

OOP trong JavaScript

--- o0o ---

SV thực hiện một số yêu cầu sau:

1. *Tạo class **DigitalProduct** kế thừa từ class **Product** (đã tạo ở Lab 04). Sản phẩm số không có số lượng tồn kho, nhưng có thêm thuộc tính **downloadLink**.*

Yêu cầu:

- Dùng extends để kế thừa
- Dùng super() để gọi constructor của lớp cha
- Ghi đè phương thức getInfo() để hiển thị link tải

Hướng dẫn:

Tạo class

```
class DigitalProduct extends Product {
  constructor(name, price, downloadLink) {
    super(name, price, Infinity); // sản phẩm số không giới hạn số lượng
    this.downloadLink = downloadLink;
  }

  getInfo() {
    return `${this.name} - ${this.price}₫\n🔗 Tải về: ${this.downloadLink}`
  }
}
```

Tạo và sử dụng object từ class vừa tạo

```
const ebook = new DigitalProduct("Ebook JavaScript", 99000, "https://example.com/js-ebook");
console.log(ebook.getInfo());
```

2. *Tạo class **AdminUser** kế thừa từ **Customer** (đã tạo ở Lab 04).*

Admin có thêm quyền `role = 'admin'` và phương thức `banUser(user)` để khóa tài khoản người dùng.

Gợi ý cách làm:

- Dùng extends Customer để kế thừa thông tin người dùng

- Dùng `super(name, email, phone)` trong constructor
- Thêm thuộc tính `role` mặc định là `'admin'`
- Phương thức `banUser(user)` có thể in ra thông báo: "Người dùng [tên] đã bị khóa bởi admin [tên admin]"

3. GV có thể cho thêm bài tập

==== Yêu cầu nộp bài

- ⇒ SV sử dụng ngôn ngữ javascript để thực hiện bài tập
- ⇒ Đóng gói lại và nộp bài lên LMS theo hướng dẫn của GV
- ⇒ KHÔNG NỘP BÀI XEM NHƯ 0 ĐIỂM.

--- Hết ---