

## MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng module.
- ✓ Sử dụng namespace.
- ✓ Cài đặt và sử dụng webpack.

## NỘI DUNG

### Bài 1: Module

TS StringValidator.ts ✕

modules > src > TS StringValidator.ts > StringValidator

```
1 export interface StringValidator {  
2   |   isAcceptable(s: string): boolean;  
3   }  
}
```

TS StaticZipCodeValidator.ts ✕

modules > src > TS StaticZipCodeValidator.ts > ...

```
1 const numberRegexp = /^[0-9]+$/;  
2  
3 export default function (s: string) {  
4   |   return s.length === 5 && numberRegexp.test(s);  
5   }  
}
```

TS ZipCodeValidator.ts ✕

modules > src > TS ZipCodeValidator.ts > ...

```
1 import { StringValidator } from "./StringValidator";  
2  
3 export const numberRegexp = /^[0-9]+$/;  
4  
5 export class ZipCodeValidator implements StringValidator {  
6   |   isAcceptable(s: string) {  
7     |   return s.length === 5 && numberRegexp.test(s);  
8     }  
9   };  
10  
11 export { ZipCodeValidator as mainValidator };
```

TS ParseIntBasedZipCodeValidator.ts ×

modules > src > TS ParseIntBasedZipCodeValidator.ts > ParseIntBasedZipCodeValidator

```
1 export class ParseIntBasedZipCodeValidator {
2     isAcceptable(s: string) {
3         return s.length === 5 && parseInt(s).toString() === s;
4     }
5 }
6
7 // Export original validator but rename it
8 export { ZipCodeValidator as RegExpBasedZipCodeValidator } from "../ZipCodeValidator";
```

TS AllValidators.ts ×

modules > src > TS AllValidators.ts

```
1 export * from "../StringValidator"; // exports 'StringValidator' interface
2 export * from "../ZipCodeValidator"; // exports 'ZipCodeValidator' and const 'numberRegex' class
3 export * from "../ParseIntBasedZipCodeValidator"; // exports the 'ParseIntBasedZipCodeValidator' class
4 // and re-exports 'RegExpBasedZipCodeValidator' as alias
5 // of the 'ZipCodeValidator' class from 'ZipCodeValidator.ts'
6 // module.
```

TS Test.ts ×

modules > src > TS Test.ts > ...

```
1 import validate from "../StaticZipCodeValidator";
2
3 let strings = ["Hello", "98052", "101"];
4
5 // Use function validate
6 strings.forEach((s) => {
7     console.log(`${s} ${validate(s) ? "matches" : "does not match"}`);
8 });
```

## Bài 2: Namespace

TS studentCalc.ts ×

namespaces > src > TS studentCalc.ts > {} studentCalc

```
1 namespace studentCalc{
2     export function AnnualFeeCalc(feeAmount: number, term: number){
3         return feeAmount * term;
4     }
5 }
6
```

TS index.ts ×

namespaces > src > TS index.ts > ...

```
1 /// <reference path = "studentCalc.ts" />
2
3 let TotalFee = studentCalc.AnnualFeeCalc(1500, 4);
4
5 console.log("Output: " +TotalFee);
6
```

## Bài 3: Webpack

- Tạo project có cấu trúc sau:

```

✓ learn-webpack-01
  > dist
  > node_modules
  > src
  <> index.html
  {} package-lock.json
  {} package.json
  {} tsconfig.json
  📦 webpack.config.js
  
```

- Cài đặt webpack cho project.
- Cấu hình tsconfig.json, webpack.config chuyển src/index.ts → dist/bundle.js
- Code các trang ví dụ phía dưới (tìm lỗi sai và tự chỉnh sửa)
- src/index.ts

```

import { Person } from "../interfaces/person";
import { Student } from "../models/student";

function greeter (person: Person){
    return "Hello, " + person.firstName + " " + person.lastName;
}

let student: Student = new Student ("Thi","Hong","Nguyen");

document.body.innerHTML = greeter(student);
  
```

- src/models/person.ts

```

export interface Person {
    firstName: string;
    lastName: string;
}
  
```

- src/models/student.ts

```

import { Person } from '../interfaces/person';

export class Student implements Person {
    fullName: string;
    firstName: string;
    middleInitial: string;
    lastName: string;
    constructor(firstName: string, middleInitial: string, lastName: string){
        this.firstName = firstName;
        this.middleInitial = middleInitial;
        this.lastName = lastName;
        this.fullName = firstName + " " + middleInitial + " " + lastName;
    }
}
  
```

- Index.html

```
<!DOCTYPE html>
<head><title>Typescript and Webpack</title></head>
<body>
  <script src="dist/bundle.js"></script>
</body>
</html>
```

## Bài 4: LEARNING TYPESCRIPT

Tạo simple project có cấu trúc sau:

- Tên project: LEARNING\_TYPESCRIPT
- Project gồm: src/app.ts (biên dịch đến dist/app.js), index.html, style.css, tsconfig.json
- Cài đặt webpack cho project

### *app.ts*

- Chuyển và export các đối tượng , class ra tập tin .ts. Đặt tên là <tên đối tượng/tên class>.model.ts  
Ví dụ: *user.model.ts*
- Import, tham chiếu và sử dụng các đối tượng, class

### \*\*\* Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---