

[CGA-JAVA-WFDA] Web Front-end Development Angular 2.1

[Dashboard](#) / [My courses](#) / [CGA-JAVA-WFDA-2.1](#) / [2. Typescript](#) / [\[Thực hành\] Xây dựng các đối tượng hình học](#)

[Thực hành] Xây dựng các đối tượng hình học

Mục tiêu

Luyện tập kiến thức về khai báo đối tượng, interface và các tính chất.

Mô tả

Xây dựng lớp Shape với 2 thuộc tính x và y là toạ độ của các đối tượng Shape.

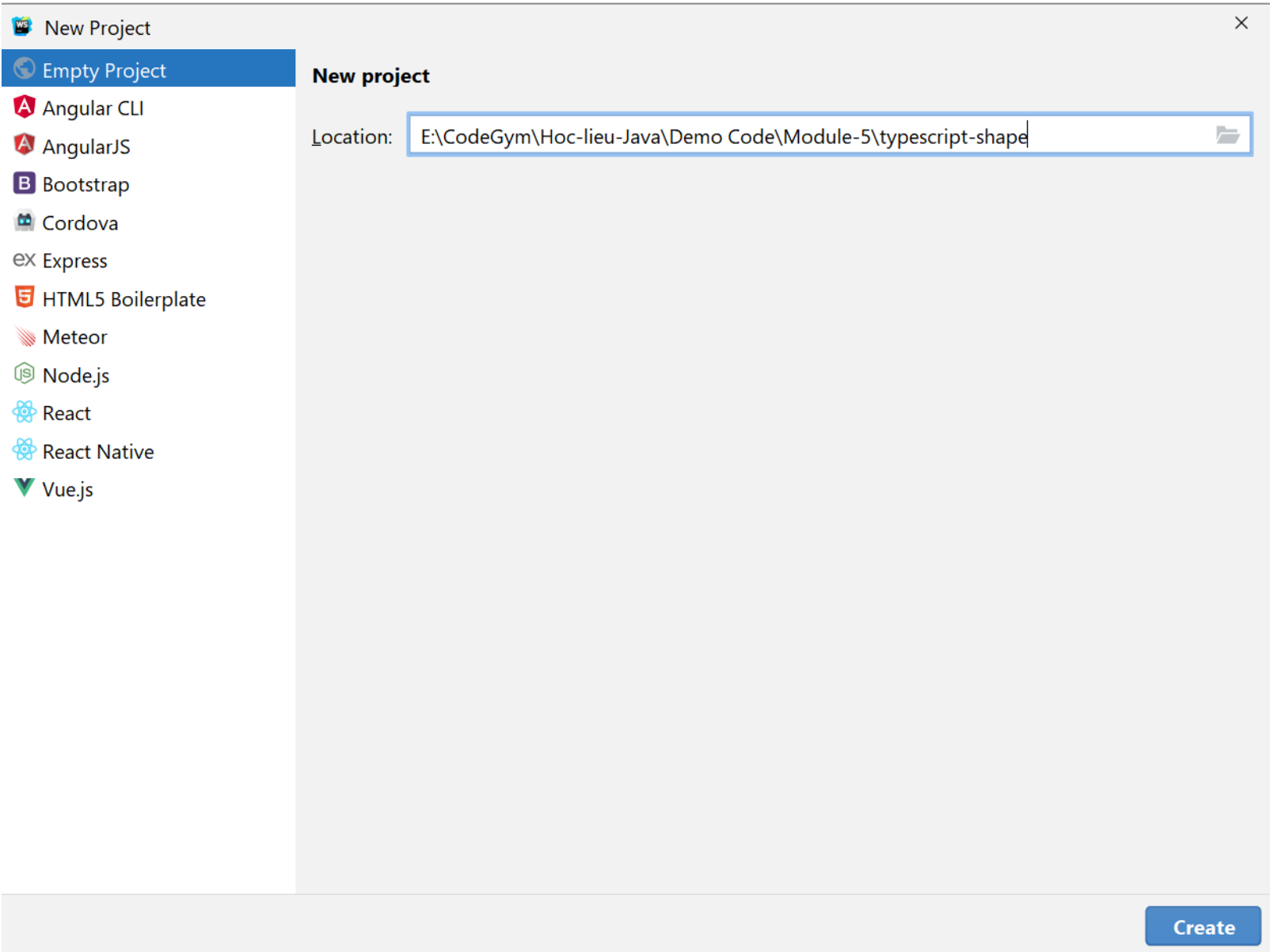
Xây dựng Interface IArea để thực hiện việc tính diện tích các hình.

Xây dựng lớp Rectangle kế thừa lớp Shape và implement interface IArea và cài đặt phương thức area.

Thực hiện việc khai báo đối tượng rectangle và tính diện tích của hình chữ nhật đó

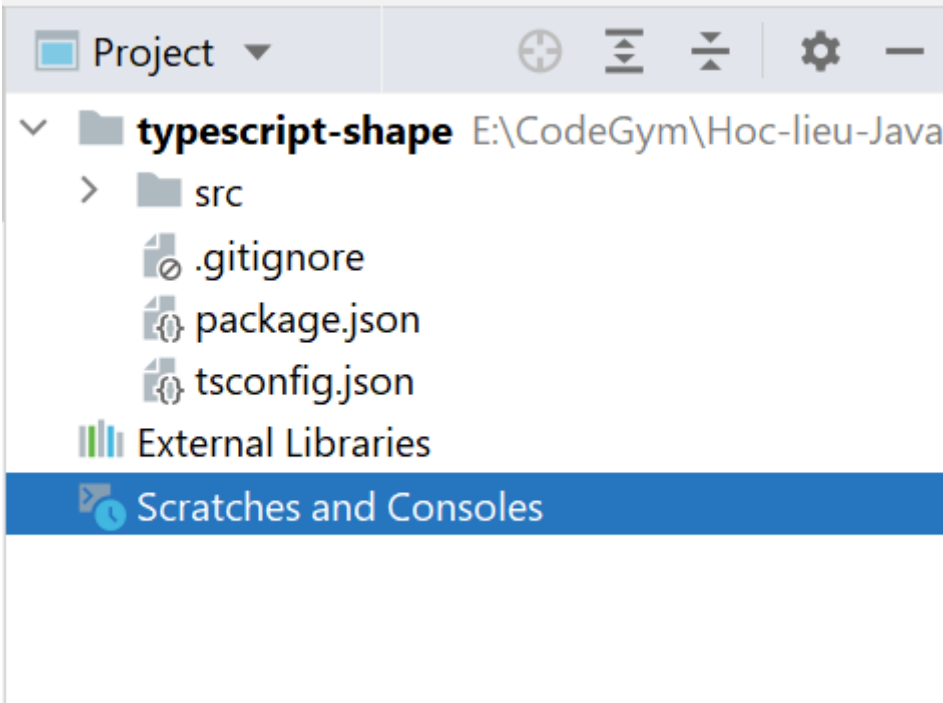
Hướng dẫn

Bước 1: Tạo một Project mới trên WebStorm bằng cách vào File chọn New -> Project -> Empty Project.

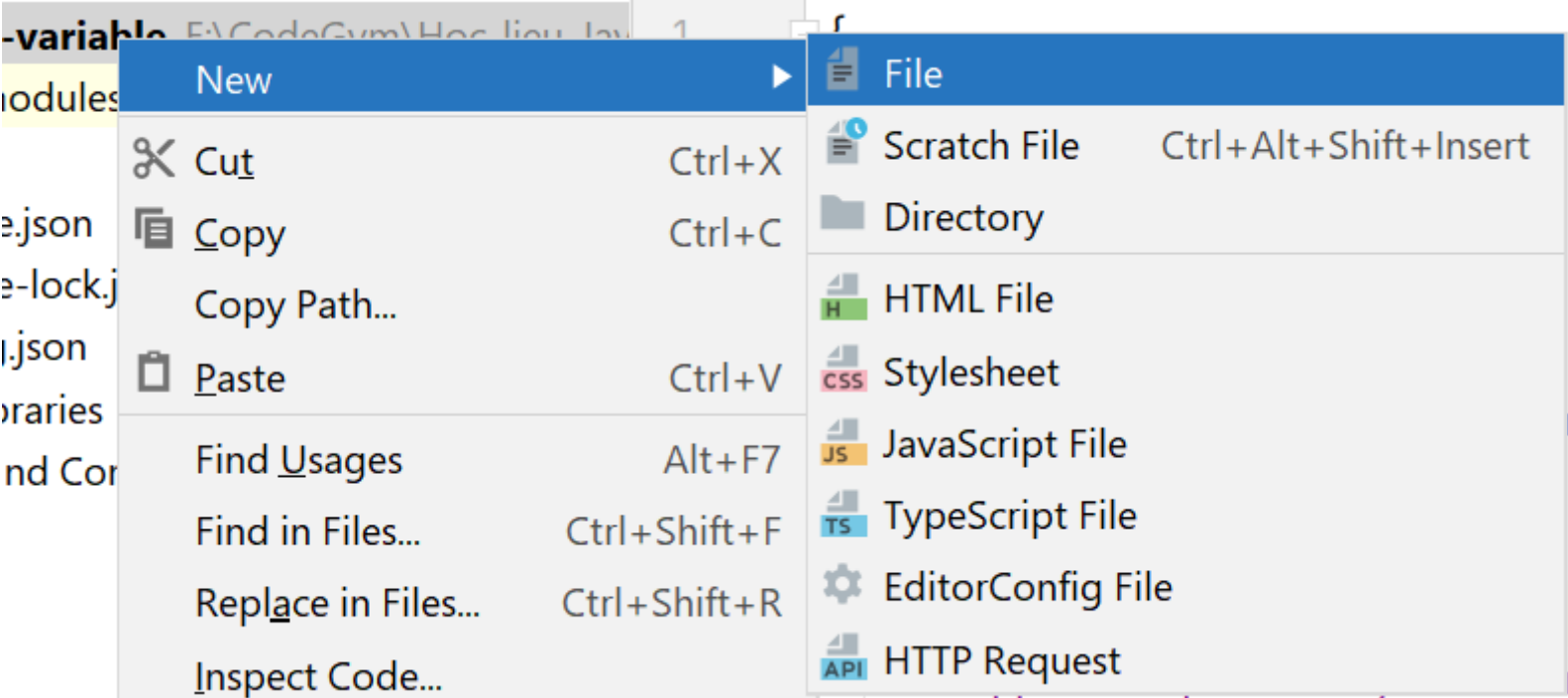


Sau đó đặt tên project là typescript-is-prime và chọn Create.

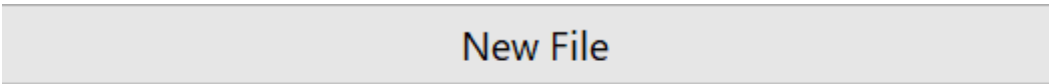
Bước 2: Cấu trúc của dự án sẽ có dạng như sau:



Bước 3: Tạo một file có tên là **tsconfig.json** bằng cách chuột phải chọn New->File



Đặt tên file là tsconfig.json



tsconfig.json

Cấu hình file tsconfig.json:

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "declaration": true,
    "sourceMap": true,
    "experimentalDecorators": true,
    "outDir": "dist",
    "lib": ["es6", "dom"],
    "noImplicitAny": true,
    "suppressImplicitAnyIndexErrors": true,
    "removeComments": true
  },
  "exclude": ["node_modules", "dist"]
}
```

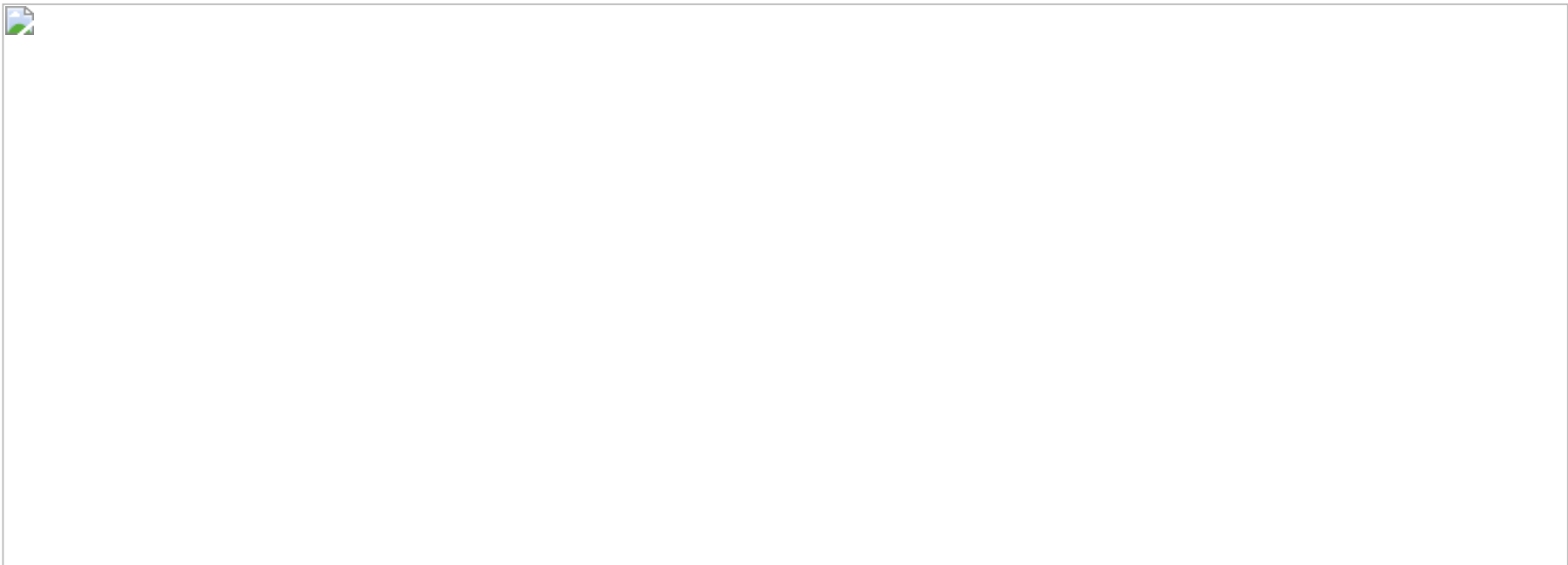
Bước 4: Tạo một file có tên là **package.json** và chúng ta sẽ cấu hình file đó dưới dạng json như dưới đây:

```
{
  "name": "typescript-is-shape",
  "version": "1.0.0",
  "dependencies": {
    "typescript": "^4.1.5"
  }
}
```

Trong đó:

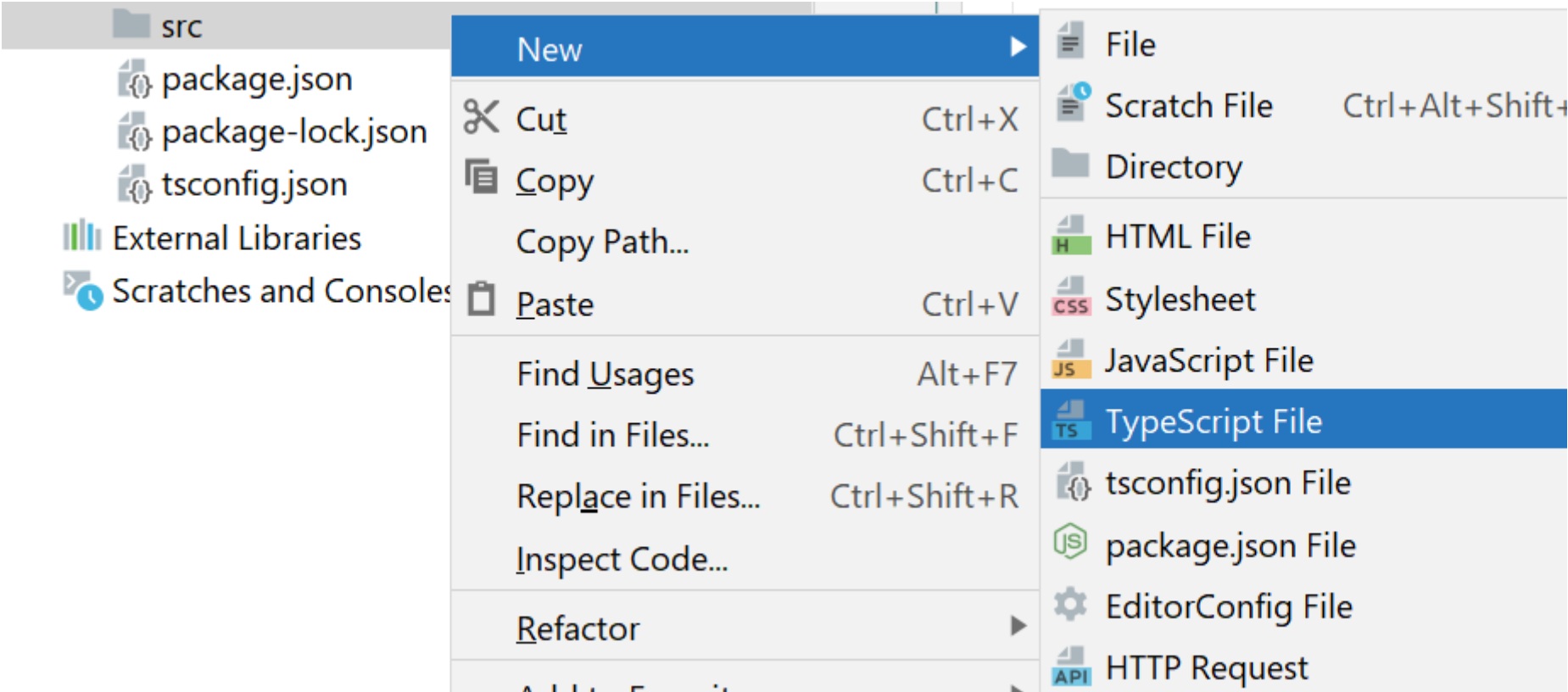
- name: tên của dự án
- version: phiên bản của dự án
- dependencies: khai báo các thư viện cần sử dụng trong dự án vào trong dependencies (ở trên là thư viện typescript version 4.1.5)

Bước 5: Sau khi tạo xong các file package.json và tsconfig.json, chúng ta mở cửa sổ terminal lên và chạy câu lệnh **npm i** để tải các thư viện typescript đã khai báo trong package.json ở trên về dự án.

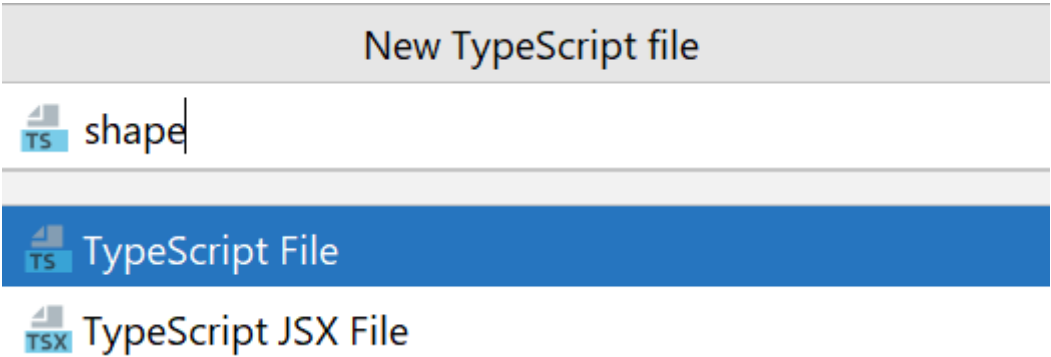


Sau khi chạy xong câu lệnh trên, dự án của chúng ta sẽ có thêm thư mục là node_modules và một file package-lock.json.

Bước 6: Chọn thư mục src sau đó chuột phải chọn New->TypeScriptFile



Đặt tên file là shape.ts



Khai báo lớp Shape như sau:

```
export class Shape {
  private _x: number;

  private _y: number;

  constructor(x: number, y: number) {
    this._x = x;
    this._y = y;
  }

  get y() {
    return this._y;
  }

  set y(y: number) {
    this._y = y;
  }

  get x() {
    return this._x;
  }

  set x(x: number) {
    this._x = x;
  }

  toString(): string {
    return `(x: ${this._x}, y: ${this._y})`;
  }
}
```

Bước 7: Tạo file IArea.ts và khai báo interface IArea với phương thức area() như dưới đây:

```
interface IArea {
  area(): number;
}
```

Bước 8: Tạo file rectangle.ts và tạo class Rectangle extends lớp Shape và implement interface IArea, cài đặt phương thức area() như sau:

```
import {Shape} from "./shape";

export class Rectangle extends Shape implements IArea{
    constructor(x: number, y: number, private _width: number, private _height: number) {
        super(x, y);
    }

    area(): number {
        return this._width*this._height;
    }

    get width() {
        return this._width;
    }

    set width(width: number) {
        this._width = width;
    }

    get height() {
        return this._height;
    }

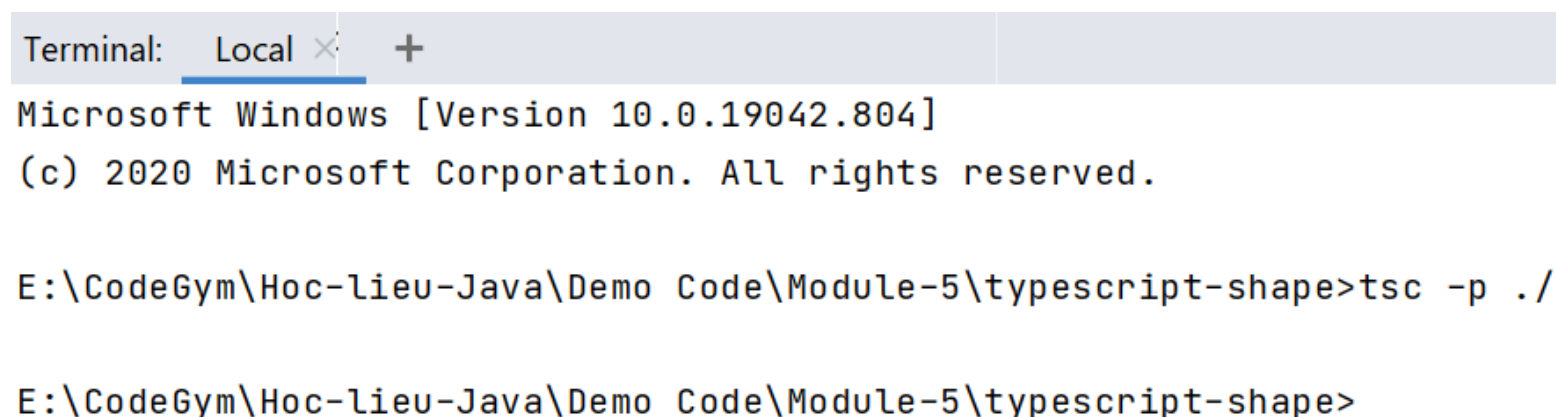
    set height(height: number) {
        this._height = height;
    }
}
```

Bước 9: Trong file main.ts viết đoạn code như sau:

```
import {Rectangle} from "./rectangle";

const rectangle = new Rectangle(5, 5, 10, 20);
console.log(rectangle.toString());
console.log(rectangle.area());
```

Bước 10: Tiến hành chạy lệnh **tsc -p ./** trên terminal để chương trình build cả thư mục src sang file javascript

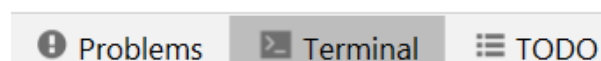


The screenshot shows a Windows terminal window with the following content:

```
Terminal: Local +
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

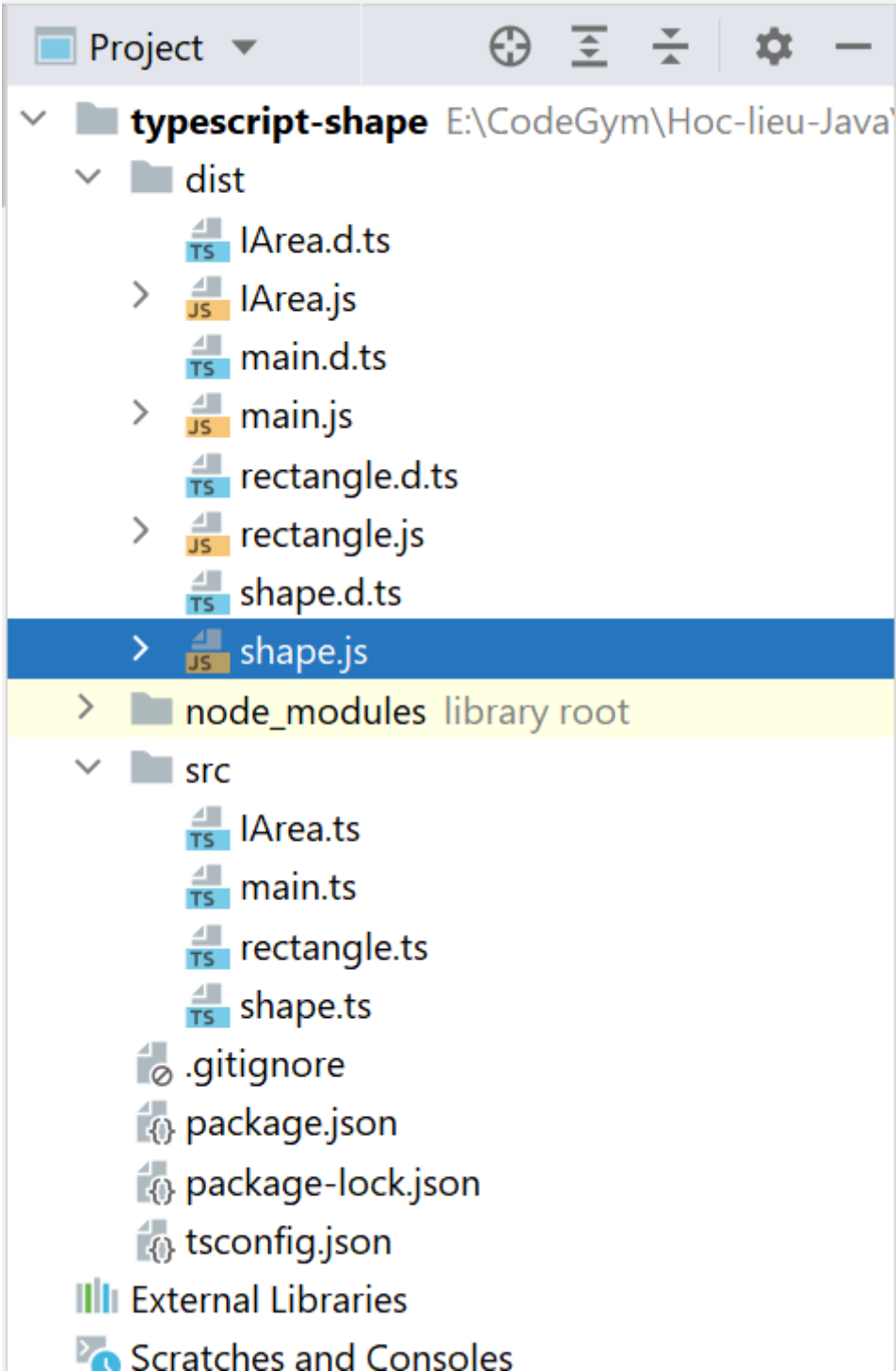
E:\CodeGym\Hoc-lieu-Java\Demo Code\Module-5\typescript-shape>tsc -p ./

E:\CodeGym\Hoc-lieu-Java\Demo Code\Module-5\typescript-shape>
```

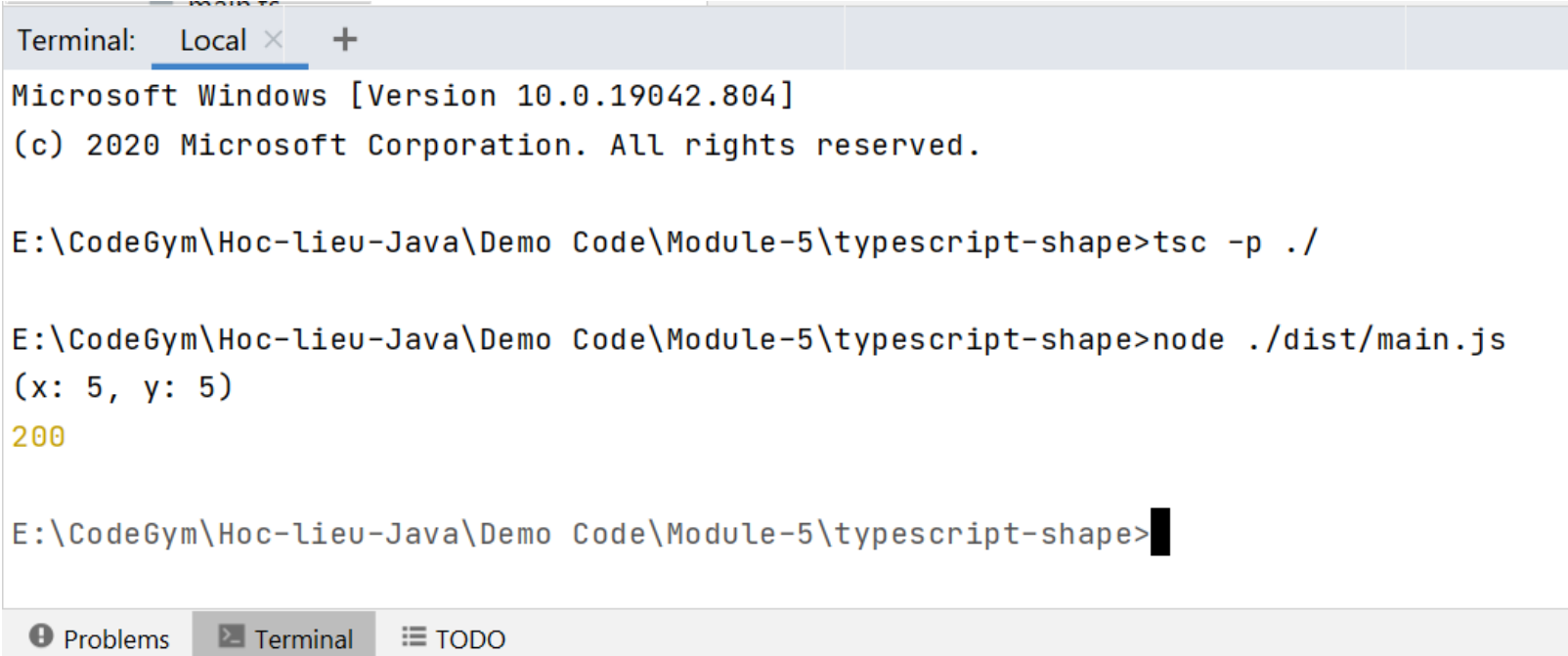


The screenshot shows the bottom status bar of Visual Studio Code with three tabs: 'Problems' (with an error icon), 'Terminal' (with a terminal icon and highlighted), and 'TODO' (with a list icon).

Sau khi chạy xong câu lệnh trên ứng dụng sẽ tạo thêm một thư mục **dist** chứa các file như sau:



Mở cửa sổ terminal và chạy lệnh **node ./dist/main.js** để hiển thị kết quả



Checkout sang nhánh develop để xem mã nguồn tham khảo: <https://github.com/codegym-vn/wfda-typescript-shape>

Hướng dẫn nộp bài:

up bài tập lên github

paste link github vào phần nộp bài

Submission status

Submission status	No attempt
Grading status	Not graded

Last modified

-

Submission
comments

▶ [Comments \(0\)](#)

Add submission

You have not made a submission yet.

Previous Activity

Jump to...

Next Activity

CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

Follow Us



CodeGym@2018. All rights reserved.

You are logged in as Dương Văn Thanh Sơn (Log out)