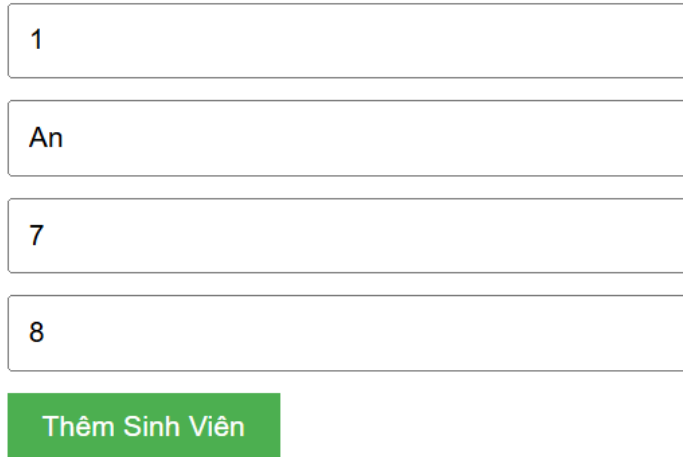


## LAB 4: HƯỚNG ĐỐI TƯỢNG

**Bài 1: Viết chương trình OOP quản lý sinh viên đơn giản: Nhập, xuất thông tin, tính điểm TB.**

### Quản Lý Sinh Viên



A form with four input fields stacked vertically. The first field contains the number '1', the second contains 'An', the third contains '7', and the fourth contains '8'. Below these fields is a green button with the text 'Thêm Sinh Viên'.

1
An
7
8

Thêm Sinh Viên

Mã sinh viên: 1  
Họ tên: An  
Điểm LT: 7  
Điểm TH: 8  
Điểm TB: 7.50

#### Yêu cầu:

Viết lớp Sinh viên như sau:

Attributes (private):

- Mã sinh viên là số nguyên.
- Họ tên: chuỗi ký tự.
- Điểm LT, điểm TH: float
- 

Constructor:

- Constructor mặc định (để khởi tạo đối tượng với các thông tin kiểu số là 0, kiểu chuỗi là chuỗi rỗng).

Methods:

- Các getter và setter cho mỗi thuộc tính.
- Tính điểm trung bình.
- Phương thức toString để diễn tả đối tượng ở dạng chuỗi.

Hướng dẫn:

File .html

```
<body>
  <h1>Quản Lý Sinh Viên</h1>
  <div class="student-form">
    <input type="number" id="studentId" placeholder="Mã sinh viên">
    <input type="text" id="studentName" placeholder="Họ tên">
    <input type="number" step="0.1" id="studentLT" placeholder="Điểm lý thuyết">
    <input type="number" step="0.1" id="studentTH" placeholder="Điểm thực hành">
    <button onclick="addStudent()">Thêm Sinh Viên</button>
  </div>

  <div class="student-info" id="studentInfo"></div>

  <script src="Bai1.js"></script>
</body>
```

File .css

```
body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

.student-form {
  margin-bottom: 20px;
}

.student-form input {
  display: block;
  margin-bottom: 10px;
  padding: 8px;
  width: 300px;
}

.student-form button {
  padding: 8px 16px;
  background-color: #4CAF50;
  color: white;
  border: none;
  cursor: pointer;
}
```

```
.student-form button:hover {
  background-color: #45a049;
}

.student-info {
  margin-top: 20px;
}

.student-info p {
  margin: 5px 0;
  font-size: 18px;
}
```

File .js

```

class Student {
    // Constructor mặc định
    constructor(id = 0, name = '', ltScore = 0, thScore = 0) {
        this._id = id;
        this._name = name;
        this._ltScore = ltScore;
        this._thScore = thScore;
    }

    // Các getter và setter cho mỗi thuộc tính
    get id() {
        return this._id;
    }

    set id(value) {
        this._id = value;
    }

    get name() {
        return this._name;
    }

    set name(value) {
        this._name = value;
    }

```

```

    get ltScore() {
        return this._ltScore;
    }

    set ltScore(value) {
        this._ltScore = value;
    }

    get thScore() {
        return this._thScore;
    }

    set thScore(value) {
        this._thScore = value;
    }

    // Tính điểm trung bình
    calculateAverage() {
        return (this._ltScore + this._thScore) / 2;
    }

    // Phương thức toString để diễn tả đối tượng ở dạng chuỗi
    toString() {
        return `Mã sinh viên: ${this._id}\nHọ tên: ${this._name}\nĐiểm LT: ${this._ltScore}\nĐiểm TH: ${this._thScore}\nĐiểm TB: ${this.calculateAverage().toFixed(2)}`;
    }
}

```

```

// Hàm thêm sinh viên và hiển thị thông tin sinh viên
function addStudent() {
    // Lấy giá trị từ các ô input
    const id = parseInt(document.getElementById('studentId').value);
    const name = document.getElementById('studentName').value;
    const ltScore = parseFloat(document.getElementById('studentLT').value);
    const thScore = parseFloat(document.getElementById('studentTH').value);

    // Tạo đối tượng sinh viên mới
    const student = new Student(id, name, ltScore, thScore);

    // Hiển thị thông tin sinh viên
    const studentInfo = document.getElementById('studentInfo');
    studentInfo.innerHTML = `
        <p>${student.toString().replace(/\n/g, '<br>')}</p>
    `;
}

```

Bài 2: Quản lý giỏ hàng

Tên Sản Phẩm: Laptop XYZ

Giá: 1000 USD

Thêm Vào Giỏ Hàng

Giỏ Hàng

1. Laptop XYZ - 1000 USD

2. Laptop XYZ - 1000 USD

3. Laptop XYZ - 1000 USD

4. Laptop XYZ - 1000 USD

Tổng giá trị: 4000 USD

Yêu cầu:

- Sử dụng class Product và tạo thêm class Cart để quản lý giỏ hàng.
- Thêm phương thức tính tổng giá trị các sản phẩm trong giỏ hàng và hiển thị trên trang.

Bài 3: Quản lý sách

Quản Lý Sách

Lập trình web

Ảnh

Thêm Sách

127.0.0.1:5500 says

Sách đã được thêm thành công.

OK

Hiển Thị Tất Cả Sách

Sắp Xếp Sách Theo Tiêu Đề

Tìm kiếm theo tiêu đề

Tìm Kiếm

Tiêu Đề cần cập nhật

Tên tác giả mới

Cập Nhật Tác Giả

Vị trí cần xóa

Xóa Sách

1. Lập trình JavaScript - Ly

2. Cơ sở dữ liệu - An

3. Nhập môn kỹ thuật lập trình - Bình

4. Hệ quản trị cơ sở dữ liệu - Chi

5. Anh Văn 1 - Tú

Yêu cầu:

- Tạo một lớp **Sách** gồm có các thông tin sau:
  - Tiêu Đề
  - Tác giả

Chương trình phải có khả năng lưu trữ 1000 cuốn sách và người dùng sẽ được phép:

- Thêm dữ liệu cho một cuốn sách

- Hiển thị tất cả các sách đã nhập (chỉ tiêu đề và tác giả, trong cùng một dòng)
- Tìm kiếm sách có tiêu đề nhất định
- Sắp xếp danh sách sách theo tiêu đề
- Cập nhật tên tác giả mới khi biết tiêu đề sách
- Xóa sách ở một vị trí đã biết (ví dụ: sách số 6)
- Thoát khỏi chương trình

### Hiển thị tất cả sách

1. Lập trình JavaScript - Ly
2. Cơ sở dữ liệu - An
3. Nhập môn kỹ thuật lập trình - Bình
4. Hệ quản trị cơ sở dữ liệu - Chi
5. Anh Văn 1 - Tú
6. Lập trình web - Anh

### Sắp xếp sách theo tiêu đề

1. Anh Văn 1 - Tú
2. Cơ sở dữ liệu - An
3. Hệ quản trị cơ sở dữ liệu - Chi
4. Lập trình JavaScript - Ly
5. Lập trình web - Anh
6. Nhập môn kỹ thuật lập trình - Bình

### Tìm kiếm tiêu đề

## Quản Lý Sách

<input type="text" value="Lập trình web"/>
<input type="text" value="Anh"/>
<input type="button" value="Thêm Sách"/>
<input type="button" value="Hiển Thị Tất Cả Sách"/>
<input type="button" value="Sắp Xếp Sách Theo Tiêu Đề"/>
<input type="text" value="dữ liệu"/>
<input type="button" value="Tìm Kiếm"/>
<input type="text" value="Cơ sở dữ liệu - An"/>
<input type="text" value="Hệ quản trị cơ sở dữ liệu - Chi"/>

Cập nhật tác giả mới

Quản Lý Sách

Hiện Thị Tất Cả Sách

Sắp Xếp Sách Theo Tiêu Đề

dữ liệu

Tìm Kiếm

Cơ sở dữ liệu

Nhật

Cập Nhật Tác Giả

1. Anh Văn 1 - Tú

2. Cơ sở dữ liệu - Nhật

3. Hệ quản trị cơ sở dữ liệu - Chi

4. Lập trình JavaScript - Ly

5. Lập trình web - Anh

6. Nhập môn kỹ thuật lập trình - Bình

Xóa sách

Quản Lý Sách

Hiện Thị Tất Cả Sách

Sắp Xếp Sách Theo Tiêu Đề

dữ liệu

Tìm Kiếm

Cơ sở dữ liệu

Nhật

Cập Nhật Tác Giả

5

Xóa Sách

1. Anh Văn 1 - Tú

2. Cơ sở dữ liệu - Nhật

3. Hệ quản trị cơ sở dữ liệu - Chi

4. Lập trình JavaScript - Ly

5. Nhập môn kỹ thuật lập trình - Bình

Bài 4: Quản lý CD

Quản Lý CD

Hiện Thị Toàn Bộ CD

Tính Số Lượng CD

Tính Tổng Giá Thành

127.0.0.1:5500 says  
CD đã được thêm thành công.

OK

Hiện thị toàn bộ CD

Mã: 12345, Tựa: Happy Wedding, Ca Sĩ: Momo, Số Bài Hát: 3, Giá: 100.00
Mã: 54321, Tựa: Bước qua đời nhau, Ca Sĩ: Khởi My, Số Bài Hát: 5, Giá: 300.00
Mã: 34567, Tựa: Bên trên tầng lầu, Ca Sĩ: Tân duy Tân, Số Bài Hát: 2, Giá: 250.00

Tính số lượng CD

Mã: 12345, Tựa: Happy Wedding, Ca Sĩ: Momo, Số Bài Hát: 3, Giá: 100.00
Mã: 54321, Tựa: Bước qua đời nhau, Ca Sĩ: Khởi My, Số Bài Hát: 5, Giá: 300.00
Mã: 34567, Tựa: Bên trên tầng lầu, Ca Sĩ: Tân duy Tân, Số Bài Hát: 2, Giá: 250.00

Số lượng CD: 3

### Tính tổng giá thành

Mã: 12345, Tựa: Happy Wedding, Ca Sĩ: Momo, Số Bài Hát: 3, Giá: 100.00
Mã: 54321, Tựa: Bước qua đời nhau, Ca Sĩ: Khởi My, Số Bài Hát: 5, Giá: 300.00
Mã: 34567, Tựa: Bên trên tầng lầu, Ca Sĩ: Tân duy Tân, Số Bài Hát: 2, Giá: 250.00

Tổng giá thành: 650.00 VND

### Yêu cầu

- Viết chương trình xây dựng đối tượng CD gồm có các thuộc tính sau:
  - Mã CD là số nguyên (ràng buộc nhập chữ báo lỗi, mã CD có 5 số)
  - Tựa CD: chuỗi ký tự, (ràng buộc nhập từ 2 ký tự trở lên)
  - Ca sĩ: chuỗi ký tự,
  - Số bài hát: số nguyên ( $>0$ ) (ràng buộc số bài hát không được nhập âm)
  - Giá thành: số thực ( $>0$ ) (ràng buộc nhập giá thành không được âm)
- Viết các constructor để khởi tạo đối tượng CD.
- Xây dựng lớp lưu danh sách các CD (*dùng mảng*).
  - Phương thức thêm 1 CD vào danh sách, thêm thành công nếu không trùng mã CD và kích thước mảng còn cho phép.
  - Phương thức nhập từng đối tượng CD
  - Phương thức xuất toàn bộ danh sách.
  - Tính số lượng CD có trong danh sách
  - Tính tổng giá thành của các CD.

### Bài 5: Kế thừa

Lớp Animal là lớp cơ sở (base class) với các thuộc tính và phương thức chung cho tất cả các động vật như: name, age

Hai phương thức speak và getDetails

- Phương thức speak trả về tên và tiếng kêu
- Phương thức getDetails trả về tên và tuổi

Lớp Dog kế thừa từ lớp Animal và có thêm phương thức riêng là bark.

Lớp Cat cũng kế thừa từ lớp Animal và có thêm phương thức riêng là meow.

Hướng dẫn

Class Animal

```
class Animal {  
    constructor(name, age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    speak() {  
        return `${this.name} is making a noise.`;  
    }  
  
    getDetails() {  
        return `Name: ${this.name}, Age: ${this.age}`;  
    }  
}
```

Class Dog

```
class Dog extends Animal {  
    constructor(name, age, breed) {  
        super(name, age); // Gọi constructor của lớp cha  
        this.breed = breed;  
    }  
  
    bark() {  
        return `${this.name} is barking! Woof woof!`;  
    }  
  
    // Ghi đè (override) phương thức speak của lớp cha  
    speak() {  
        return this.bark();  
    }  
  
    getDetails() {  
        return `${super.getDetails()}, Breed: ${this.breed}`;  
    }  
}
```

Class cat



```
class Cat extends Animal {
  constructor(name, age, color) {
    super(name, age);
    this.color = color;
  }

  meow() {
    return `${this.name} is meowing! Meow meow!`;
  }

  // Ghi đè (override) phương thức speak của lớp cha
  speak() {
    return this.meow();
  }

  getDetails() {
    return `${super.getDetails()}, Color: ${this.color}`;
  }
}
```

Sử dụng class

```
// Tạo đối tượng Dog
const myDog = new Dog('Rex', 5, 'Golden Retriever');
console.log(myDog.getDetails()); // Name: Rex, Age: 5, Breed: Golden Retriever
console.log(myDog.speak()); // Rex is barking! Woof woof!

// Tạo đối tượng Cat
const myCat = new Cat('Whiskers', 3, 'Black');
console.log(myCat.getDetails()); // Name: Whiskers, Age: 3, Color: Black
console.log(myCat.speak()); // Whiskers is meowing! Meow meow!
```

## Bài 6: Quản lý hệ thống nhân viên

Yêu cầu:

### 1. Lớp Employee (Nhân viên):

- Thuộc tính:
  - name (Tên)
  - age (Tuổi)
  - salary (Lương cơ bản)
- Phương thức:
  - getDetails() trả về thông tin của nhân viên.
  - calculateSalary() tính lương thực tế (lương cơ bản cho lớp cha).

### 2. Lớp Manager (Quản lý) kế thừa từ Employee:

- Thuộc tính:
  - bonus (Thưởng)
- Ghi đè phương thức:
  - calculateSalary() trả về lương thực tế là lương cơ bản cộng với thưởng.

### 3. Lớp Developer (Nhà phát triển) kế thừa từ Employee:

- Thuộc tính:
  - programmingLanguages (Mảng ngôn ngữ lập trình)
- Ghi đè phương thức:
  - getDetails() trả về thông tin nhân viên kèm các ngôn ngữ lập trình.
  - calculateSalary() có thể tính thêm lương thưởng dựa trên số lượng ngôn ngữ lập trình mà nhà phát triển biết.

### 4. Lớp Company (Công ty):

- Thuộc tính:
  - employees (Mảng chứa các đối tượng Employee)
- Phương thức:
  - addEmployee() để thêm nhân viên vào công ty.
  - calculateTotalSalary() để tính tổng lương của tất cả nhân viên trong công ty.
  - getAllDetails() để trả về chi tiết của tất cả nhân viên.

Hướng dẫn:

```
class Employee {
    constructor(name, age, salary) {
        this.name = name;
        this.age = age;
        this.salary = salary;
    }

    getDetails() {
        return `Name: ${this.name}, Age: ${this.age}, Salary: ${this.salary}`;
    }

    calculateSalary() {
        return this.salary;
    }
}
```

```
// lớp con manager
class Manager extends Employee {
  constructor(name, age, salary, bonus) {
    super(name, age, salary);
    this.bonus = bonus;
  }

  calculateSalary() {
    return this.salary + this.bonus;
  }

  getDetails() {
    return `${super.getDetails()}, Bonus: ${this.bonus}`;
  }
}
```

```
// lớp developer
class Developer extends Employee {
  constructor(name, age, salary, programmingLanguages) {
    super(name, age, salary);
    this.programmingLanguages = programmingLanguages;
  }

  calculateSalary() {
    const additionalPay = this.programmingLanguages.length * 100;
    return this.salary + additionalPay;
  }

  getDetails() {
    return `${super.getDetails()}, Programming Languages: ${this.programmingLanguages.join(', ')}`;
  }
}
```

```
// company quản lý công ty
class Company {
  constructor() {
    this.employees = [];
  }

  addEmployee(employee) {
    this.employees.push(employee);
  }

  calculateTotalSalary() {
    return this.employees.reduce((total, employee) => total + employee.calculateSalary(), 0);
  }

  getAllDetails() {
    return this.employees.map(employee => employee.getDetails()).join('\n');
  }
}
```

```
// hàm main
// Tạo một số đối tượng nhân viên
const emp1 = new Employee('Alice', 30, 3000);
const manager1 = new Manager('Bob', 45, 5000, 2000);
const dev1 = new Developer('Charlie', 25, 4000, ['JavaScript', 'Python']);
const dev2 = new Developer('Dave', 28, 4500, ['C++', 'Java', 'Python']);

// Tạo đối tượng công ty và thêm nhân viên
const myCompany = new Company();
myCompany.addEmployee(emp1);
myCompany.addEmployee(manager1);
myCompany.addEmployee(dev1);
myCompany.addEmployee(dev2);

// Hiển thị thông tin tất cả nhân viên
console.log('All Employee Details:');
console.log(myCompany.getAllDetails());

// Tính tổng lương phải trả cho tất cả nhân viên
console.log('Total Salary:', myCompany.calculateTotalSalary());
```

### Bài 7: Quản lý sách

Thông tin về các loại sách:

- Sách giáo khoa: Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ).
  - ✓ Nếu tình trạng sách là mới thì: thành tiền = số lượng \* đơn giá.
  - ✓ Nếu tình trạng sách là cũ thì: thành tiền = số lượng \* đơn giá \* 50%
- Sách tham khảo: Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, thuế. Thành tiền = số lượng \* đơn giá + thuế

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các loại sách.
- Tính tổng thành tiền cho từng loại.
- Tính trung bình cộng đơn giá của các sách tham khảo.
- Xuất ra các sách giáo khoa của nhà xuất bản X.
- Xóa sách khi biết mã sách
- Cho biết sách nào có thành tiền lớn nhất.
- Sắp xếp tăng dần các loại sách theo số lượng

### Bài 8: Quản lý hóa đơn điện

Thông tin bao gồm các loại khách hàng:

- Khách hàng Việt Nam: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (số KW tiêu thụ), đơn giá, định mức. Thành tiền được tính như sau:
  - ✓ Nếu số lượng ≤ định mức thì: thành tiền = số lượng \* đơn giá.
  - ✓ Ngược lại thì: thành tiền = số lượng \* đơn giá \* định mức + số lượng KW vượt định mức \* Đơn giá \* 2.5.

- Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm theo định dạng dd/mm/yyyy), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng \* đơn giá.

Thực hiện các yêu cầu sau:

- Xây dựng các lớp với chức năng thừa kế thể hiện tính đa hình.
- Nhập xuất danh sách các hóa đơn khách hàng.
- Tính tổng số lượng cho từng loại khách hàng.
- Tính trung bình thành tiền của khách hàng người nước ngoài.
- Xuất ra các hoá đơn trong tháng 09 năm 2013 (của cả 2 loại khách hàng).
- Xóa khách hàng ra khỏi danh sách