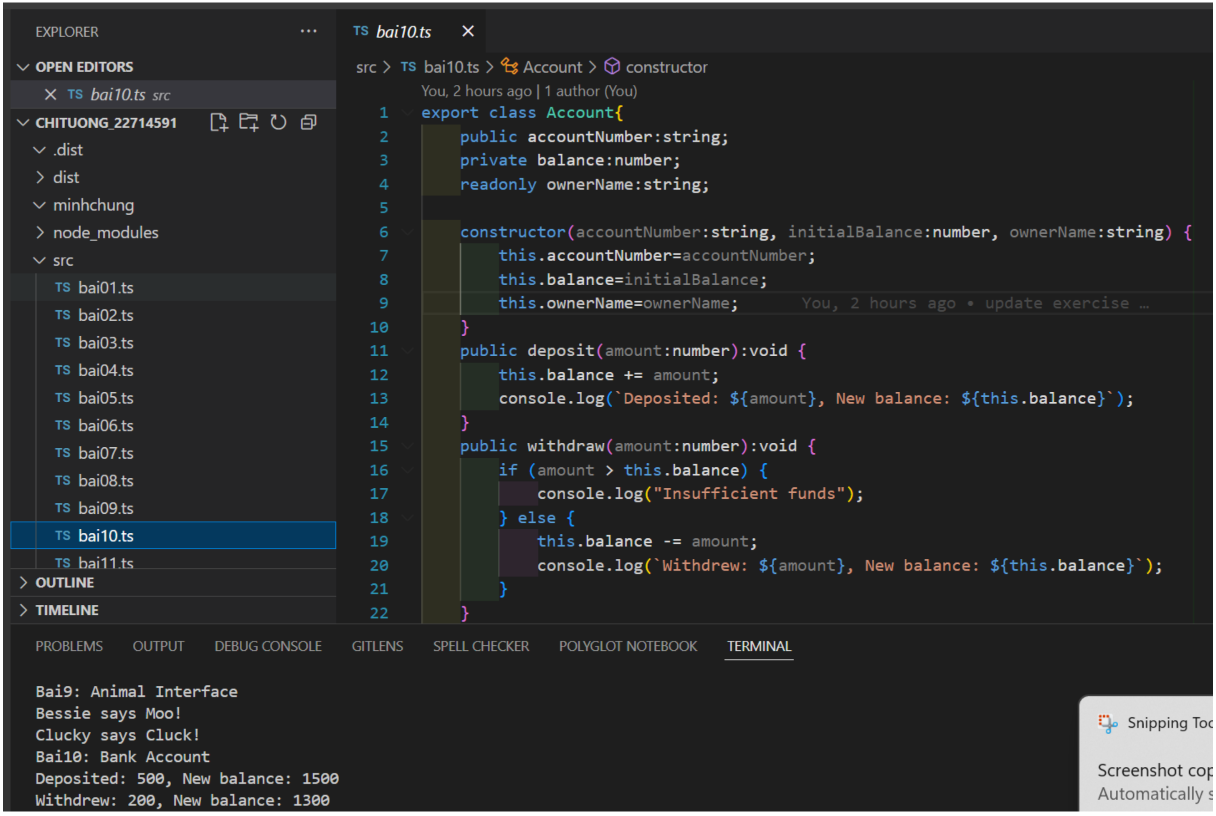
9. Define an interface Animal with name and method sound().



```
export interface Animal {
    name: string;
    makeSound(): void;
}
export class Cow implements Animal {
    public name:string;
    constructor(name:string) {
        this.name=name;
    }
    public makeSound(): void {
        console.log(`${this.name} says Moo!`);
    }
}
export class Chicken implements Animal {
    public name:string;
    constructor(name:string) {
        this.name=name;
    }
    public makeSound(): void {
        console.log(`${this.name} says Cluck!`);
```

```
Bai9: Animal Interface
Bessie says Moo!
Clucky says Cluck!
Bai10: Bank Account
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
Account Balance: 1300
Bai11: Inheritance
Buddy barks: Woof! Woof!
```

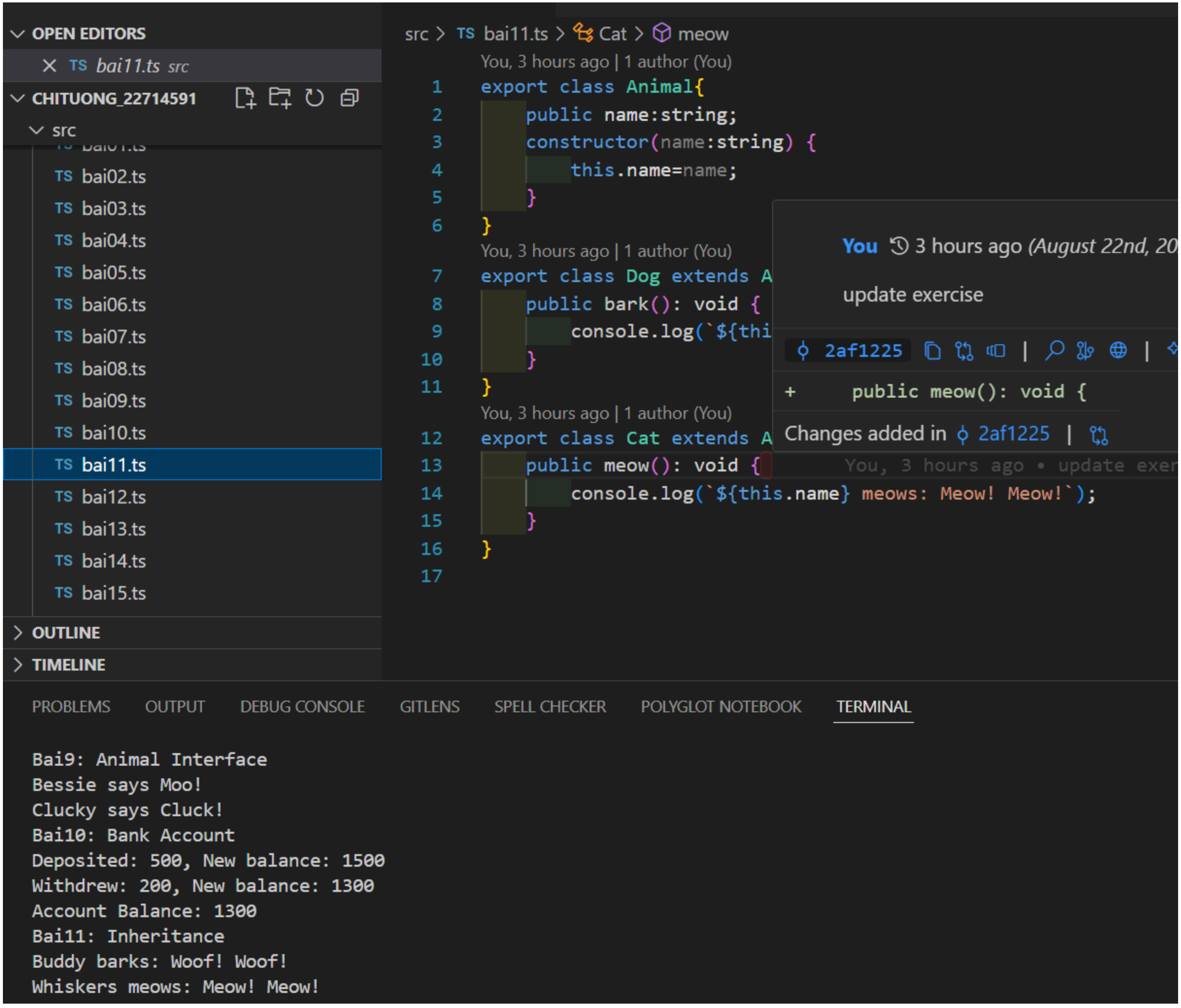
10. Create a class Account with public, private and readonly fields.



```typescript
export class Account{
    public accountNumber:string;
    private balance:number;
    readonly ownerName:string;

    constructor(accountNumber:string, initialBalance:number, ownerName:string) {
        this.accountNumber=accountNumber;
        this.balance=initialBalance;
        this.ownerName=ownerName;
    }
    public deposit(amount:number):void {
        this.balance += amount;
        console.log(`Deposited: ${amount}, New balance: ${this.balance}`);
    }
    public withdraw(amount:number):void {
        if (amount > this.balance) {
            console.log("Insufficient funds");
        } else {
            this.balance -= amount;
            console.log(`Withdrew: ${amount}, New balance: ${this.balance}`);
        }
    }
```

```
Bai9: Animal Interface
Bessie says Moo!
Clucky says Cluck!
Bai10: Bank Account
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
```

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

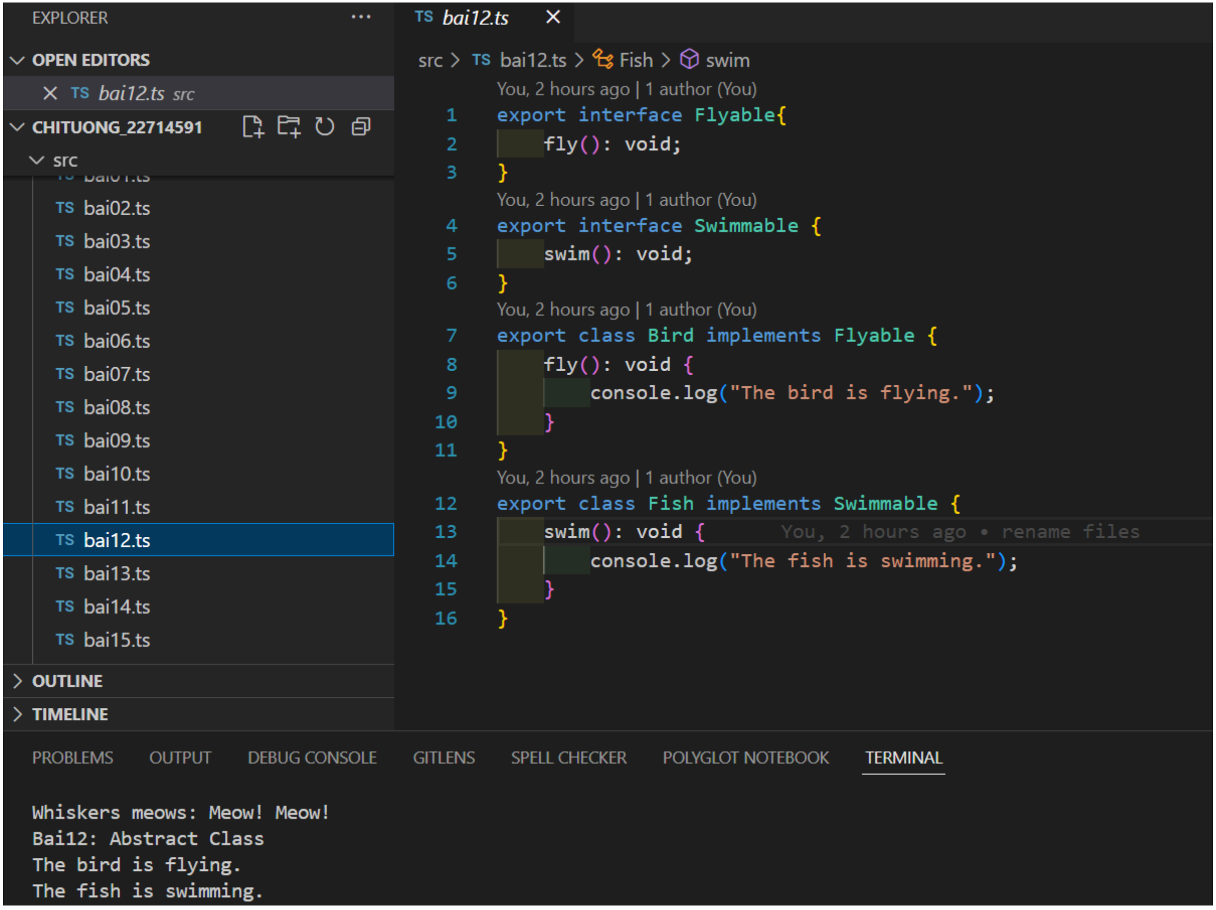


```
export class Animal{
    public name:string;
    constructor(name:string) {
        this.name=name;
    }
}
export class Dog extends A
    public bark(): void {
        console.log(`${thi
    }
}
export class Cat extends A
    public meow(): void {
        console.log(`${this.name} meows: Meow! Meow!`);
    }
}
```

```
+       public meow(): void {
```

```
Bai9: Animal Interface
Bessie says Moo!
Clucky says Cluck!
Bai10: Bank Account
Deposited: 500, New balance: 1500
Withdrew: 200, New balance: 1300
Account Balance: 1300
Bai11: Inheritance
Buddy barks: Woof! Woof!
Whiskers meows: Meow! Meow!
```

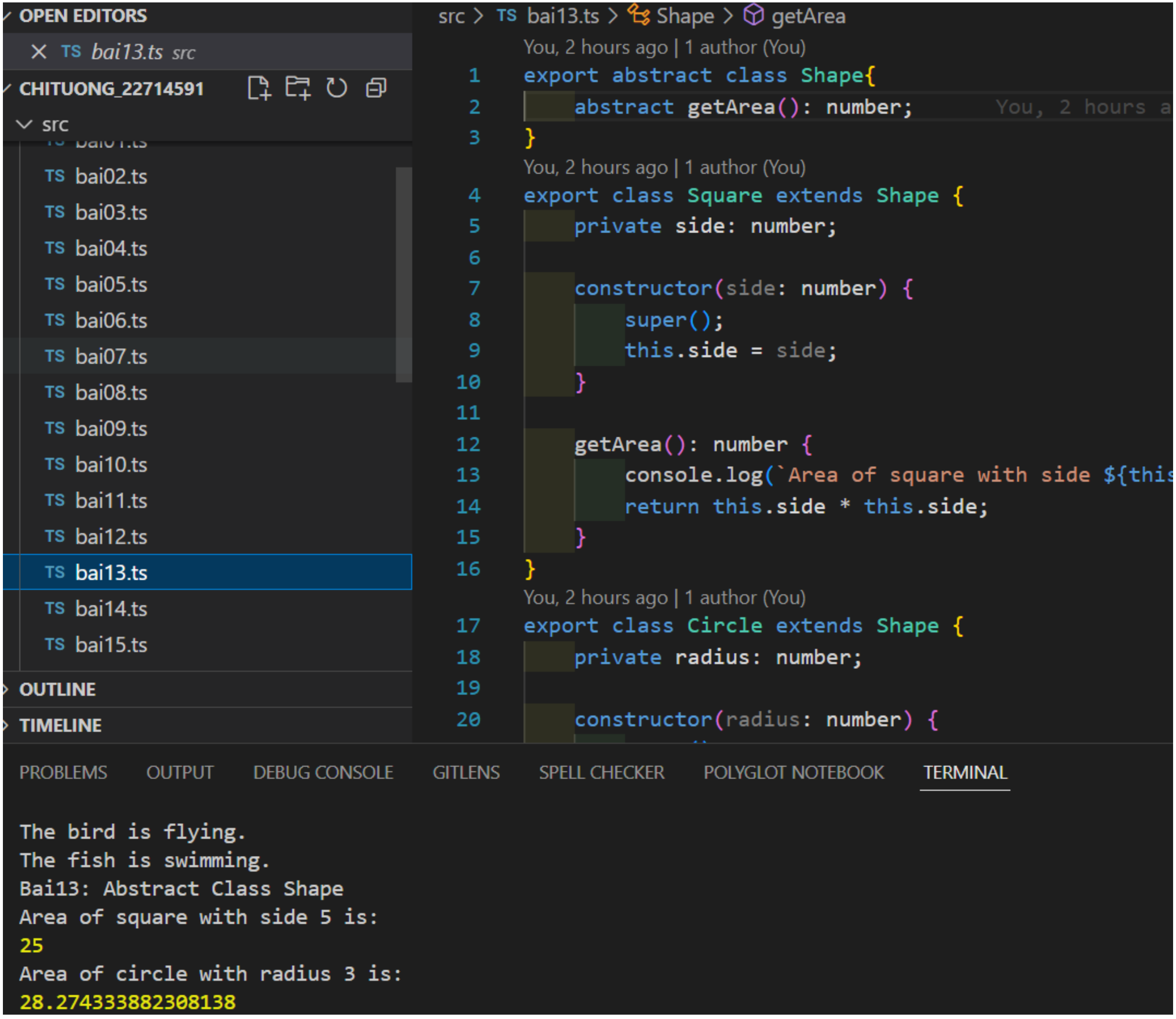
12. Define interfaces Flyable and Swimmable. Implement them in Bird and Fish classes.



```typescript
You, 2 hours ago | 1 author (You)
export interface Flyable{
    fly(): void;
}
You, 2 hours ago | 1 author (You)
export interface Swimmable {
    swim(): void;
}
You, 2 hours ago | 1 author (You)
export class Bird implements Flyable {
    fly(): void {
        console.log("The bird is flying.");
    }
}
You, 2 hours ago | 1 author (You)
export class Fish implements Swimmable {
    swim(): void {        You, 2 hours ago • rename files
        console.log("The fish is swimming.");
    }
}
```

```
Whiskers meows: Meow! Meow!
Bai12: Abstract Class
The bird is flying.
The fish is swimming.
```

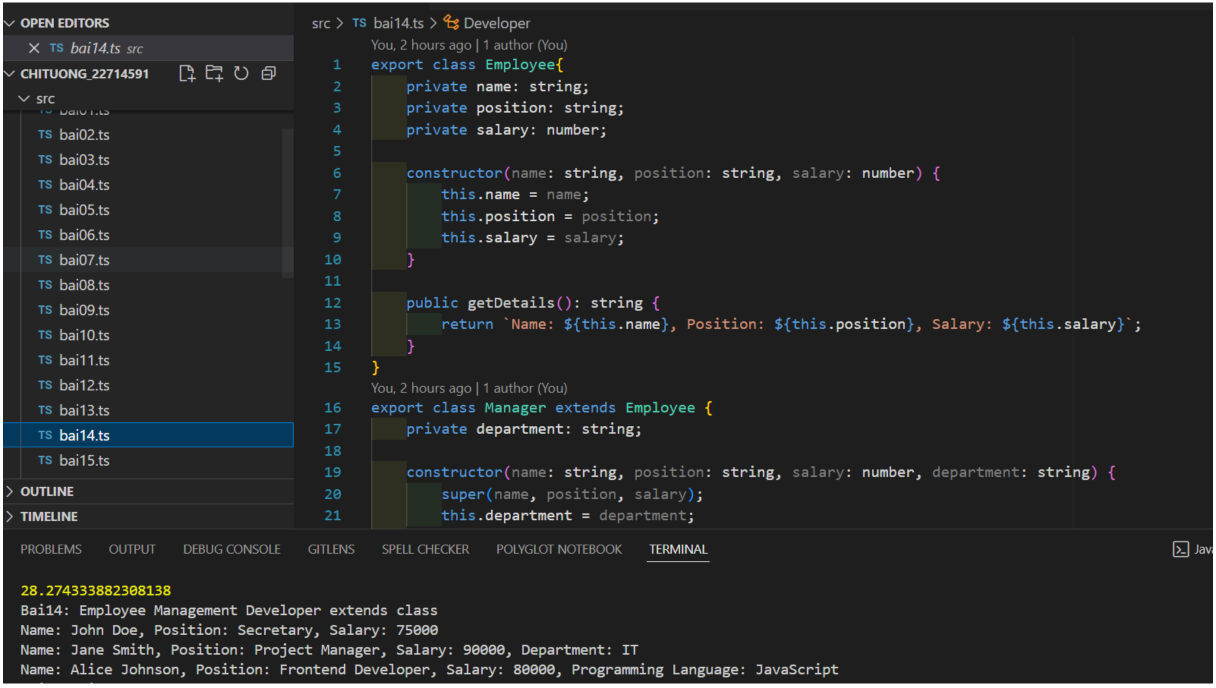
13. Create an abstract class Shape with method area(). Implement Square and Circle.



```typescript
export abstract class Shape{
    abstract getArea(): number;
}
export class Square extends Shape {
    private side: number;

    constructor(side: number) {
        super();
        this.side = side;
    }

    getArea(): number {
        console.log(`Area of square with side ${this
        return this.side * this.side;
    }
}
export class Circle extends Shape {
    private radius: number;

    constructor(radius: number) {
```

```
The bird is flying.
The fish is swimming.
Bai13: Abstract Class Shape
Area of square with side 5 is:
25
Area of circle with radius 3 is:
28.274333882308138
```

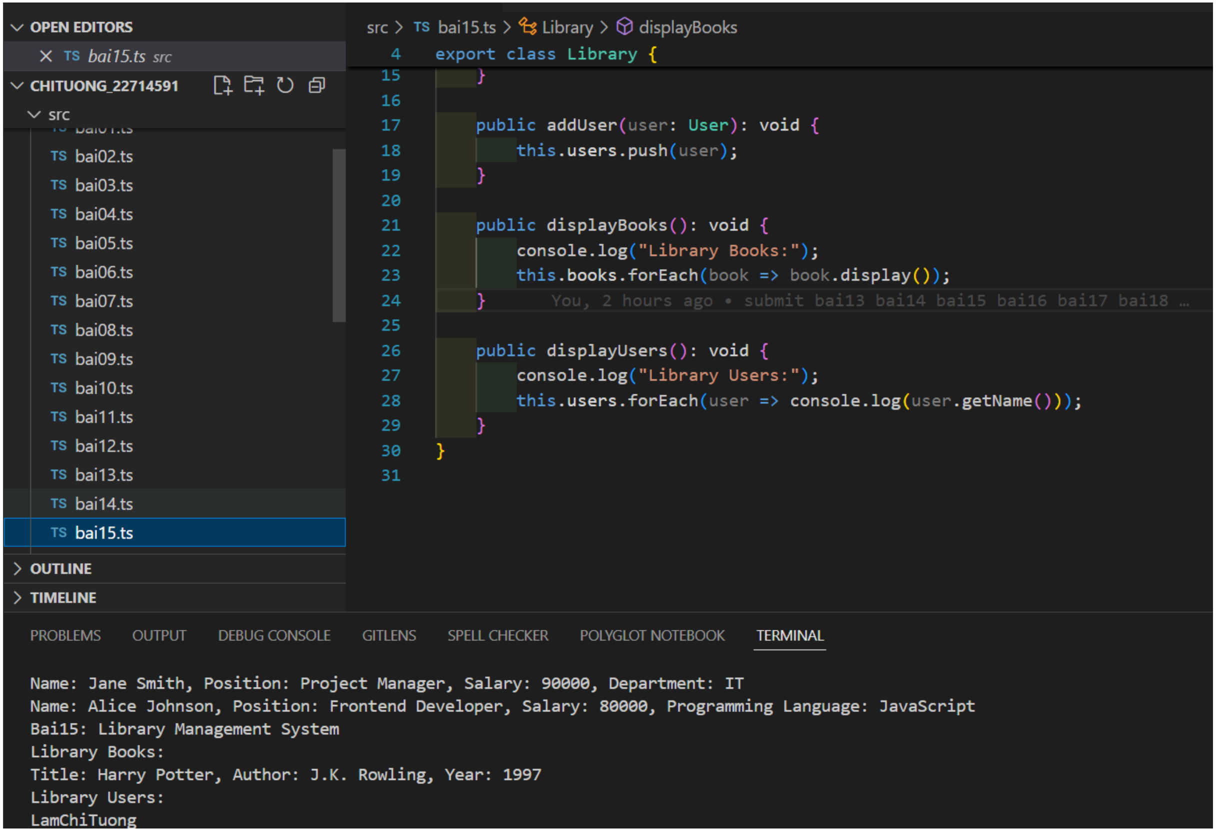
14. Create a base class Employee. Extend Manager and Developer with specific methods.



```typescript
export class Employee{
    private name: string;
    private position: string;
    private salary: number;

    constructor(name: string, position: string, salary: number) {
        this.name = name;
        this.position = position;
        this.salary = salary;
    }

    public getDetails(): string {
        return `Name: ${this.name}, Position: ${this.position}, Salary: ${this.salary}`;
    }
}
export class Manager extends Employee {
    private department: string;

    constructor(name: string, position: string, salary: number, department: string) {
        super(name, position, salary);
        this.department = department;
```

```
28.274333882308138
Bai14: Employee Management Developer extends class
Name: John Doe, Position: Secretary, Salary: 75000
Name: Jane Smith, Position: Project Manager, Salary: 90000, Department: IT
Name: Alice Johnson, Position: Frontend Developer, Salary: 80000, Programming Language: JavaScript
```

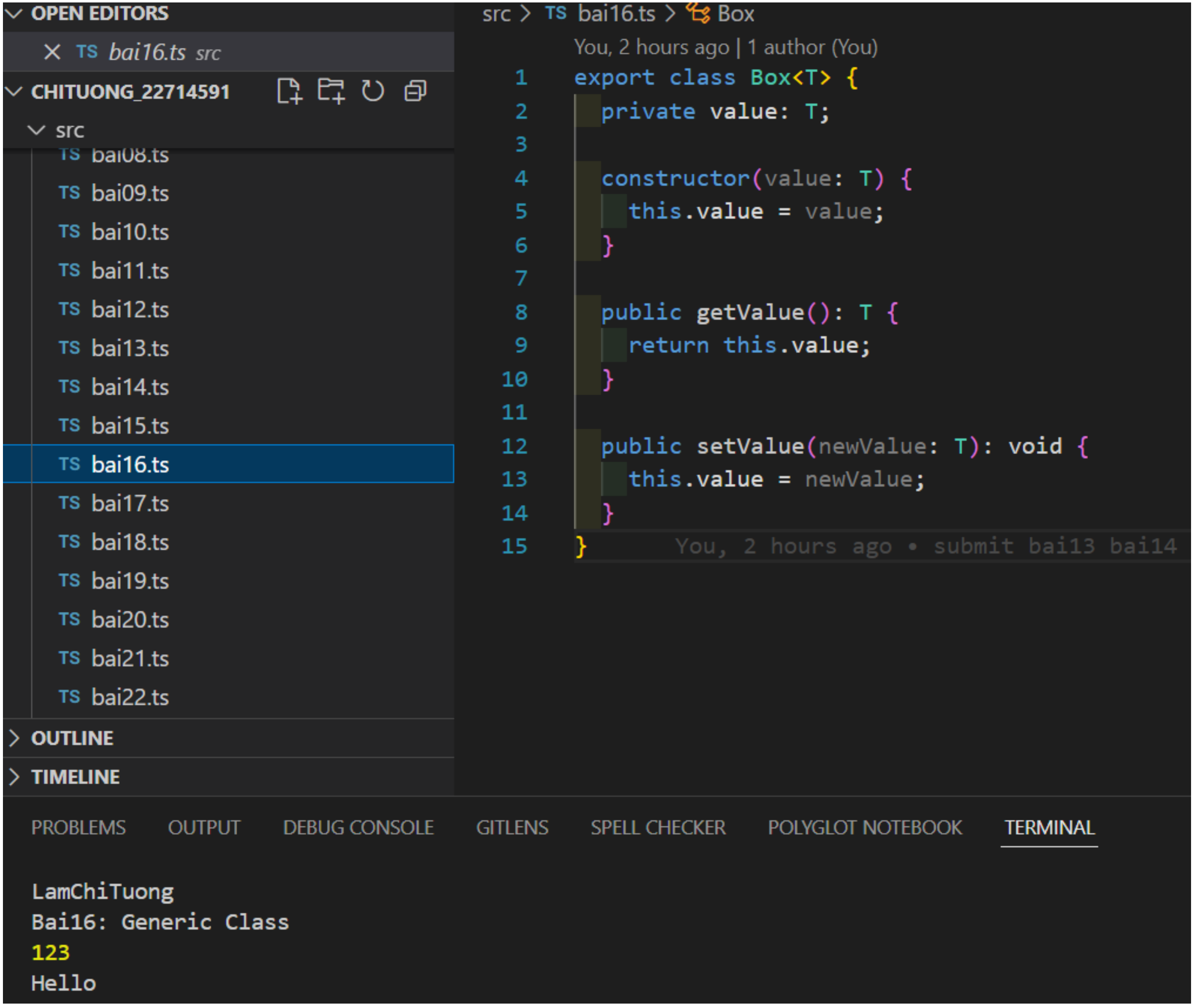
15. Create a Library class that can store Book and User objects. Add method to add books.



```
export class Library {
    }

    public addUser(user: User): void {
        this.users.push(user);
    }

    public displayBooks(): void {
        console.log("Library Books:");
        this.books.forEach(book => book.display());
    }

    public displayUsers(): void {
        console.log("Library Users:");
        this.users.forEach(user => console.log(user.getName()));
    }
}
```

```
Name: Jane Smith, Position: Project Manager, Salary: 90000, Department: IT
Name: Alice Johnson, Position: Frontend Developer, Salary: 80000, Programming Language: JavaScript
Bai15: Library Management System
Library Books:
Title: Harry Potter, Author: J.K. Rowling, Year: 1997
Library Users:
LamChiTuong
```

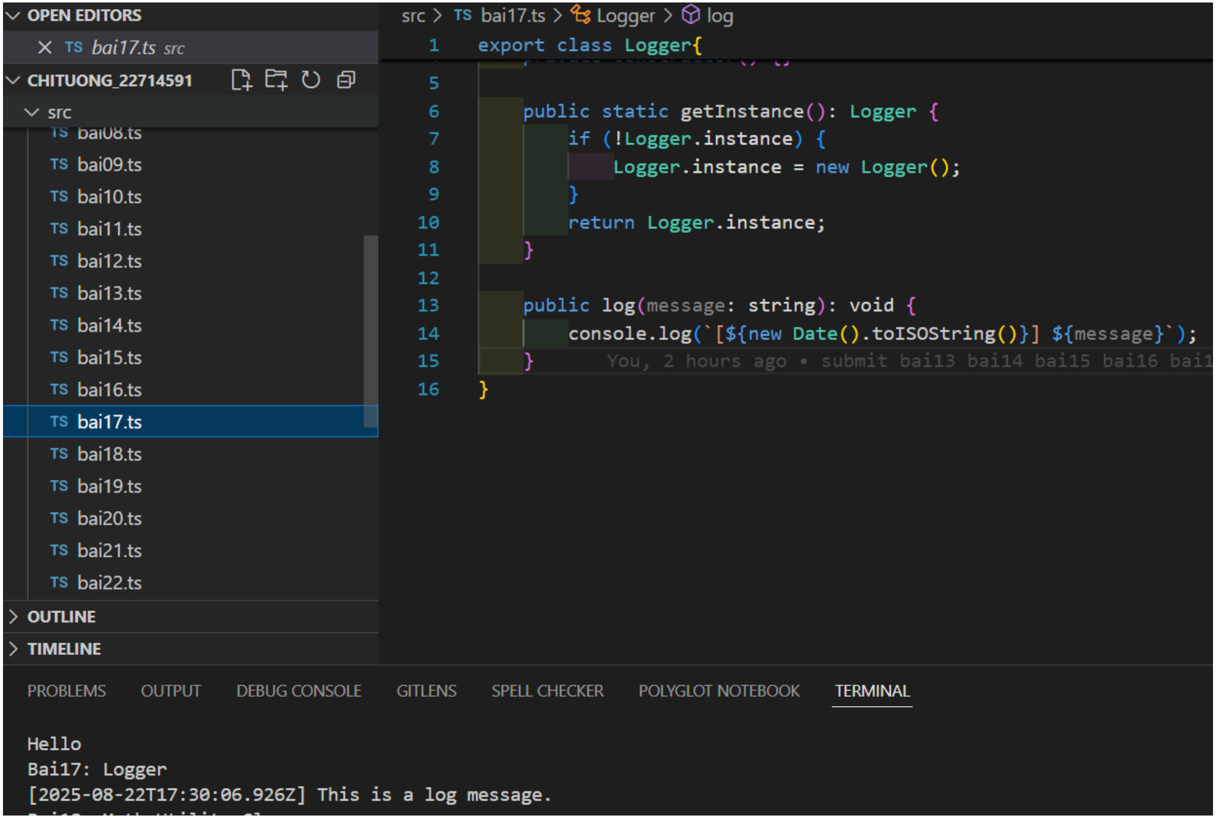
16. Create a generic class Box that can store any type of value.



```typescript
export class Box<T> {
    private value: T;

    constructor(value: T) {
        this.value = value;
    }

    public getValue(): T {
        return this.value;
    }

    public setValue(newValue: T): void {
        this.value = newValue;
    }
}
```

```
LamChiTuong
Bai16: Generic Class
123
Hello
```

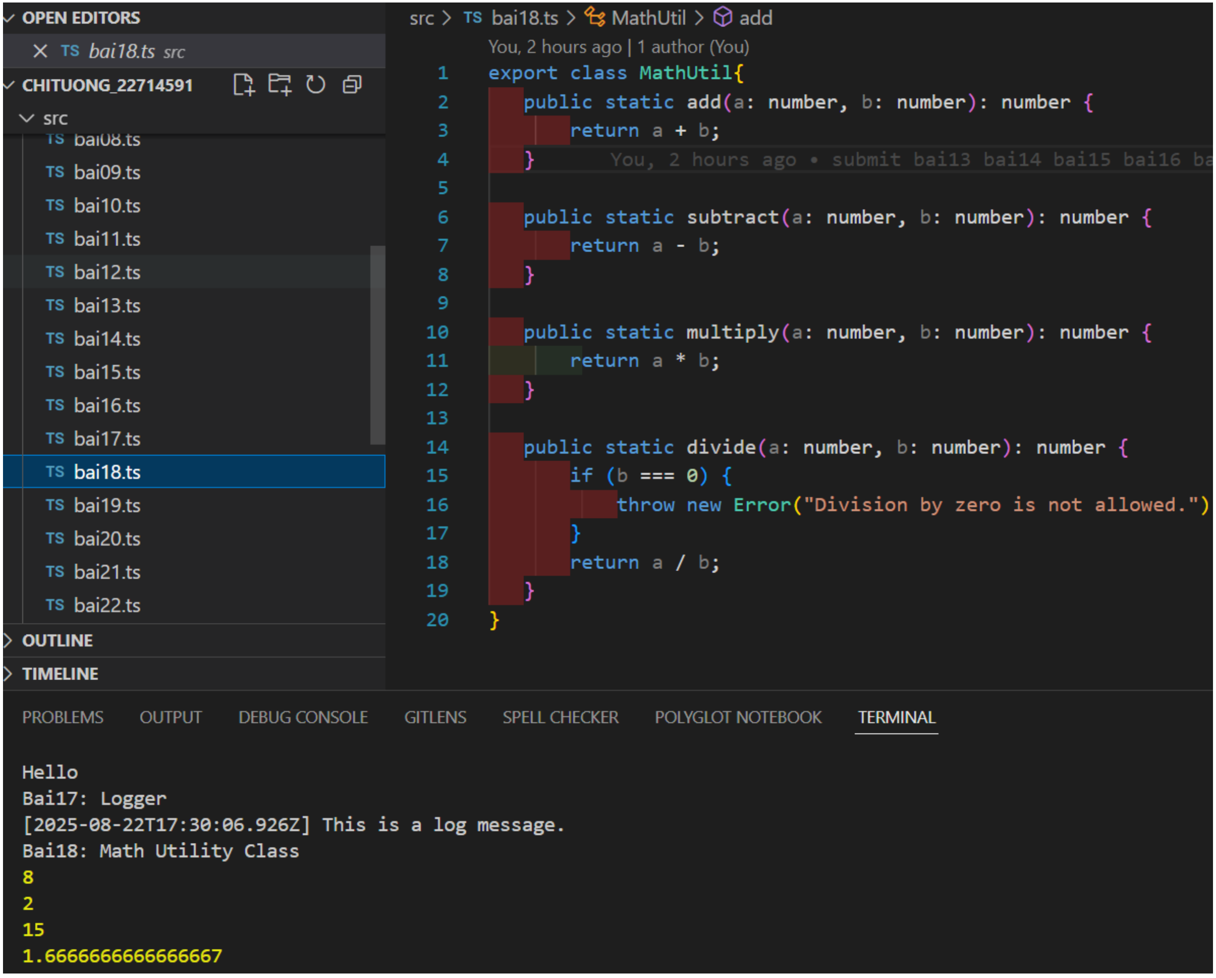
17. Write a singleton Logger class that logs messages to console.



```typescript
export class Logger{

    public static getInstance(): Logger {
        if (!Logger.instance) {
            Logger.instance = new Logger();
        }
        return Logger.instance;
    }

    public log(message: string): void {
        console.log(`[${new Date().toISOString()}] ${message}`);
    }
}
```

```
Hello
Bai17: Logger
[2025-08-22T17:30:06.926Z] This is a log message.
```

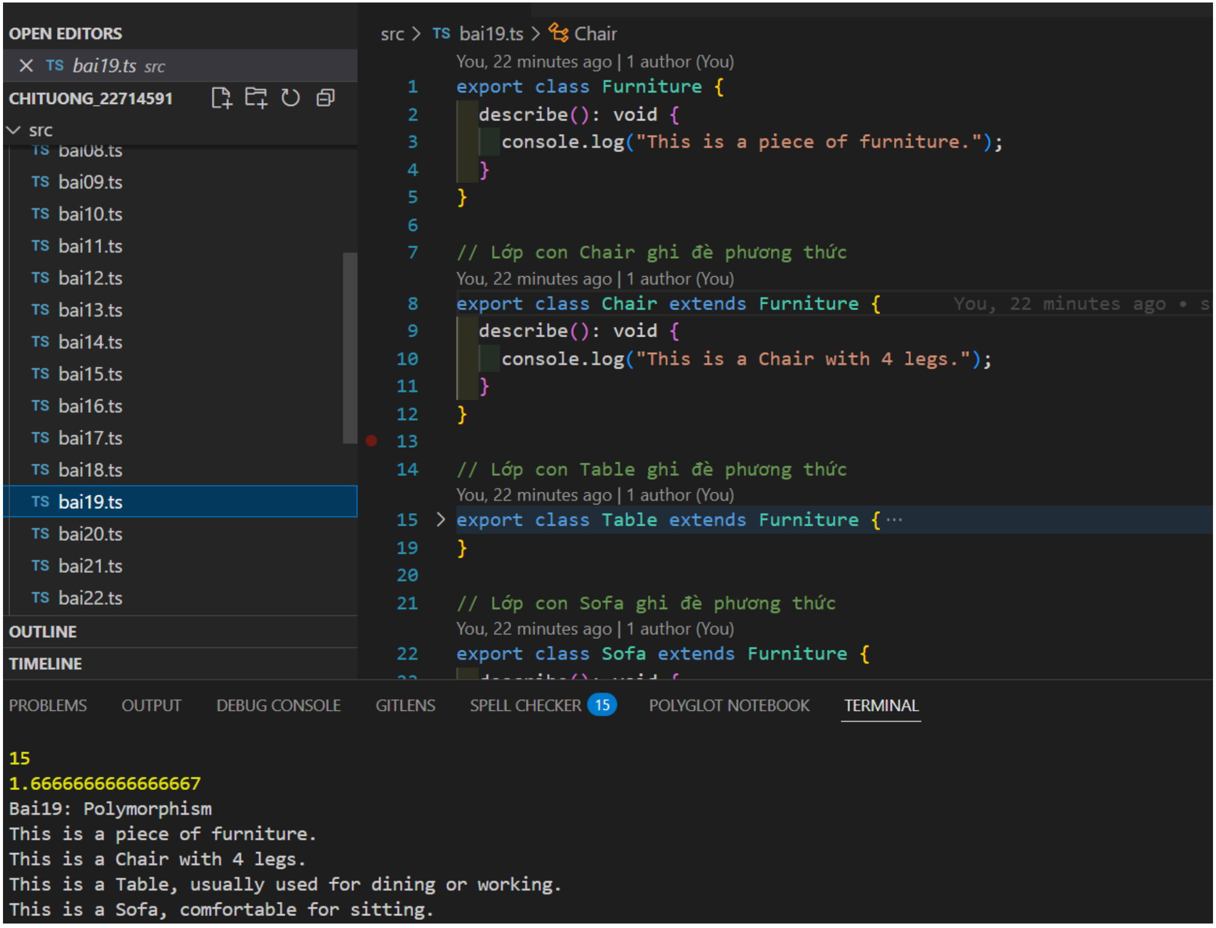
18. Create a static class MathUtil with methods add(), subtract(), multiply(), divide().



```
export class MathUtil{
    public static add(a: number, b: number): number {
        return a + b;
    }       You, 2 hours ago • submit bai13 bai14 bai15 bai16 ba

    public static subtract(a: number, b: number): number {
        return a - b;
    }

    public static multiply(a: number, b: number): number {
        return a * b;
    }

    public static divide(a: number, b: number): number {
        if (b === 0) {
            throw new Error("Division by zero is not allowed.")
        }
        return a / b;
    }
}
```

```
Hello
Bai17: Logger
[2025-08-22T17:30:06.926Z] This is a log message.
Bai18: Math Utility Class
8
2
15
1.6666666666666667
```

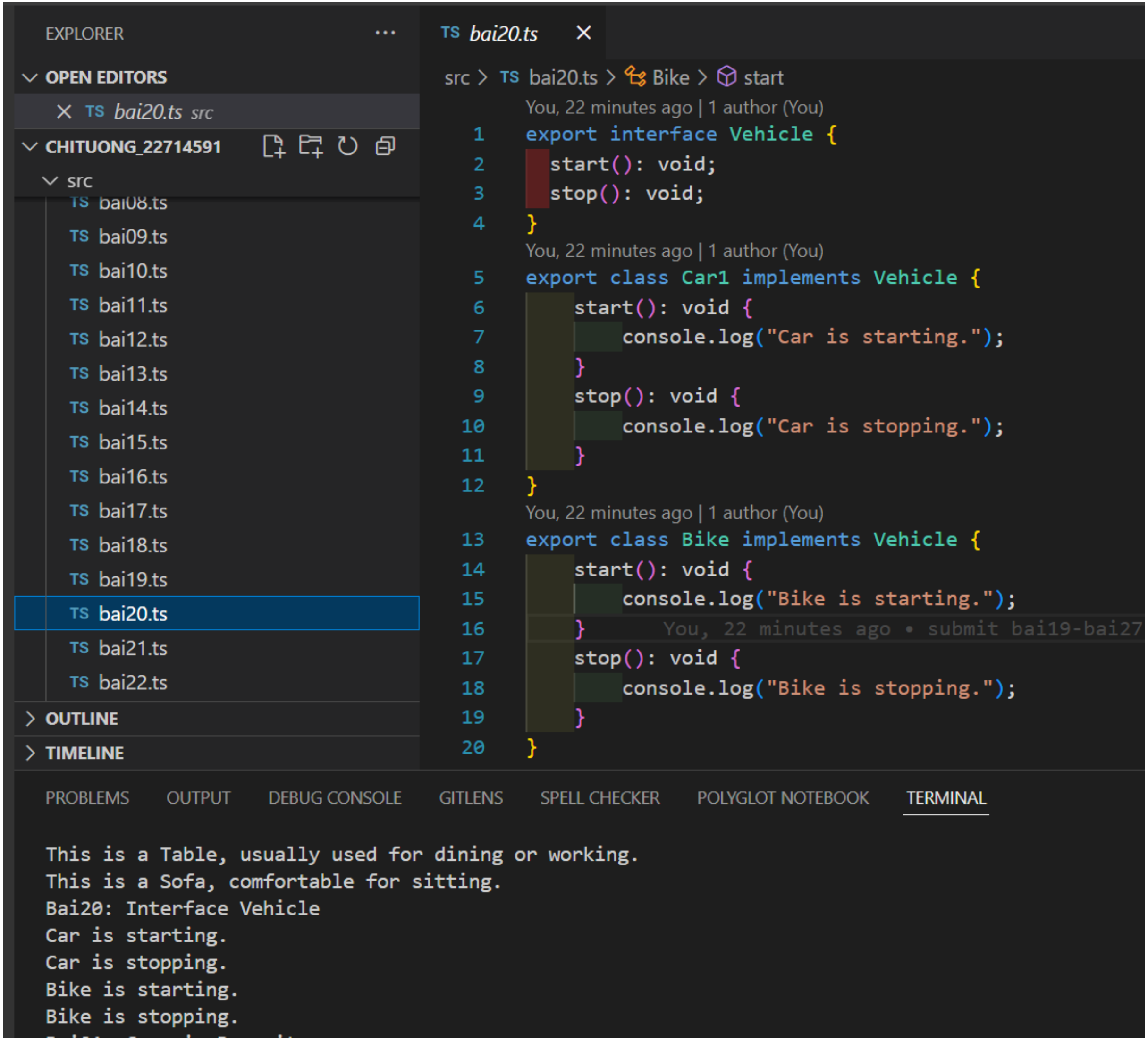
19. Demonstrate method overriding using polymorphism with Animal and subclasses.



```typescript
export class Furniture {
  describe(): void {
    console.log("This is a piece of furniture.");
  }
}

// Lớp con Chair ghi đè phương thức
export class Chair extends Furniture {
  describe(): void {
    console.log("This is a Chair with 4 legs.");
  }
}

// Lớp con Table ghi đè phương thức
export class Table extends Furniture {···
}

// Lớp con Sofa ghi đè phương thức
export class Sofa extends Furniture {
```

```
15
1.6666666666666667
Bai19: Polymorphism
This is a piece of furniture.
This is a Chair with 4 legs.
This is a Table, usually used for dining or working.
This is a Sofa, comfortable for sitting.
```

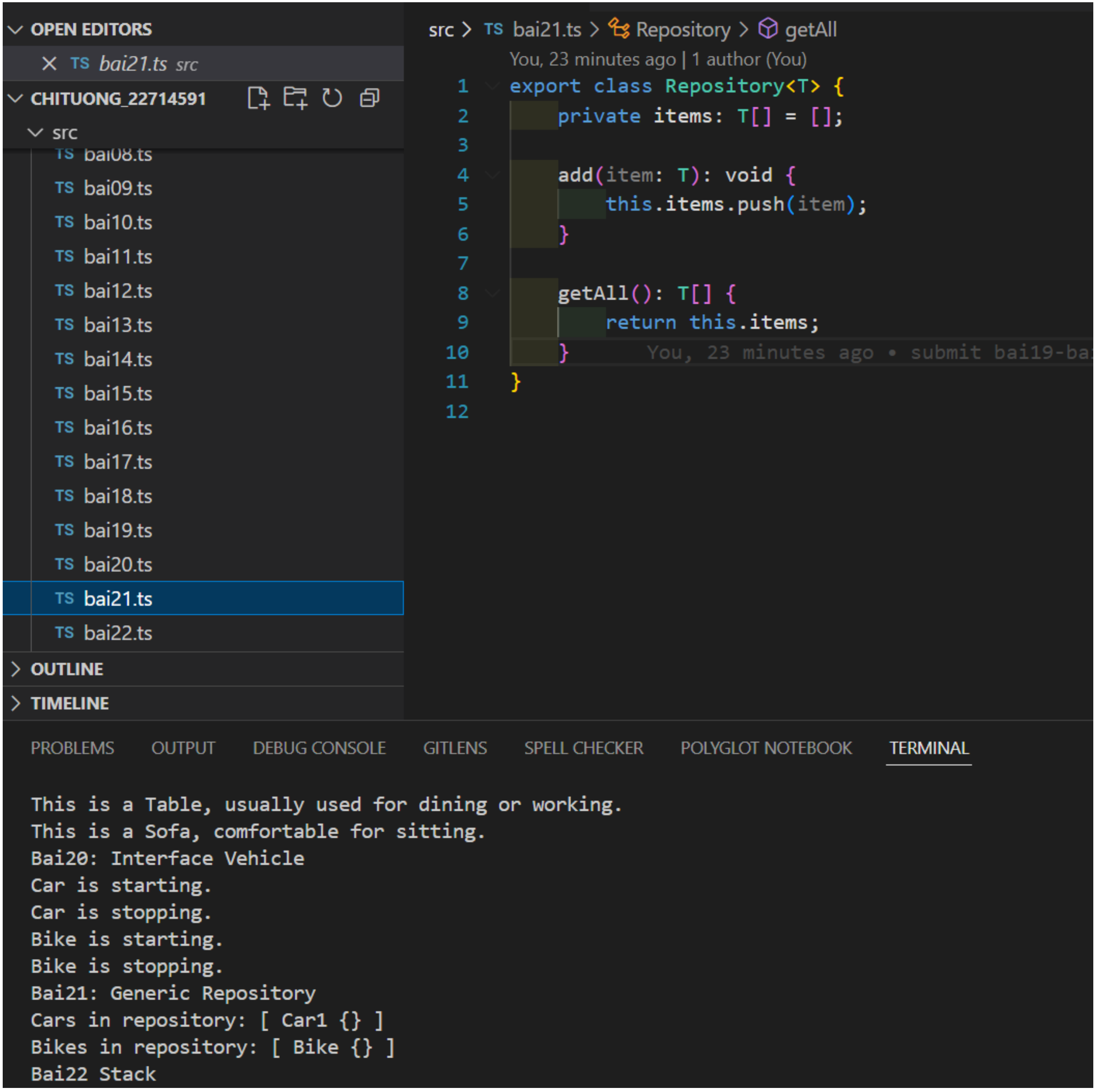
20. Write a Vehicle interface and implement it in Car and Bike classes.



```typescript
export interface Vehicle {
  start(): void;
  stop(): void;
}
export class Car1 implements Vehicle {
    start(): void {
        console.log("Car is starting.");
    }
    stop(): void {
        console.log("Car is stopping.");
    }
}
export class Bike implements Vehicle {
    start(): void {
        console.log("Bike is starting.");
    }
    stop(): void {
        console.log("Bike is stopping.");
    }
}
```

```
This is a Table, usually used for dining or working.
This is a Sofa, comfortable for sitting.
Bai20: Interface Vehicle
Car is starting.
Car is stopping.
Bike is starting.
Bike is stopping.
```

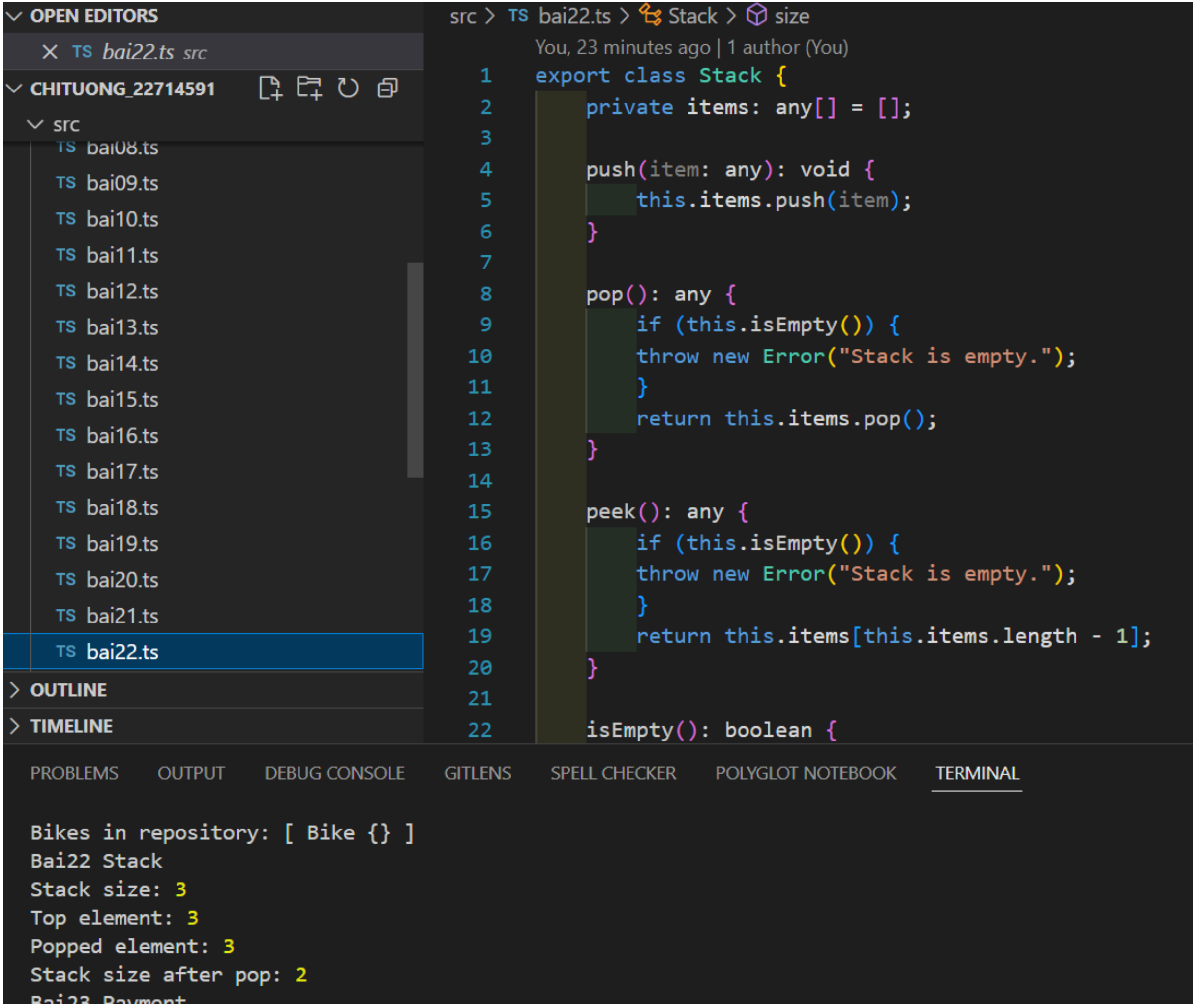
21. Create a generic Repository class with methods add(), getAll().



```
export class Repository<T> {
    private items: T[] = [];

    add(item: T): void {
        this.items.push(item);
    }

    getAll(): T[] {
        return this.items;
    }       You, 23 minutes ago • submit bai19-ba
}
```

```
This is a Table, usually used for dining or working.
This is a Sofa, comfortable for sitting.
Bai20: Interface Vehicle
Car is starting.
Car is stopping.
Bike is starting.
Bike is stopping.
Bai21: Generic Repository
Cars in repository: [ Car1 {} ]
Bikes in repository: [ Bike {} ]
Bai22 Stack
```

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

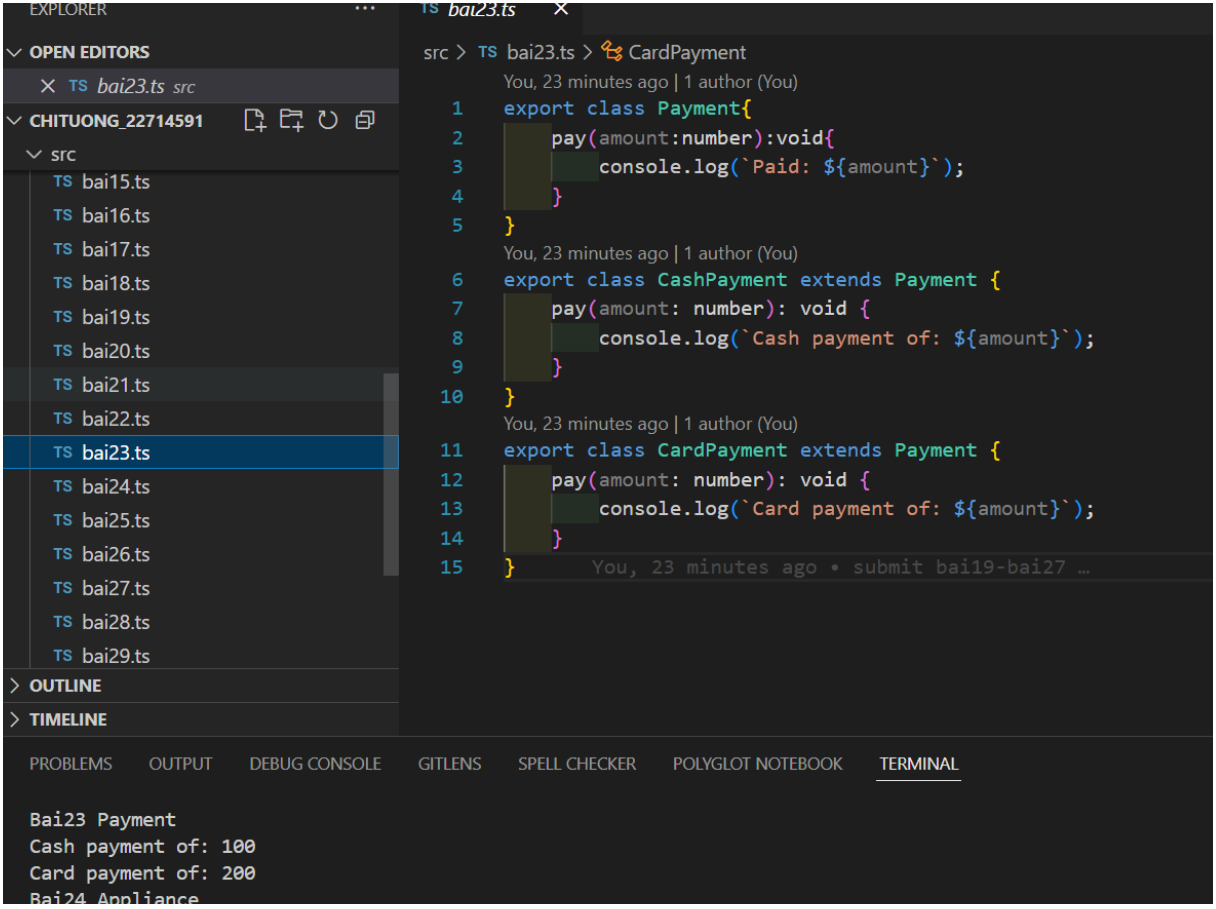


```typescript
export class Stack {
    private items: any[] = [];

    push(item: any): void {
        this.items.push(item);
    }

    pop(): any {
        if (this.isEmpty()) {
        throw new Error("Stack is empty.");
        }
        return this.items.pop();
    }

    peek(): any {
        if (this.isEmpty()) {
        throw new Error("Stack is empty.");
        }
        return this.items[this.items.length - 1];
    }

    isEmpty(): boolean {
```

```
Bikes in repository: [ Bike {} ]
Bai22 Stack
Stack size: 3
Top element: 3
Popped element: 3
Stack size after pop: 2
Bai23 Payment
```

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

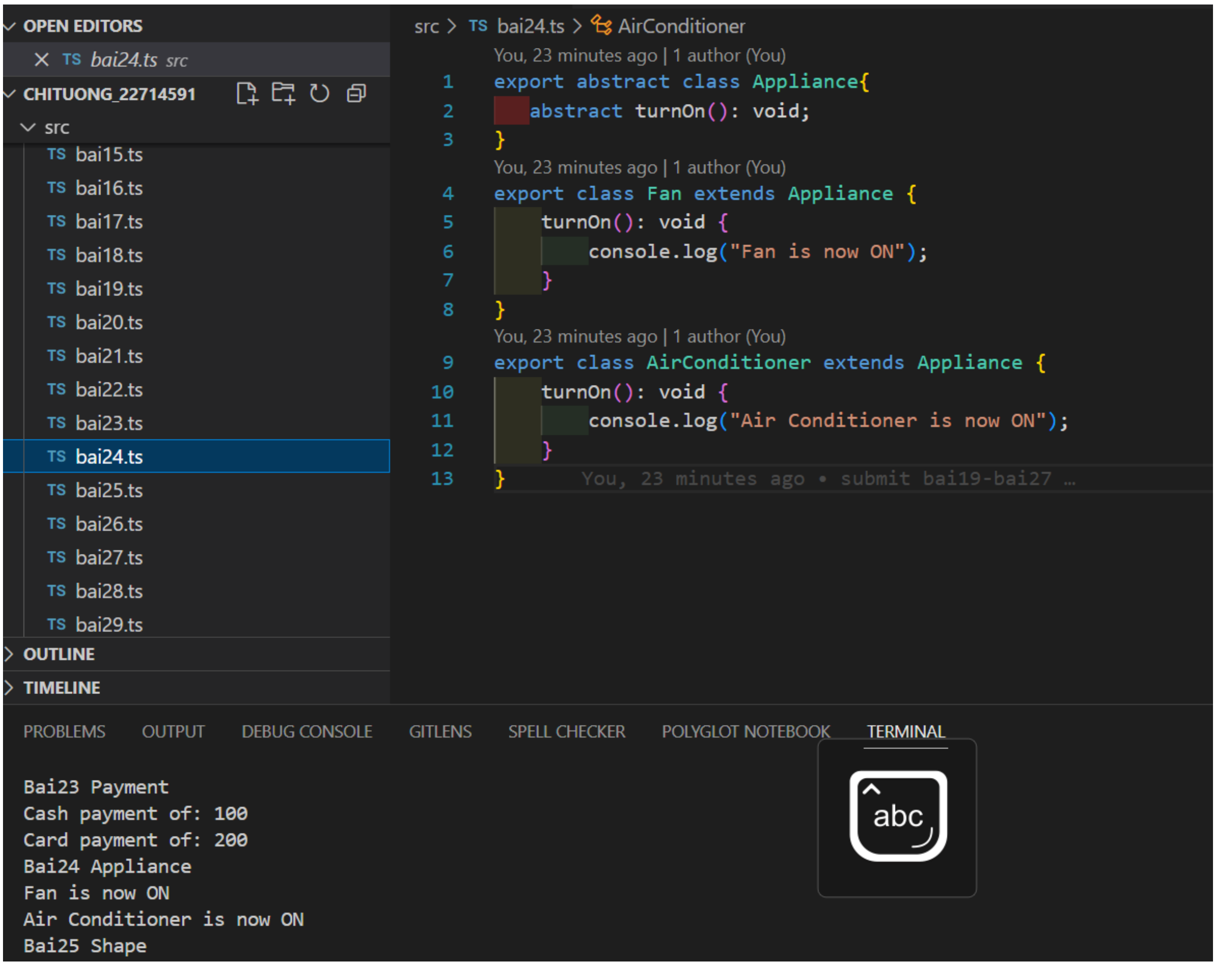
CardPayment.



```typescript
export class Payment{
    pay(amount:number):void{
        console.log(`Paid: ${amount}`);
    }
}
export class CashPayment extends Payment {
    pay(amount: number): void {
        console.log(`Cash payment of: ${amount}`);
    }
}
export class CardPayment extends Payment {
    pay(amount: number): void {
        console.log(`Card payment of: ${amount}`);
    }
}       You, 23 minutes ago • submit bai19-bai27 …
```

```
Bai23 Payment
Cash payment of: 100
Card payment of: 200
Bai24 Appliance
```

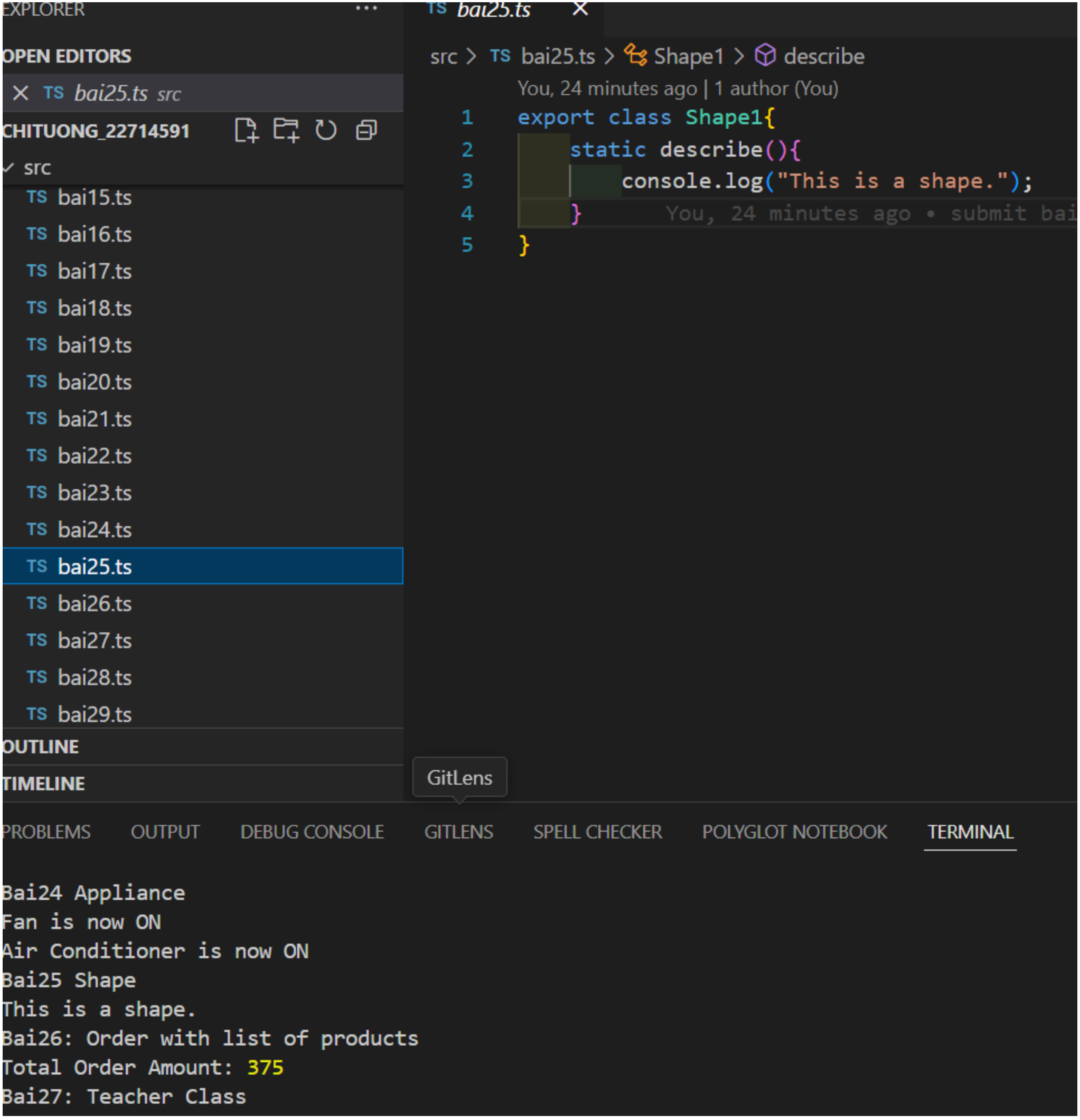
24. Create an abstract class Appliance with method turnOn(). Implement Fan and AirConditioner.



```
You, 23 minutes ago | 1 author (You)
export abstract class Appliance{
    abstract turnOn(): void;
}
You, 23 minutes ago | 1 author (You)
export class Fan extends Appliance {
    turnOn(): void {
        console.log("Fan is now ON");
    }
}
You, 23 minutes ago | 1 author (You)
export class AirConditioner extends Appliance {
    turnOn(): void {
        console.log("Air Conditioner is now ON");
    }
}       You, 23 minutes ago • submit bai19-bai27 …
```

```
Bai23 Payment
Cash payment of: 100
Card payment of: 200
Bai24 Appliance
Fan is now ON
Air Conditioner is now ON
Bai25 Shape
```

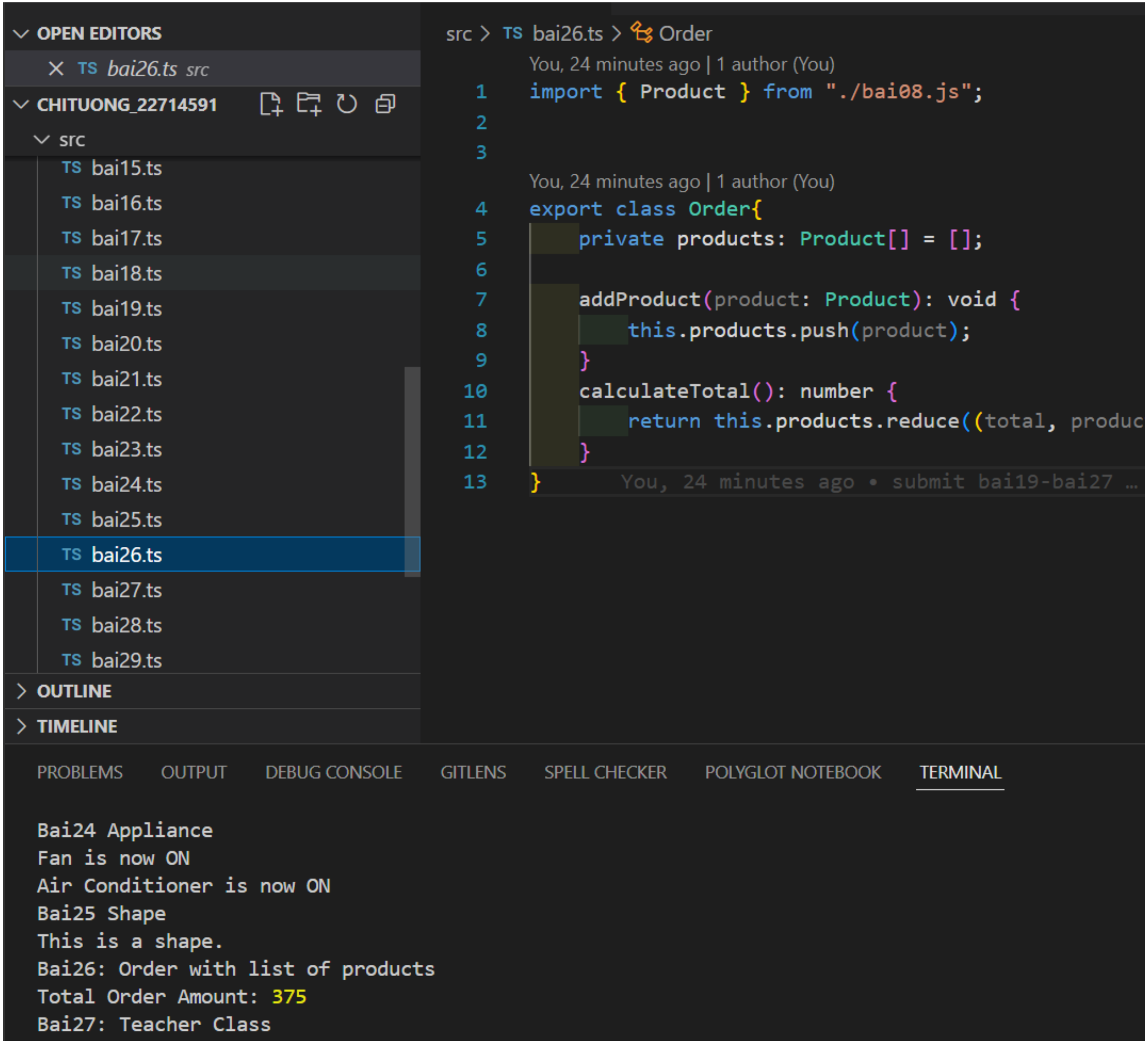
25. Create a class Shape with a static method describe().



```
export class Shape1{
    static describe(){
        console.log("This is a shape.");
    }        You, 24 minutes ago • submit bai
}
```

```
Bai24 Appliance
Fan is now ON
Air Conditioner is now ON
Bai25 Shape
This is a shape.
Bai26: Order with list of products
Total Order Amount: 375
Bai27: Teacher Class
```

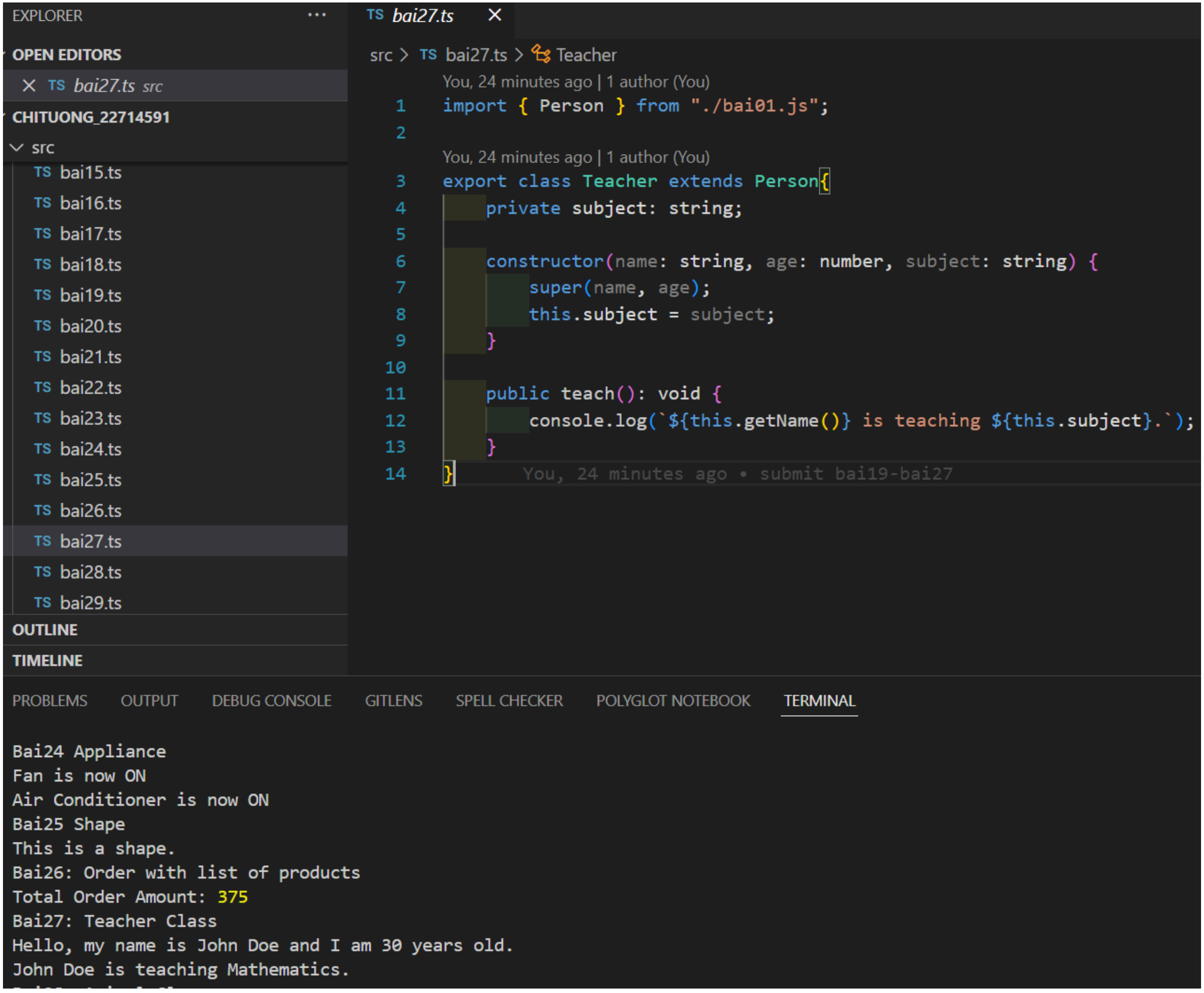
26. Create a class Order with list of products. Add method to calculate total price.



```
import { Product } from "./bai08.js";


export class Order{
    private products: Product[] = [];

    addProduct(product: Product): void {
        this.products.push(product);
    }
    calculateTotal(): number {
        return this.products.reduce((total, produc
    }
}
```

```
Bai24 Appliance
Fan is now ON
Air Conditioner is now ON
Bai25 Shape
This is a shape.
Bai26: Order with list of products
Total Order Amount: 375
Bai27: Teacher Class
```

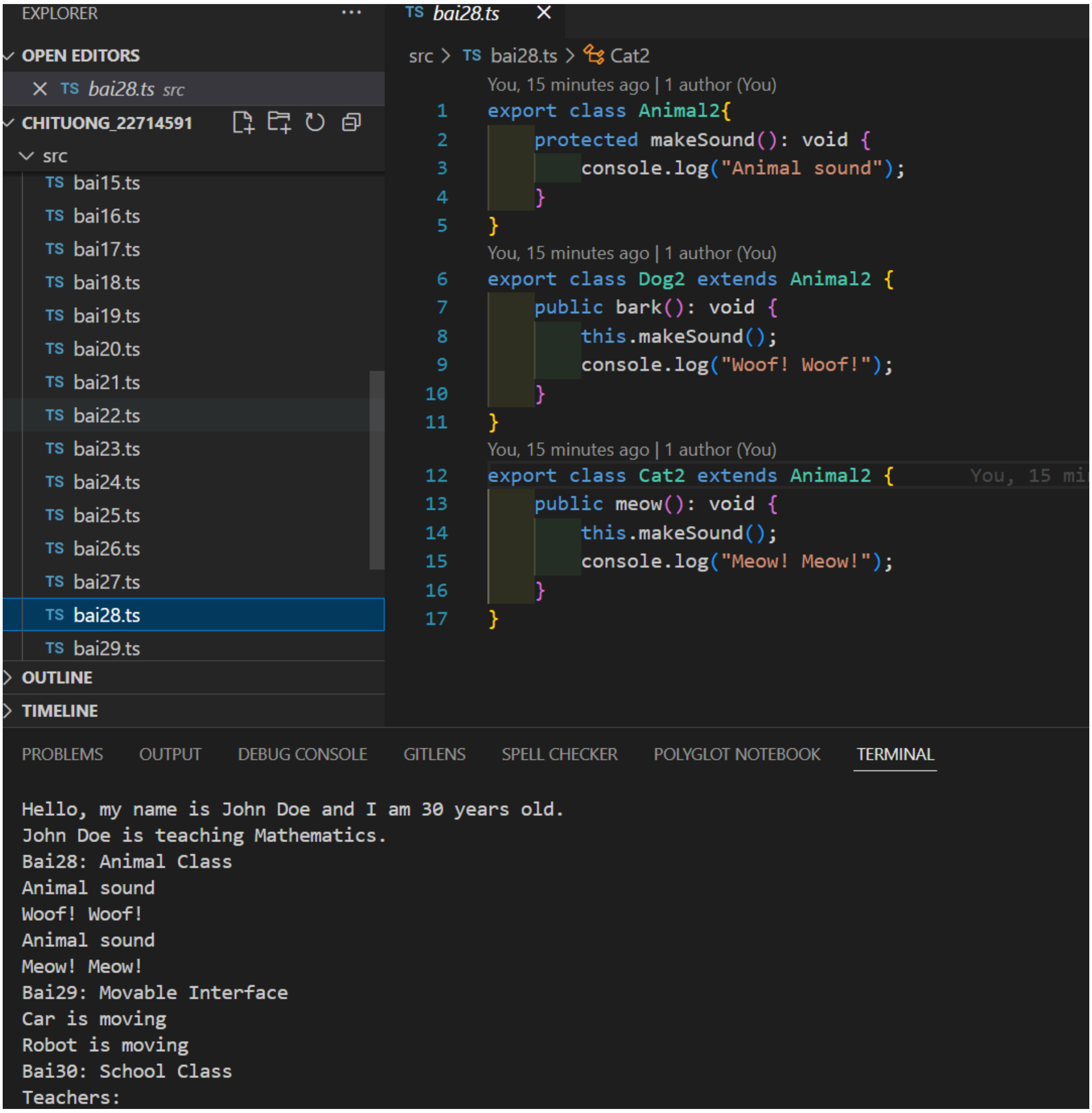
27. Create a class Teacher that extends Person. Add subject attribute and introduce method.



```typescript
import { Person } from "./bai01.js";

export class Teacher extends Person{
    private subject: string;

    constructor(name: string, age: number, subject: string) {
        super(name, age);
        this.subject = subject;
    }

    public teach(): void {
        console.log(`${this.getName()} is teaching ${this.subject}.`);
    }
}
```

```
Bai24 Appliance
Fan is now ON
Air Conditioner is now ON
Bai25 Shape
This is a shape.
Bai26: Order with list of products
Total Order Amount: 375
Bai27: Teacher Class
Hello, my name is John Doe and I am 30 years old.
John Doe is teaching Mathematics.
```

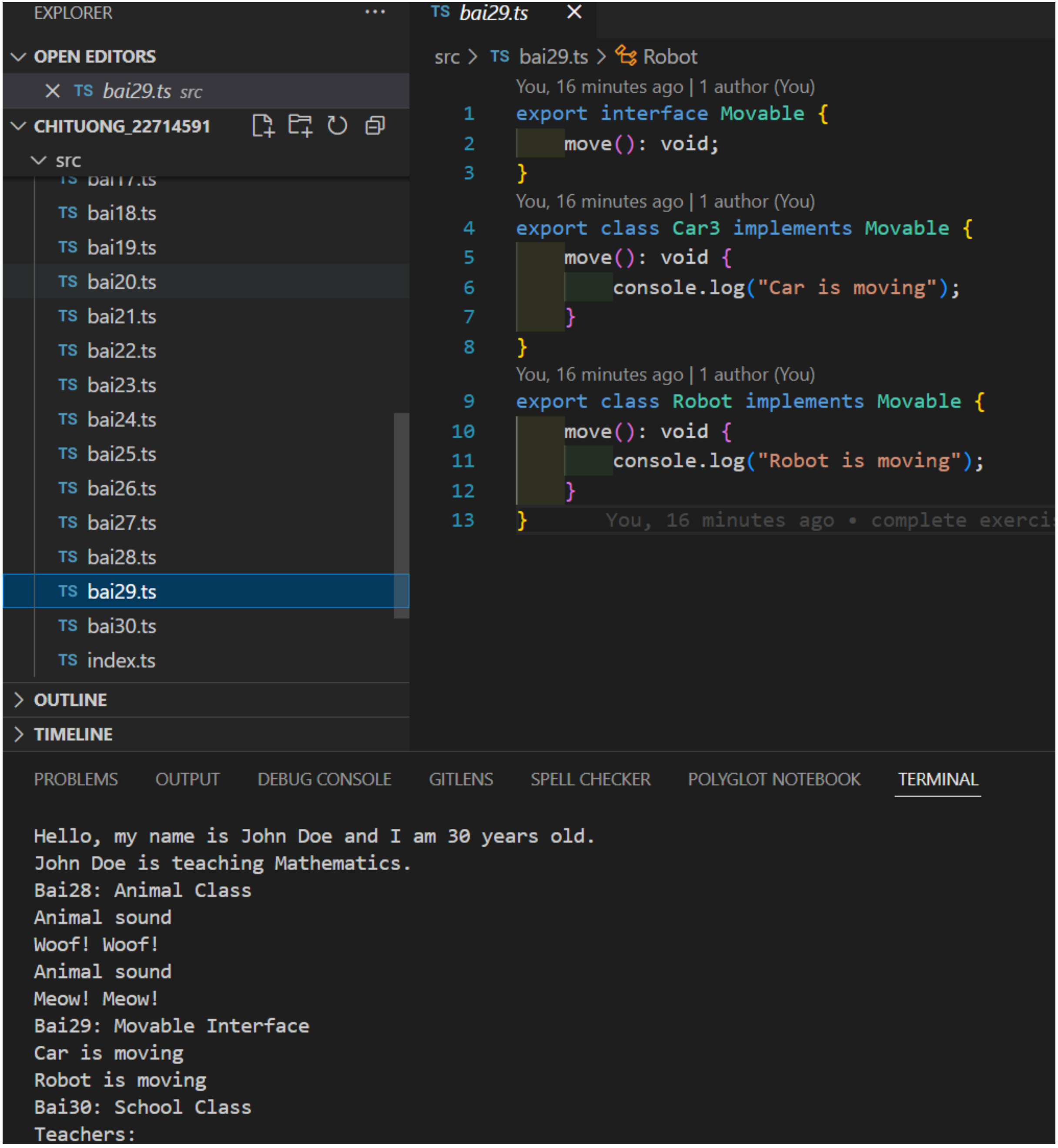
28. Create a class Animal with protected method makeSound(). Extend Dog and Cat to override it.



```
You, 15 minutes ago | 1 author (You)
export class Animal2{
    protected makeSound(): void {
        console.log("Animal sound");
    }
}
You, 15 minutes ago | 1 author (You)
export class Dog2 extends Animal2 {
    public bark(): void {
        this.makeSound();
        console.log("Woof! Woof!");
    }
}
You, 15 minutes ago | 1 author (You)
export class Cat2 extends Animal2 {          You, 15 mi
    public meow(): void {
        this.makeSound();
        console.log("Meow! Meow!");
    }
}
```

```
Hello, my name is John Doe and I am 30 years old.
John Doe is teaching Mathematics.
Bai28: Animal Class
Animal sound
Woof! Woof!
Animal sound
Meow! Meow!
Bai29: Movable Interface
Car is moving
Robot is moving
Bai30: School Class
Teachers:
```

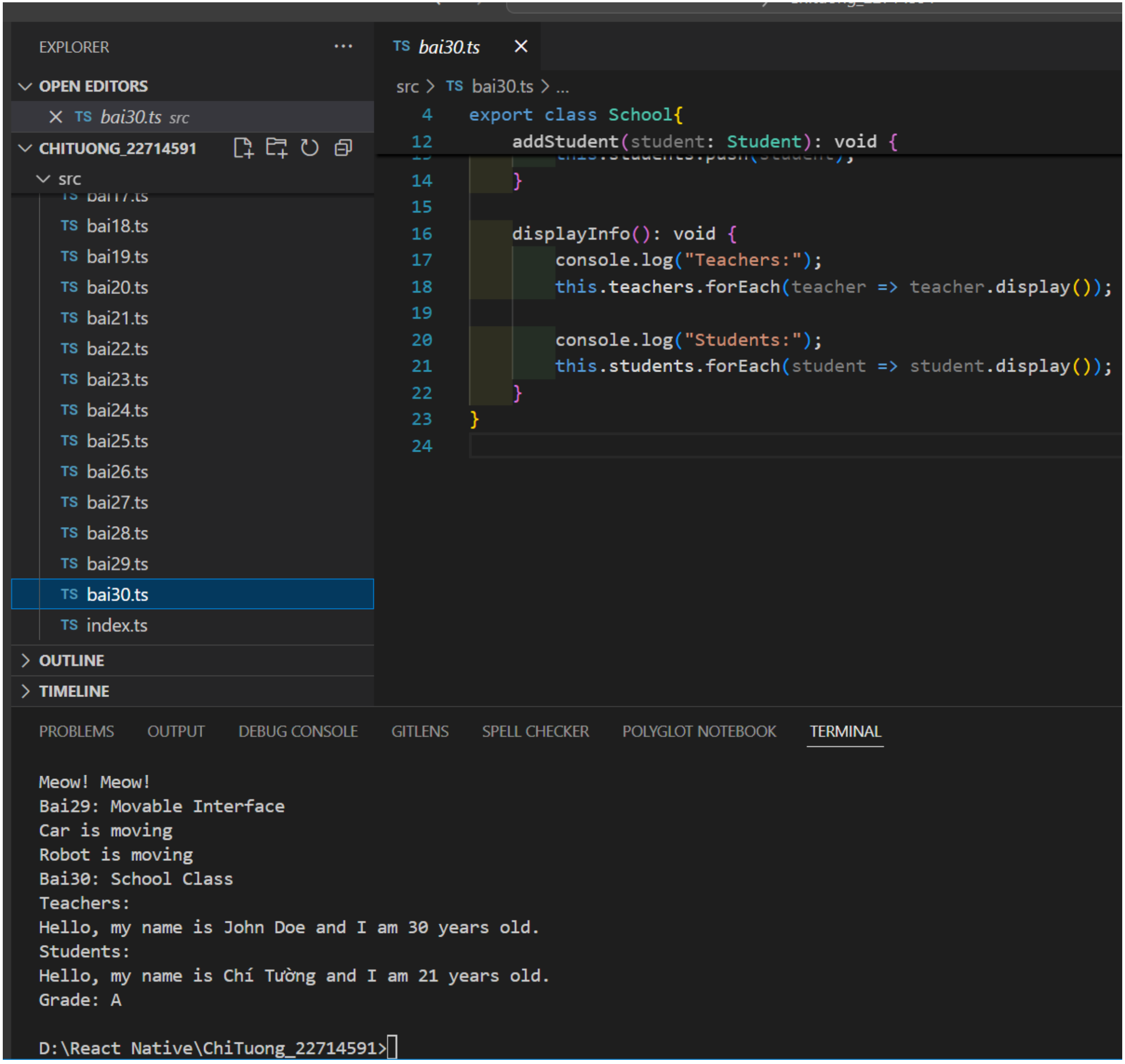
29. Create an interface Movable with method move(). Implement it in Car and Robot.



```
You, 16 minutes ago | 1 author (You)
export interface Movable {
    move(): void;
}
You, 16 minutes ago | 1 author (You)
export class Car3 implements Movable {
    move(): void {
        console.log("Car is moving");
    }
}
You, 16 minutes ago | 1 author (You)
export class Robot implements Movable {
    move(): void {
        console.log("Robot is moving");
    }
}        You, 16 minutes ago • complete exerci
```

```
Hello, my name is John Doe and I am 30 years old.
John Doe is teaching Mathematics.
Bai28: Animal Class
Animal sound
Woof! Woof!
Animal sound
Meow! Meow!
Bai29: Movable Interface
Car is moving
Robot is moving
Bai30: School Class
Teachers:
```

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



```
export class School{
    addStudent(student: Student): void {
        this.students.push(student);
    }

    displayInfo(): void {
        console.log("Teachers:");
        this.teachers.forEach(teacher => teacher.display());

        console.log("Students:");
        this.students.forEach(student => student.display());
    }
}
```

```
Meow! Meow!
Bai29: Movable Interface
Car is moving
Robot is moving
Bai30: School Class
Teachers:
Hello, my name is John Doe and I am 30 years old.
Students:
Hello, my name is Chí Tường and I am 21 years old.
Grade: A

D:\React Native\ChiTuong_22714591>
```