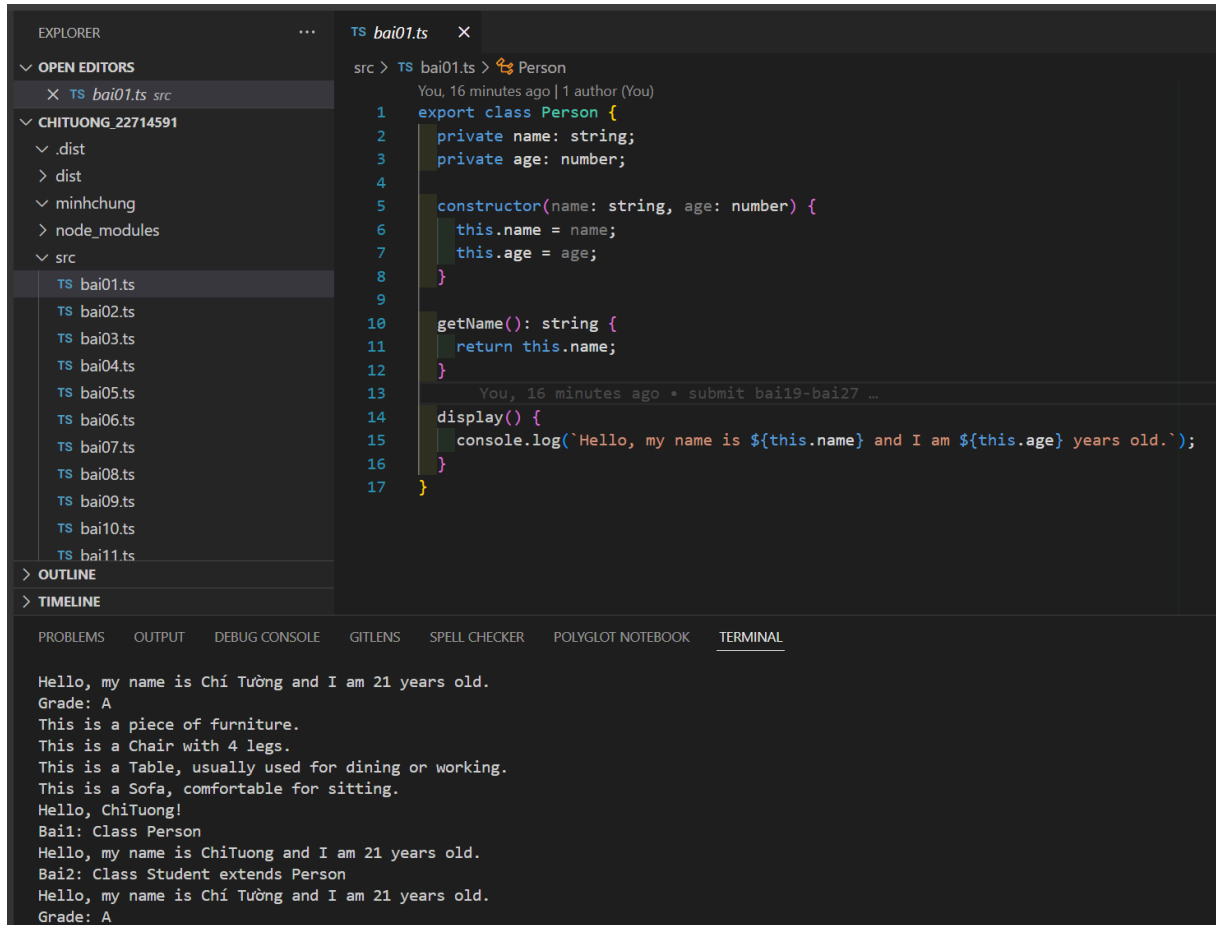


Minh Chứng BT React Native Lab01

Nguyễn Lâm Chí Tường

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



```
EXPLORER
  OPEN EDITORS
    TS bai01.ts src
  CHITUONG_22714591
    .dist
    dist
    minhchung
    node_modules
    src
      TS bai01.ts
      TS bai02.ts
      TS bai03.ts
      TS bai04.ts
      TS bai05.ts
      TS bai06.ts
      TS bai07.ts
      TS bai08.ts
      TS bai09.ts
      TS bai10.ts
      TS bai11.ts
    OUTLINE
    TIMELINE
  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  GITLENS
  SPELL CHECKER
  POLYGLOT NOTEBOOK
  TERMINAL

src > TS bai01.ts > Person
You, 16 minutes ago | 1 author (You)
1 export class Person {
2   private name: string;
3   private age: number;
4
5   constructor(name: string, age: number) {
6     this.name = name;
7     this.age = age;
8   }
9
10  getName(): string {
11    return this.name;
12  }
13
14  display() {
15    console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
16  }
17 }
```

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.

The image shows a screenshot of the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a project structure with a folder named 'src' containing several TypeScript files. The file 'bai02.ts' is selected and open in the editor. The editor shows the following code:

```
src > TS bai02.ts > Student > grade
You, 2 hours ago | 1 author (You)
1 import { Person } from "../bai01.js";
2
You, 14 hours ago | 1 author (You)
3 export class Student extends Person {
4     grade: string;
5
6     constructor(name: string, age: number, grade: string) {
7         super(name, age);
8         this.grade = grade;
9     }
10
11     display(): void {
12         super.display();
13         console.log(`Grade: ${this.grade}`);
14     }
15 }
16
17 const student = new Student("Chí Tường", 21, "A");
18 student.display();
19
```

The bottom panel shows the TERMINAL output, which displays the following text:

```
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.

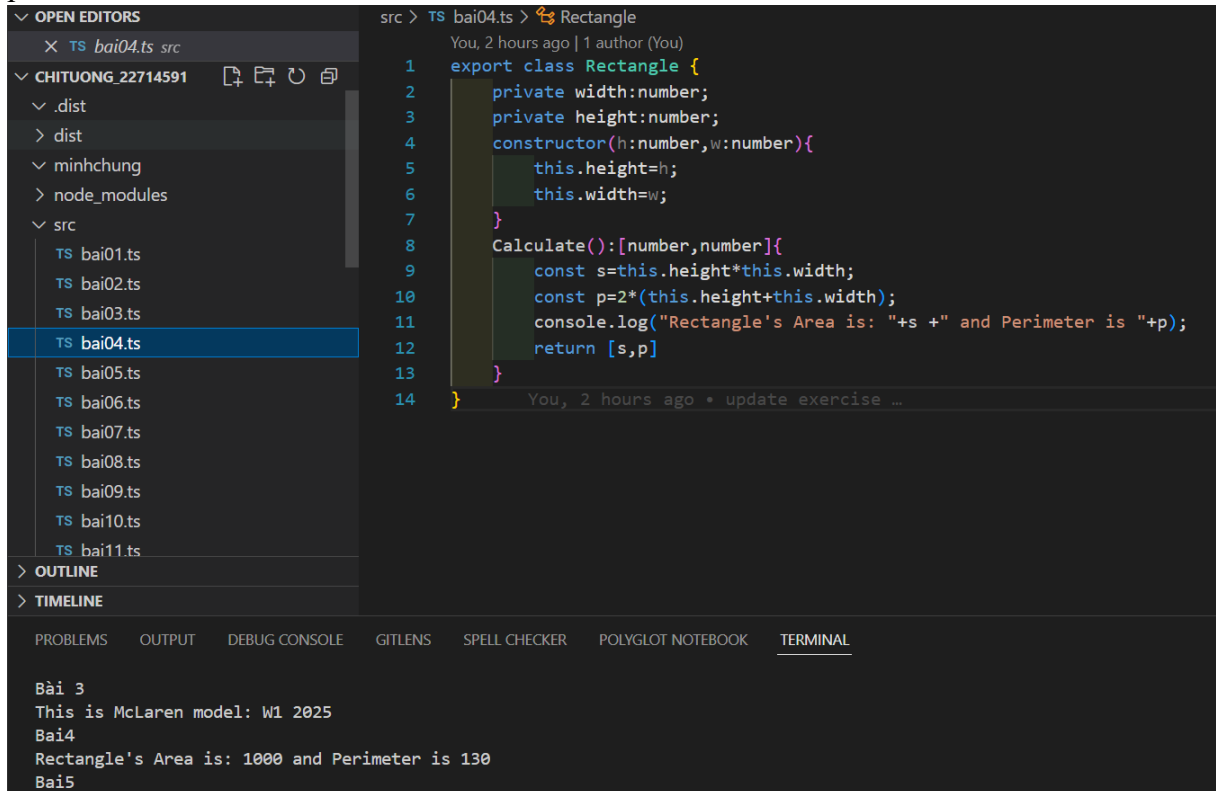
The image shows a VS Code editor with a TypeScript file named `bai03.ts` open. The file contains the following code:

```
src > TS bai03.ts > Car > constructor
You, 2 hours ago | 1 author (You)
1 export class Car {
2     private brand:string;
3     private model:string;
4     private year:number;
5     constructor(brand:string,model:string,year:number){
6         this.brand=brand;
7         this.model=model;
8         this.year=year;
9     }
10    display():void{
11        console.log(`This is ${this.brand} model: ${this
12    }
13 }
```

The terminal output at the bottom shows the results of running the program:

```
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.



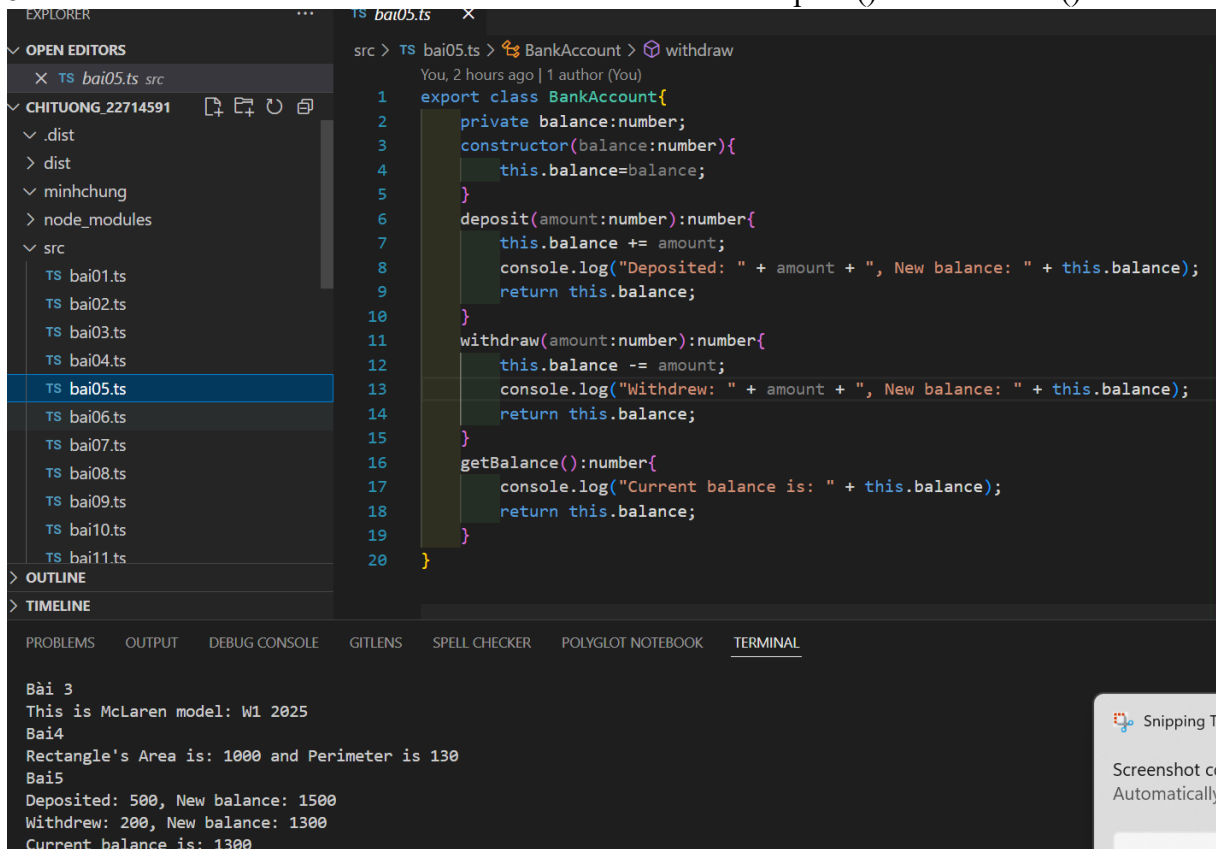
The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' folder containing files 'bai01.ts' through 'bai11.ts'. The file 'bai04.ts' is selected. The code editor shows the following TypeScript code for the 'Rectangle' class:

```
src > TS bai04.ts > Rectangle
You, 2 hours ago | 1 author (You)
1 export class Rectangle {
2     private width:number;
3     private height:number;
4     constructor(h:number,w:number){
5         this.height=h;
6         this.width=w;
7     }
8     Calculate():[number,number]{
9         const s=this.height*this.width;
10        const p=2*(this.height+this.width);
11        console.log("Rectangle's Area is: "+s+" and Perimeter is "+p);
12        return [s,p]
13    }
14 }
```

The bottom of the screenshot shows the 'TERMINAL' tab with the following output:

```
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().



The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' folder containing files 'bai01.ts' through 'bai11.ts'. The file 'bai05.ts' is selected. The code editor shows the following TypeScript code for the 'BankAccount' class:

```
src > TS bai05.ts > BankAccount > withdraw
You, 2 hours ago | 1 author (You)
1 export class BankAccount{
2     private balance:number;
3     constructor(balance:number){
4         this.balance=balance;
5     }
6     deposit(amount:number):number{
7         this.balance += amount;
8         console.log("Deposited: " + amount + ", New balance: " + this.balance);
9         return this.balance;
10    }
11    withdraw(amount:number):number{
12        this.balance -= amount;
13        console.log("Withdrew: " + amount + ", New balance: " + this.balance);
14        return this.balance;
15    }
16    getBalance():number{
17        console.log("Current balance is: " + this.balance);
18        return this.balance;
19    }
20 }
```

The bottom of the screenshot shows the 'TERMINAL' tab with the following output:

```
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.

The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure with a file named `bai06.ts` selected. The main editor area displays the following TypeScript code:

```
src > TS bai06.ts > Book
You, 2 hours ago | 1 author (You)
1 export class Book{
2     private title:string;
3     private author:string;
4     private year:number;
5     constructor(title:string, author:string, year:number){
6         this.title = title;
7         this.author = author;
8         this.year = year;
9     }
10    display():void{
11        console.log(`Title: ${this.title}, Author: ${this.author}, Year: ${this.year}`);
12    }
13 }
```

At the bottom, the TERMINAL panel shows the following output:

```
Deposited: 500, New balance: 1500
Withdraw: 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.

The image shows a Visual Studio Code editor window with a TypeScript file named `bai07.ts` open. The file contains a `User` class with a private `name` property, a constructor, and `getName` and `setName` methods. The Explorer sidebar on the left shows a project structure with a `src` directory containing several `bai01.ts` through `bai11.ts` files. The Terminal at the bottom displays the output of a program, showing the creation of a `User` object and the listing of expensive products.

```
src > TS bai07.ts > User > getName
You, 2 hours ago | 1 author (You)
1 export class User{
2     private name:string;
3     constructor(name:string){
4         this.name=name;
5     }
6     public getName():string{
7         return this.name;
8     }
9     public setName(name:string):void{
10        this.name=name;
11    }
12 }
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

```
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.

The image shows a VS Code editor window with a TypeScript file named `bai08.ts` in the `src` directory. The file contains a `Product` class with the following code:

```
1 export class Product{
2     private name:string;
3     private price:number;
4     constructor(name:string,price:number){
5         this.name=name;
6         this.price=price;
7     }
8     public getName():string{
9         return this.name;
10    }
11    public getPrice():number{
12        return this.price;
13    }
14    public setName(name:string):void{
15        this.name=name;
16    }
17    public setPrice(price:number):void{
18        this.price=price;
19    }
20    public displayInfo(): void {
21        console.log(`Product Name: ${this.getName()}`);
22    }
23 }
```

The terminal output at the bottom shows the following text:

```
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().

The image shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'src' directory containing files 'bai01.ts' through 'bai11.ts'. The code editor shows the content of 'bai09.ts', which defines an 'Animal' interface and two classes, 'Cow' and 'Chicken', that implement it. The 'Animal' interface has a 'name' property and a 'makeSound()' method. The 'Cow' class has a 'name' property, a constructor that sets 'this.name', and a 'makeSound()' method that logs 'Moo!'. The 'Chicken' class has a 'name' property, a constructor that sets 'this.name', and a 'makeSound()' method that logs 'Cluck!'. The terminal at the bottom shows the output of the code: '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', and 'Buddy barks: Woof! Woof!'.

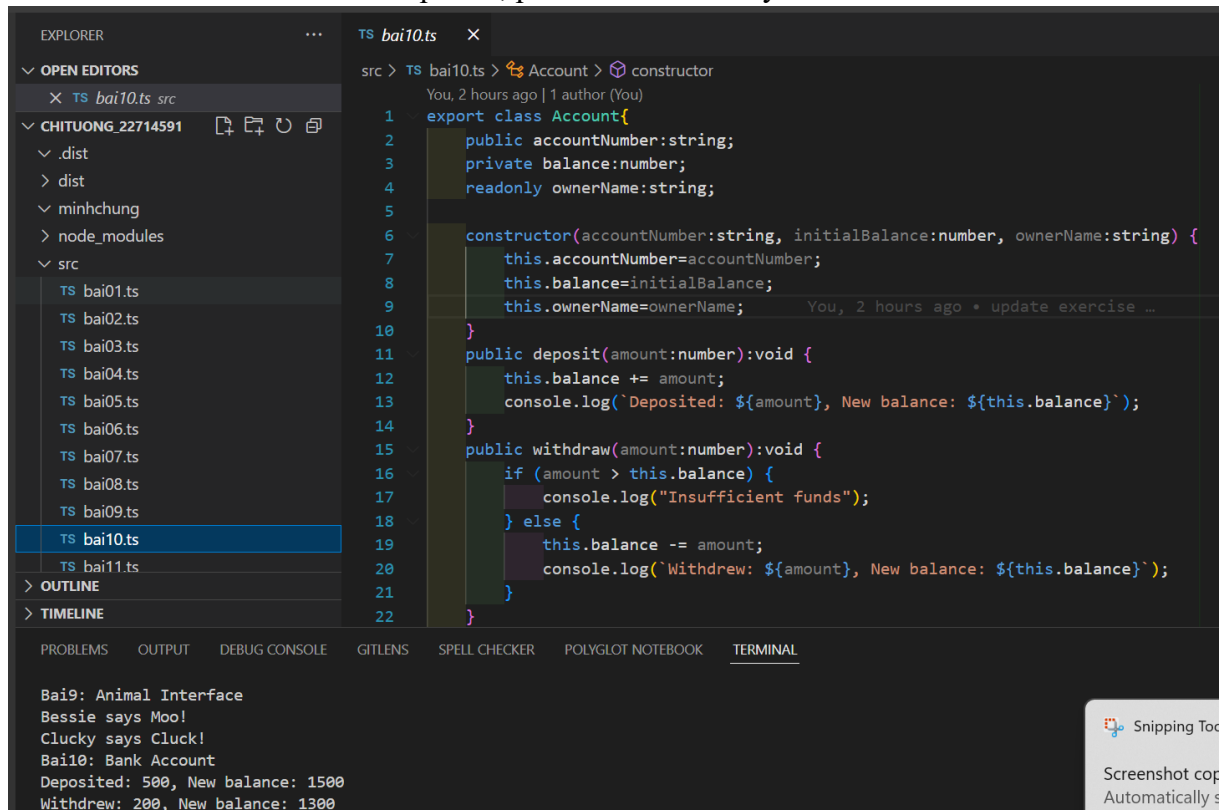
```
EXPLORER
  OPEN EDITORS
    TS bai09.ts src
  CHITUONG_22714591
    .dist
    dist
    minhchung
    node_modules
    src
      TS bai01.ts
      TS bai02.ts
      TS bai03.ts
      TS bai04.ts
      TS bai05.ts
      TS bai06.ts
      TS bai07.ts
      TS bai08.ts
      TS bai09.ts
      TS bai10.ts
      TS bai11.ts
    OUTLINE
    TIMELINE

  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  GITLENS
  SPELL CHECKER
  POLYGLOT NOTEBOOK
  TERMINAL

  TS bai09.ts
    src > TS bai09.ts > Chicken > makeSound
    You, 2 hours ago | 1 author (You)
    1 export interface Animal {
    2     name: string;
    3     makeSound(): void;
    4 }
    You, 2 hours ago | 1 author (You)
    5 export class Cow implements Animal {
    6     public name:string;
    7     constructor(name:string) {
    8         this.name=name;
    9     }
    10    public makeSound(): void {
    11        console.log(`${this.name} says Moo!`);
    12    }
    13 }
    You, 2 hours ago | 1 author (You)
    14 export class Chicken implements Animal {
    15     public name:string;
    16     constructor(name:string) {
    17         this.name=name;
    18     }
    19    public makeSound(): void {
    20        console.log(`${this.name} says Cluck!`);
    21    }
    22 }
```

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.



The screenshot shows a Visual Studio Code editor with a TypeScript file named `bai10.ts`. The file contains a class `Account` with the following code:

```
src > TS bai10.ts > Account > constructor
You, 2 hours ago | 1 author (You)
1 export class Account{
2     public accountNumber:string;
3     private balance:number;
4     readonly ownerName:string;
5
6     constructor(accountNumber:string, initialBalance:number, ownerName:string) {
7         this.accountNumber=accountNumber;
8         this.balance=initialBalance;
9         this.ownerName=ownerName;
10    }
11    public deposit(amount:number):void {
12        this.balance += amount;
13        console.log(`Deposited: ${amount}, New balance: ${this.balance}`);
14    }
15    public withdraw(amount:number):void {
16        if (amount > this.balance) {
17            console.log("Insufficient funds");
18        } else {
19            this.balance -= amount;
20            console.log(`Withdrew: ${amount}, New balance: ${this.balance}`);
21        }
22    }
}
```

The terminal output shows the following:

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

A Snipping Tool notification is visible in the bottom right corner, stating "Screenshot copied. Automatically saved to clipboard."

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

src > TS bai11.ts > Cat > meow

You, 3 hours ago | 1 author (You)

```
1 export class Animal{
2     public name:string;
3     constructor(name:string) {
4         this.name=name;
5     }
6 }
7
8 export class Dog extends Animal{
9     public bark(): void {
10         console.log(`${this.name} barks: Woof! Woof!`);
11     }
12 }
13
14 export class Cat extends Animal{
15     public meow(): void {
16         console.log(`${this.name} meows: Meow! Meow!`);
17     }
18 }
```

You 3 hours ago (August 22nd, 2024) update exercise

2af1225 | Changes added in 2af1225 | You, 3 hours ago • update exercise

public meow(): void {

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.

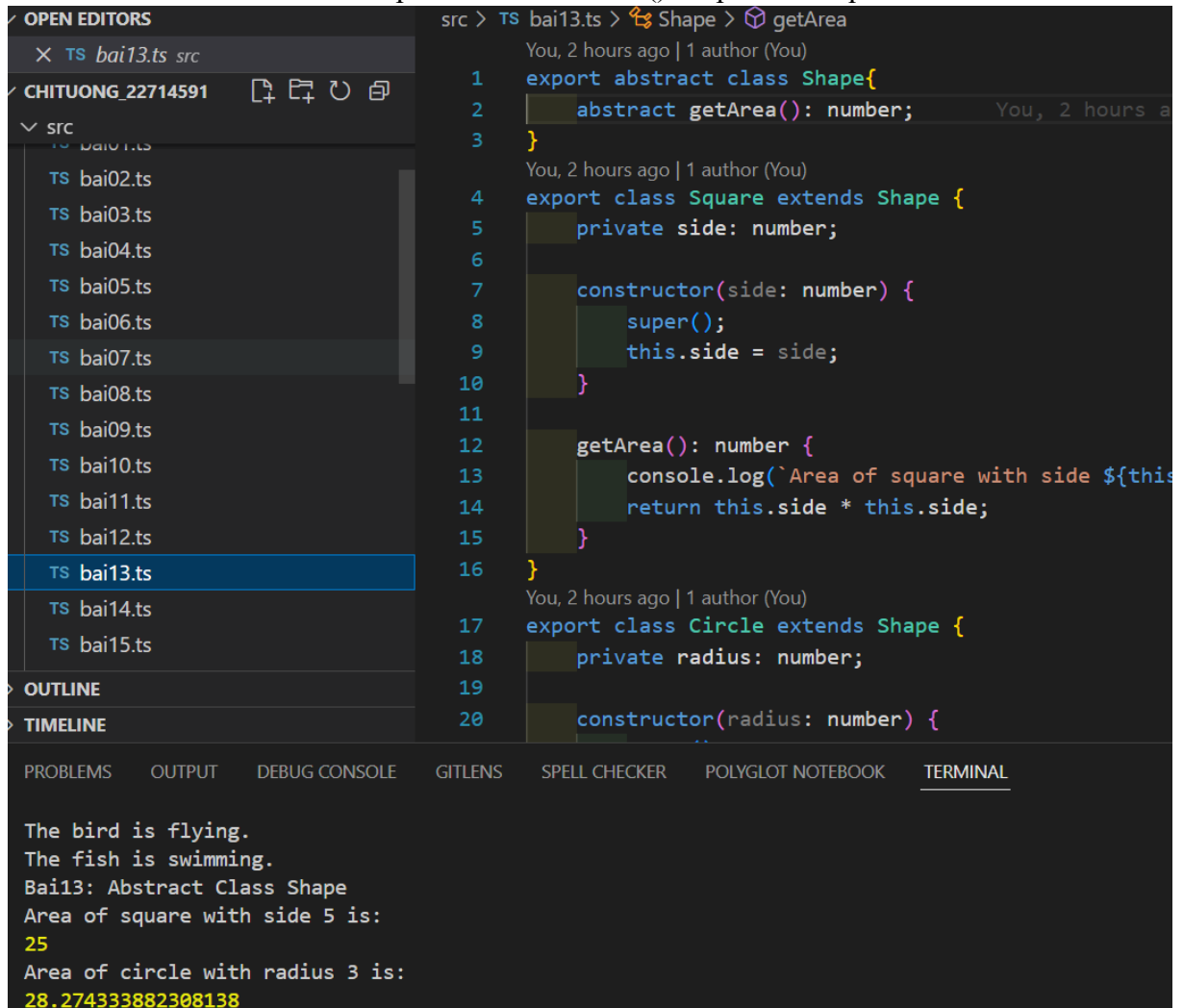
The screenshot shows the Visual Studio Code interface with the Explorer, Source Explorer, and Editor views. The Explorer view on the left shows a project named 'CHITUONG_22714591' with a 'src' folder containing several TypeScript files. The 'src' folder is expanded, showing files from 'bai01.ts' to 'bai15.ts'. The 'bai12.ts' file is selected and its content is displayed in the Editor view. The code defines two interfaces, 'Flyable' and 'Swimmable', and two classes, 'Bird' and 'Fish', which implement these interfaces. The 'Bird' class implements the 'Flyable' interface with a 'fly()' method that logs 'The bird is flying.' The 'Fish' class implements the 'Swimmable' interface with a 'swim()' method that logs 'The fish is swimming.' The terminal view at the bottom shows the output of the code execution: 'Whiskers meows: Meow! Meow!', 'Bai12: Abstract Class', 'The bird is flying.', and 'The fish is swimming.'.

```
src > TS bai12.ts > Fish > swim
You, 2 hours ago | 1 author (You)
1 export interface Flyable{
2     fly(): void;
3 }
You, 2 hours ago | 1 author (You)
4 export interface Swimmable {
5     swim(): void;
6 }
You, 2 hours ago | 1 author (You)
7 export class Bird implements Flyable {
8     fly(): void {
9         console.log("The bird is flying.");
10    }
11 }
You, 2 hours ago | 1 author (You)
12 export class Fish implements Swimmable {
13     swim(): void {
14         console.log("The fish is swimming.");
15     }
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

```
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.

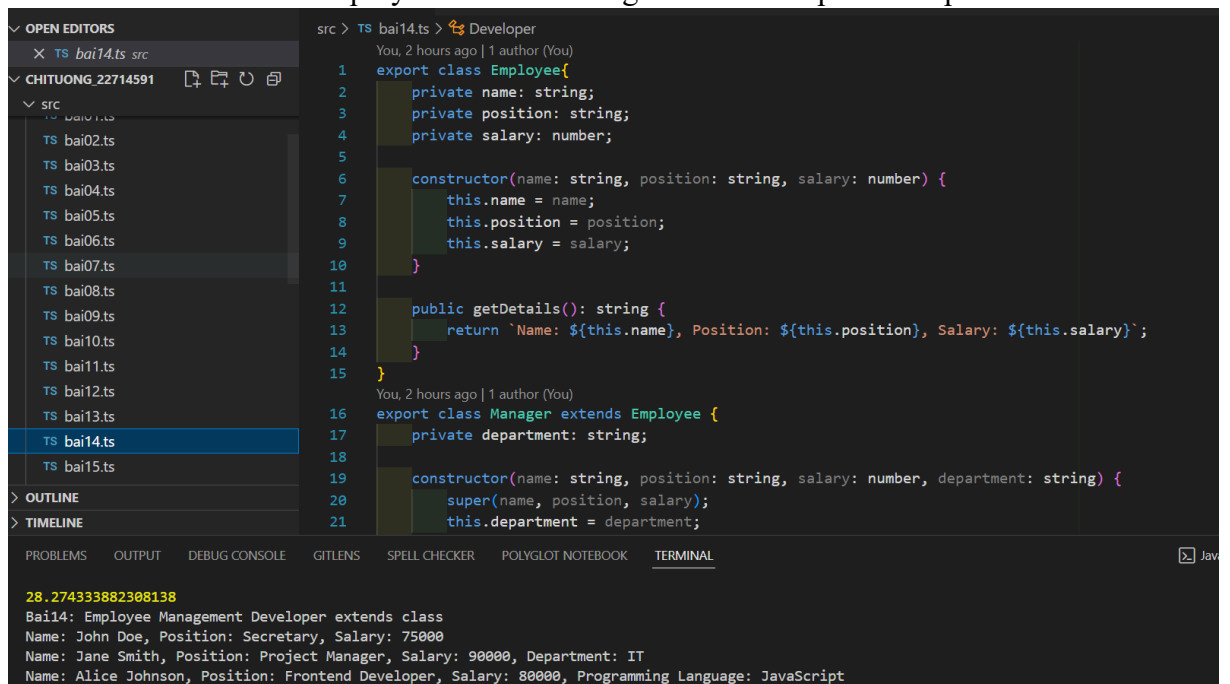


```
src > TS bai13.ts > Shape > getArea
You, 2 hours ago | 1 author (You)
1 export abstract class Shape{
2     abstract getArea(): number;
3 }
You, 2 hours ago | 1 author (You)
4 export class Square extends Shape {
5     private side: number;
6
7     constructor(side: number) {
8         super();
9         this.side = side;
10    }
11
12    getArea(): number {
13        console.log(`Area of square with side ${this.side}`);
14        return this.side * this.side;
15    }
16 }
You, 2 hours ago | 1 author (You)
17 export class Circle extends Shape {
18     private radius: number;
19
20    constructor(radius: number) {
21        super();
22        this.radius = radius;
23    }
24
25    getArea(): number {
26        console.log(`Area of circle with radius ${this.radius}`);
27        return Math.PI * this.radius * this.radius;
28    }
29 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.

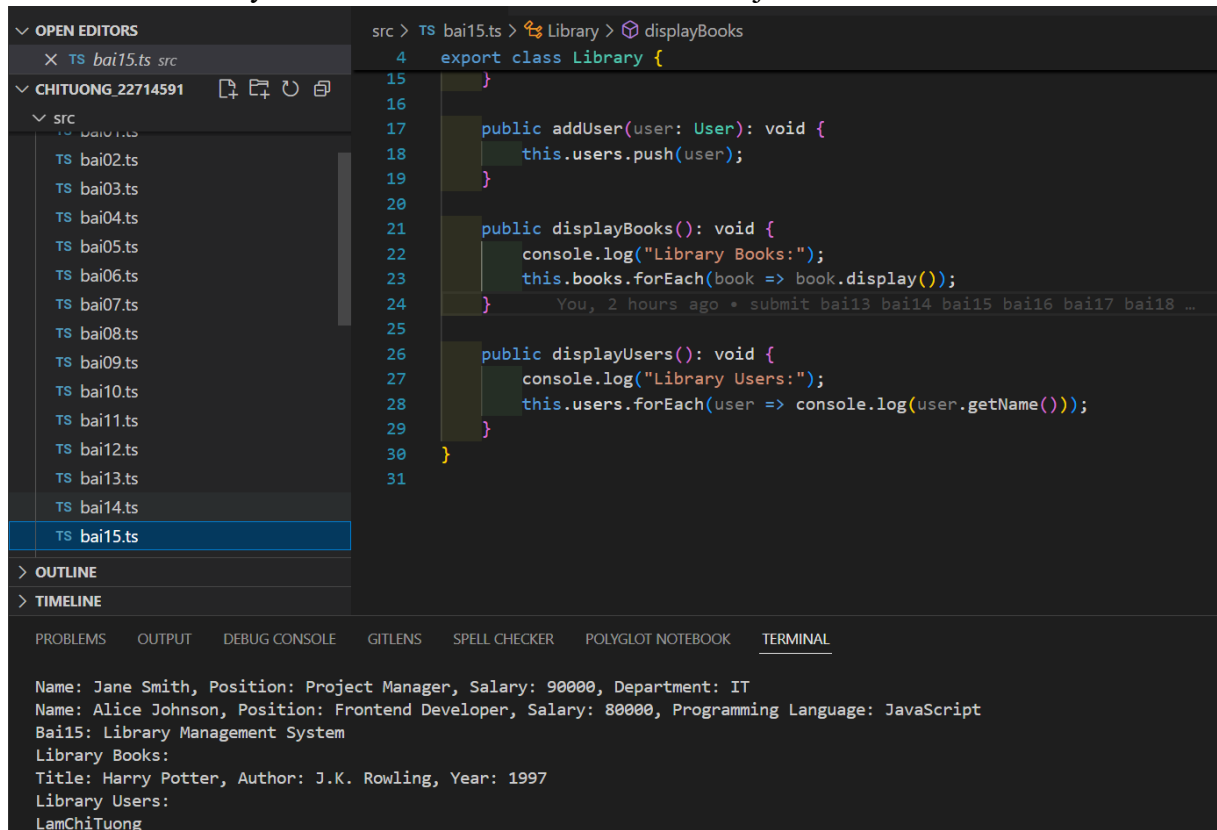


```
src > TS bai14.ts > Developer
You, 2 hours ago | 1 author (You)
1 export class Employee{
2     private name: string;
3     private position: string;
4     private salary: number;
5
6     constructor(name: string, position: string, salary: number) {
7         this.name = name;
8         this.position = position;
9         this.salary = salary;
10    }
11
12    public getDetails(): string {
13        return `Name: ${this.name}, Position: ${this.position}, Salary: ${this.salary}`;
14    }
15 }
You, 2 hours ago | 1 author (You)
16 export class Manager extends Employee {
17     private department: string;
18
19     constructor(name: string, position: string, salary: number, department: string) {
20         super(name, position, salary);
21         this.department = department;
22     }
23
24     getDetails(): string {
25         return `Name: ${this.name}, Position: ${this.position}, Salary: ${this.salary}, Department: ${this.department}`;
26     }
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.



The screenshot shows a VS Code editor with a TypeScript file named `bai15.ts` in the `src` directory. The file contains a `Library` class with three methods: `addUser`, `displayBooks`, and `displayUsers`. The `addUser` method adds a user to the `users` array. The `displayBooks` method logs the library books and calls `book.display()` for each book. The `displayUsers` method logs the library users and calls `console.log(user.getName())` for each user. The terminal output shows the results of running the code, including the names and positions of Jane Smith and Alice Johnson, the title and author of the Harry Potter book, and the name of the user LamChiTuong.

```
src > TS bai15.ts > Library > displayBooks
4  export class Library {
15
16
17      public addUser(user: User): void {
18          this.users.push(user);
19      }
20
21      public displayBooks(): void {
22          console.log("Library Books:");
23          this.books.forEach(book => book.display());
24      }
25
26      public displayUsers(): void {
27          console.log("Library Users:");
28          this.users.forEach(user => console.log(user.getName()));
29      }
30  }
31
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.

The screenshot shows the VS Code editor interface. On the left, the 'EXPLORER' sidebar displays a file tree with a folder named 'src' containing various TypeScript files. The file 'bai16.ts' is selected and highlighted. The main editor area shows the code for 'Box.ts'. The code defines a generic class 'Box' with a type parameter 'T'. It includes a private property 'value' of type 'T', a constructor that takes a 'T' value and assigns it to 'this.value', a 'getValue()' method that returns 'this.value', and a 'setValue(newValue: T): void' method that assigns 'newValue' to 'this.value'. The bottom status bar indicates the current file is 'Bai16: Generic Class'.

```
src > TS bai16.ts > Box
You, 2 hours ago | 1 author (You)
1  export class Box<T> {
2      private value: T;
3
4      constructor(value: T) {
5          this.value = value;
6      }
7
8      public getValue(): T {
9          return this.value;
10     }
11
12     public setValue(newValue: T): void {
13         this.value = newValue;
14     }
15 }
```

LamChiTuong
Bai16: Generic Class
123
Hello

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

The screenshot shows the VS Code editor interface. On the left, the 'EXPLORER' sidebar displays a file tree with a folder named 'src' containing various TypeScript files. The file 'bai17.ts' is selected and highlighted. The main editor area shows the code for 'Logger.ts'. The code defines a 'Logger' class with a static 'instance' property. It includes a static 'getInstance()' method that checks if 'instance' is null; if so, it creates a new 'Logger' object and assigns it to 'instance', then returns it. It also includes a 'log(message: string): void' method that uses 'console.log()' to output a message along with a timestamp. The bottom status bar indicates the current file is 'Bai17: Logger'.

```
src > TS bai17.ts > Logger > log
1  export class Logger{
2
3
4
5
6      public static getInstance(): Logger {
7          if (!Logger.instance) {
8              Logger.instance = new Logger();
9          }
10         return Logger.instance;
11     }
12
13     public log(message: string): void {
14         console.log(`[${new Date().toISOString()}] ${message}`);
15     }
16 }
```

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().

The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure with a 'src' folder containing files 'bai08.ts' through 'bai22.ts'. 'bai18.ts' is selected. The main editor area shows the content of 'bai18.ts', which defines a static class 'MathUtil' with four methods: 'add', 'subtract', 'multiply', and 'divide'. The 'divide' method includes a check for division by zero. The bottom panel shows the 'TERMINAL' tab with the following output:

```
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.

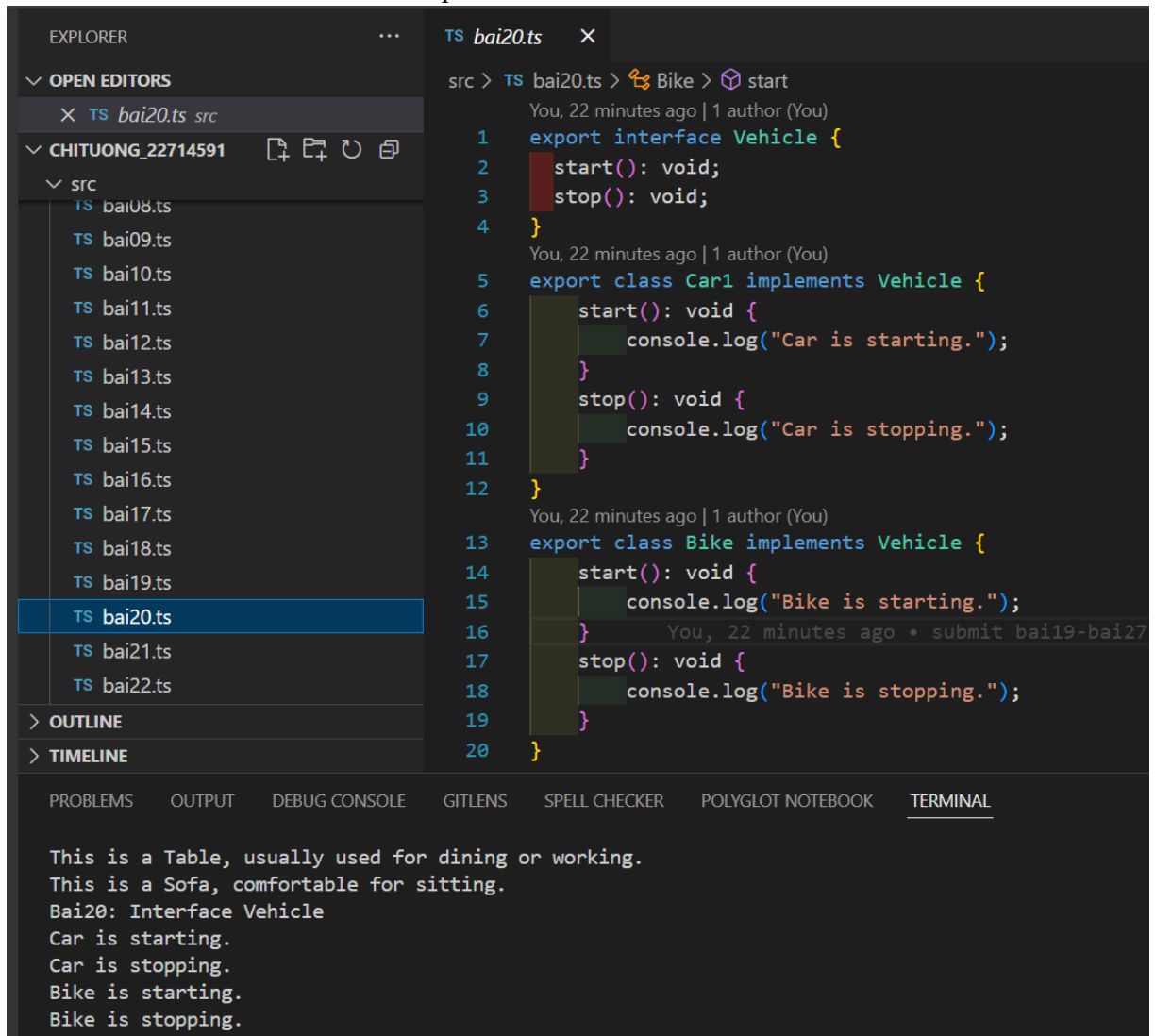
The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'src' with files 'bai08.ts' through 'bai22.ts'. The code editor shows the following TypeScript code:

```
src > TS bai19.ts > Chair
You, 22 minutes ago | 1 author (You)
1 export class Furniture {
2     describe(): void {
3         console.log("This is a piece of furniture.");
4     }
5 }
6
7 // Lớp con Chair ghi đè phương thức
You, 22 minutes ago | 1 author (You)
8 export class Chair extends Furniture {
9     describe(): void {
10        console.log("This is a Chair with 4 legs.");
11    }
12 }
13
14 // Lớp con Table ghi đè phương thức
You, 22 minutes ago | 1 author (You)
15 > export class Table extends Furniture { ...
19 }
20
21 // Lớp con Sofa ghi đè phương thức
You, 22 minutes ago | 1 author (You)
22 export class Sofa extends Furniture {
23     describe(): void {
```

The terminal at the bottom shows the output of the code:

```
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.



The image shows a VS Code editor with a TypeScript file named `bai20.ts` open. The file defines a `Vehicle` interface and two classes, `Car1` and `Bike`, that implement it. The `Vehicle` interface has two methods: `start(): void` and `stop(): void`. `Car1` implements these methods by logging "Car is starting." and "Car is stopping." respectively. `Bike` implements them by logging "Bike is starting." and "Bike is stopping." respectively. The terminal at the bottom shows the output of running the code, including the interface definition and the logs from both classes.

```
src > TS bai20.ts > Bike > start
You, 22 minutes ago | 1 author (You)
1 export interface Vehicle {
2   start(): void;
3   stop(): void;
4 }
You, 22 minutes ago | 1 author (You)
5 export class Car1 implements Vehicle {
6   start(): void {
7     console.log("Car is starting.");
8   }
9   stop(): void {
10    console.log("Car is stopping.");
11  }
12 }
You, 22 minutes ago | 1 author (You)
13 export class Bike implements Vehicle {
14   start(): void {
15     console.log("Bike is starting.");
16   }
17   stop(): void {
18     console.log("Bike is stopping.");
19   }
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

```
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().

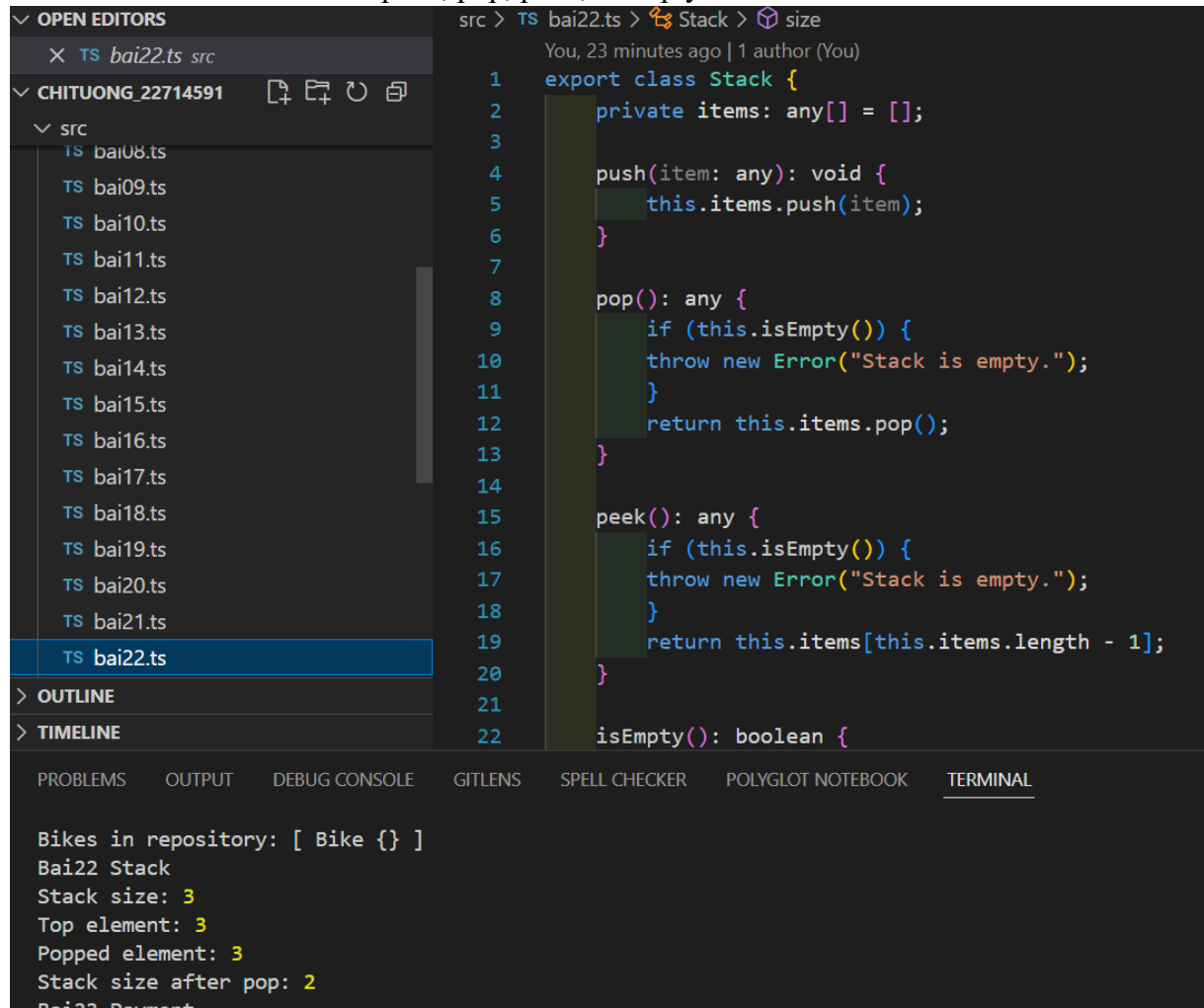
The screenshot shows the Visual Studio Code editor interface. On the left, the 'EXPLORER' sidebar displays a file tree with a folder named 'src' containing multiple TypeScript files (bai08.ts to bai22.ts). The file 'bai21.ts' is selected and highlighted. The main editor area shows the code for 'Repository.ts'. The code defines a generic class 'Repository' with a private array 'items' of type 'T[]'. It includes two methods: 'add(item: T): void' which pushes the item to the 'items' array, and 'getAll(): T[]' which returns the 'items' array. The bottom of the screen features the 'TERMINAL' panel, which displays the output of the program. The output shows a table, a sofa, and the execution of a generic repository, listing cars and bikes in the repository.

```
src > TS bai21.ts > Repository > getAll
You, 23 minutes ago | 1 author (You)
1 export class Repository<T> {
2     private items: T[] = [];
3
4     add(item: T): void {
5         this.items.push(item);
6     }
7
8     getAll(): T[] {
9         return this.items;
10    }
11 }
12
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.



```
src > TS bai22.ts > Stack > size
You, 23 minutes ago | 1 author (You)
1  export class Stack {
2      private items: any[] = [];
3
4      push(item: any): void {
5          this.items.push(item);
6      }
7
8      pop(): any {
9          if (this.isEmpty()) {
10             throw new Error("Stack is empty.");
11          }
12          return this.items.pop();
13      }
14
15      peek(): any {
16          if (this.isEmpty()) {
17             throw new Error("Stack is empty.");
18          }
19          return this.items[this.items.length - 1];
20      }
21
22      isEmpty(): boolean {
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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

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

CardPayment.

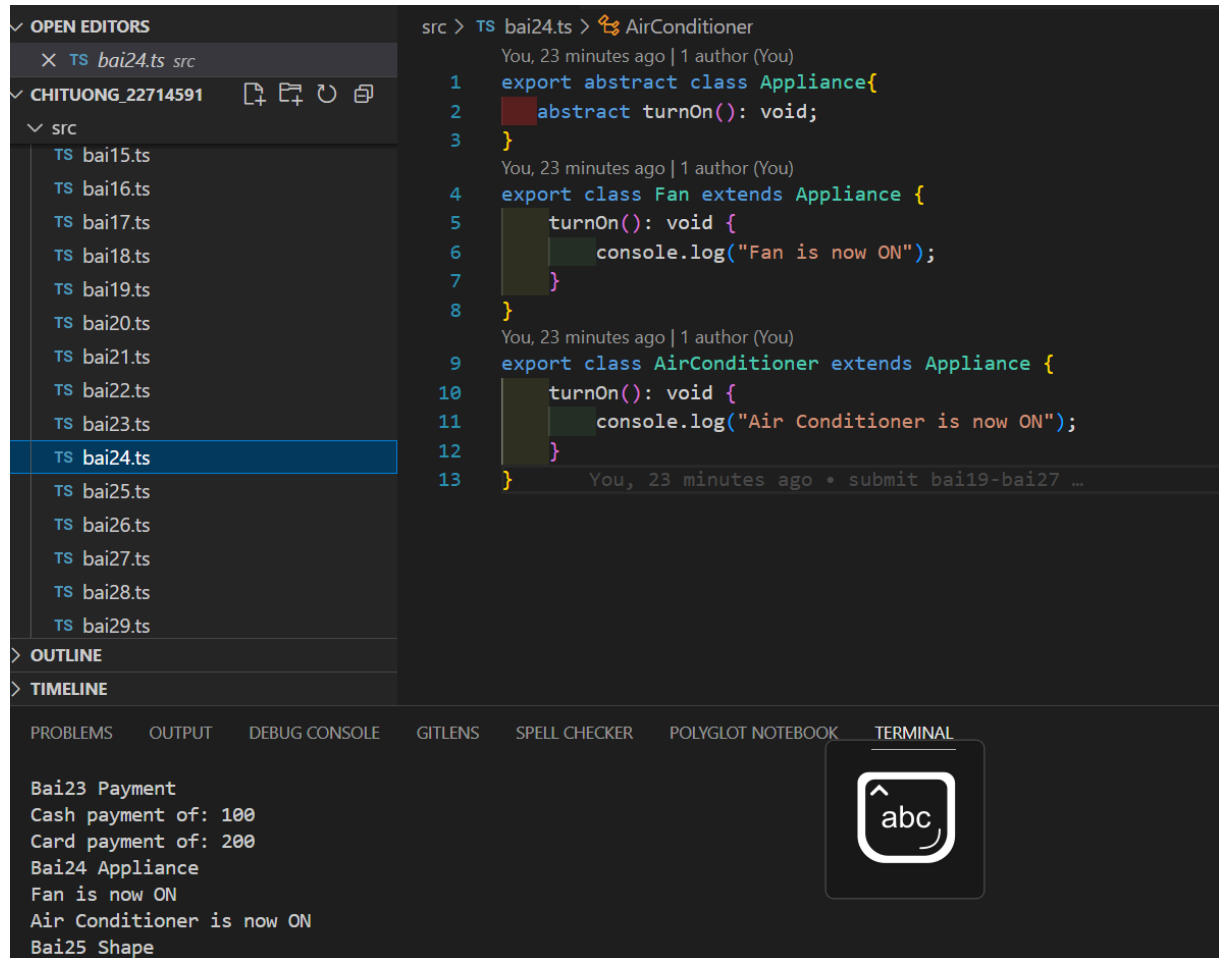
The image shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'CHITUONG_22714591' with a 'src' folder containing files 'bai15.ts' through 'bai29.ts'. The file 'bai23.ts' is selected. The code editor shows the following TypeScript code:

```
src > TS bai23.ts > CardPayment
You, 23 minutes ago | 1 author (You)
1 export class Payment{
2     pay(amount:number):void{
3         console.log(`Paid: ${amount}`);
4     }
5 }
You, 23 minutes ago | 1 author (You)
6 export class CashPayment extends Payment {
7     pay(amount: number): void {
8         console.log(`Cash payment of: ${amount}`);
9     }
10 }
You, 23 minutes ago | 1 author (You)
11 export class CardPayment extends Payment {
12     pay(amount: number): void {
13         console.log(`Card payment of: ${amount}`);
14     }
15 }
```

The terminal at the bottom shows the output of the code:

```
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.



The screenshot shows the VS Code editor interface. On the left, the 'OPEN EDITORS' panel shows a file named 'bai24.ts' in the 'src' directory. The 'CHITUONG_22714591' workspace is also visible. The main editor area displays the following TypeScript code:

```
src > TS bai24.ts > AirConditioner
You, 23 minutes ago | 1 author (You)
1 export abstract class Appliance{
2   abstract turnOn(): void;
3 }
You, 23 minutes ago | 1 author (You)
4 export class Fan extends Appliance {
5   turnOn(): void {
6     console.log("Fan is now ON");
7   }
8 }
You, 23 minutes ago | 1 author (You)
9 export class AirConditioner extends Appliance {
10   turnOn(): void {
11     console.log("Air Conditioner is now ON");
12   }
13 }
```

At the bottom, the 'TERMINAL' panel shows the output of the code execution:

```
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().

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a file tree with a folder 'src' containing files 'bai15.ts' through 'bai29.ts'. 'bai25.ts' is selected. The main editor displays the code for 'bai25.ts':

```
src > TS bai25.ts > Shape1 > describe
You, 24 minutes ago | 1 author (You)
1 export class Shape1{
2     static describe(){
3         console.log("This is a shape.");
4     }
5 }
```

Below the editor, the GitLens sidebar is visible. At the bottom, the TERMINAL panel shows the following output:

```
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.

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER (Left):** Shows the file explorer with a list of files under the `src` directory. The file `TS bai26.ts` is selected.
- EDITOR (Center):** Displays the code for `TS bai26.ts`. The code defines an `Order` class with a private `products` array and two methods: `addProduct` and `calculateTotal`.
- TERMINAL (Bottom):** Shows the output of the program, including the creation of an `Order` object and the calculation of the total price.

Code in `TS bai26.ts`:

```
src > TS bai26.ts > Order
You, 24 minutes ago | 1 author (You)
1 import { Product } from "../bai08.js";
2
3
4 export class Order{
5     private products: Product[] = [];
6
7     addProduct(product: Product): void {
8         this.products.push(product);
9     }
10    calculateTotal(): number {
11        return this.products.reduce((total, product) => total + product.price, 0);
12    }
13 }
```

Terminal Output:

```
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.

The screenshot shows the VS Code interface with the Explorer, Open Editors, and Source Control panels on the left. The Explorer panel shows a file tree with a 'src' folder containing files from 'bai15.ts' to 'bai29.ts'. The Open Editors panel shows 'bai27.ts' as the active file. The Source Control panel shows the file is part of a commit by 'CHITUONG_22714591'. The main editor area shows the code for 'bai27.ts', which imports the 'Person' class from './bai01.js' and defines a 'Teacher' class that extends 'Person'. The 'Teacher' class has a private 'subject' attribute, a constructor that calls 'super' and sets 'this.subject', and a 'teach' method that logs a message. The terminal panel at the bottom shows the output of the program, including the message 'John Doe is teaching Mathematics.'.

```
src > TS bai27.ts > Teacher
You, 24 minutes ago | 1 author (You)
1 import { Person } from "./bai01.js";
2
3 You, 24 minutes ago | 1 author (You)
4 export class Teacher extends Person{
5     private subject: string;
6
7     constructor(name: string, age: number, subject: string) {
8         super(name, age);
9         this.subject = subject;
10    }
11
12    public teach(): void {
13        console.log(`${this.getName()} is teaching ${this.subject}.`);
14    }
15 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

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.

The image shows a VS Code editor with a TypeScript project. The Explorer sidebar on the left shows a file tree with files named bai15.ts through bai29.ts. The file bai28.ts is selected and open in the editor. The code in bai28.ts defines three classes: Animal2, Dog2, and Cat2. Animal2 has a protected method makeSound() that logs "Animal sound". Dog2 extends Animal2 and has a public method bark() that calls this.makeSound() and logs "Woof! Woof!". Cat2 extends Animal2 and has a public method meow() that calls this.makeSound() and logs "Meow! Meow!". The terminal at the bottom shows the output of the program, which includes the text "Hello, my name is John Doe and I am 30 years old.", "John Doe is teaching Mathematics.", and the outputs of the makeSound() method for each class: "Animal sound", "Woof! Woof!", and "Meow! Meow!".

```
src > TS bai28.ts > Cat2
You, 15 minutes ago | 1 author (You)
1 export class Animal2{
2     protected makeSound(): void {
3         console.log("Animal sound");
4     }
5 }
You, 15 minutes ago | 1 author (You)
6 export class Dog2 extends Animal2 {
7     public bark(): void {
8         this.makeSound();
9         console.log("Woof! Woof!");
10    }
11 }
You, 15 minutes ago | 1 author (You)
12 export class Cat2 extends Animal2 {
13     public meow(): void {
14         this.makeSound();
15         console.log("Meow! Meow!");
16     }
17 }
```

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS SPELL CHECKER POLYGLOT NOTEBOOK TERMINAL

```
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.

The image shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom.

File Explorer: Shows a project structure with a folder named `src` containing several TypeScript files. The file `bai29.ts` is selected.

Code Editor: Displays the content of `bai29.ts`. The code defines an interface `Movable` with a `move(): void` method. It then implements this interface in two classes: `Car3` and `Robot`. The `Car3` class logs "Car is moving" and the `Robot` class logs "Robot is moving".

```
src > TS bai29.ts > Robot
You, 16 minutes ago | 1 author (You)
1 export interface Movable {
2     move(): void;
3 }
You, 16 minutes ago | 1 author (You)
4 export class Car3 implements Movable {
5     move(): void {
6         console.log("Car is moving");
7     }
8 }
You, 16 minutes ago | 1 author (You)
9 export class Robot implements Movable {
10    move(): void {
11        console.log("Robot is moving");
12    }
13 }
```

Terminal: Shows the output of the program execution. It includes a greeting message, a list of teachers, and the output of the `move()` method for both `Car3` and `Robot`.

```
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.

The screenshot shows the Visual Studio Code editor with a project named 'CHITUONG_22714591'. The Explorer sidebar on the left shows a 'src' folder containing several TypeScript files, with 'TS bai30.ts' selected. The main editor displays the code for 'TS bai30.ts', which defines a 'School' class with two methods: 'addStudent' and 'displayInfo'. The 'addStudent' method takes a 'Student' object and pushes it to the 'students' array. The 'displayInfo' method logs 'Teachers' and 'Students' information to the console. The bottom panel shows the 'TERMINAL' output, which includes the execution of previous tasks (Bai29) and the output of the current task (Bai30), showing the 'School' class and its 'displayInfo' method output.

```
src > TS bai30.ts > ...
4   export class School{
12   addStudent(student: Student): void {
13       this.students.push(student);
14   }
15
16   displayInfo(): void {
17       console.log("Teachers:");
18       this.teachers.forEach(teacher => teacher.display());
19   }
20       console.log("Students:");
21       this.students.forEach(student => student.display());
22   }
23 }
24
```

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>