Cải tiến thông qua đánh giá Cohesion, Coupling và SOLID của CartController

1.Cohension

getAmount	Functional Cohesion	Tính tổng tiền của các sản phẩm được chọn dựa trên số lượng và giá.	Cân nhắc đưa phương thức này vào lớp Cart để giảm trách nhiệm của CartController.

```
//Chuyển từ CartController vào lớp Cart
20215458-phuongnm
public int getAmount() {
    return listCartMedia.stream() Stream<CartMedia>
            .filter(CartMedia::isChecked)
            .mapToInt(cartMedia -> cartMedia.getQuantity() * cartMedia.getPrice()) IntStream
            .sum();
}
```

checkAvailabilityOfSpecificProduct	Functional Cohesion	Kiểm tra tính khả dụng của sản phẩm trong kho so với yêu cầu.	Cân nhắc chuyển logic này sang CartMedia hoặc một class riêng biệt.

```
//Chuyển từ CartController vào CartMedia
3 usages  20215458-phuongnm
public boolean checkAvailability() { return quantity <= media.getQuantity(); }
```

addMediaToCart	Logical Cohesion	Kiểm tra các điều kiện liên quan (sản phẩm đã có trong giỏ, số lượng khả dụng) rồi thêm sản phẩm vào danh sách.	Chuyển một phần logic kiểm tra này sang lớp Cart hoặc sử dụng các phương thức riêng trong CartMedia.

```
//Chuyển từ CartController sang Cart
2 usages  20215458-phuongnm
public void addMediaToCart(CartMedia media) throws CartAlreadyInCartException, NotEnoughMediaException {
    if (isMediaInCart(media)) {
        throw new CartAlreadyInCartException();
    }

    if (!media.checkAvailability()) {
        throw new NotEnoughMediaException("Not enough media");
    }

    listCartMedia.add(media);
}
```

updateMediaCheck	Logical Cohesion	Đổi trạng thái "checked" của sản phẩm trong danh sách dựa trên trạng thái hiện tại.	Sử dụng các phương thức phụ trợ trong CartMedia để tăng tính rõ ràng.

```
//Sử dụng phương thức phụ trợ trong CartMedia
1 usage   20215458-phuongnm
public void updateMediaCheck(CartMedia media) { media.toggleChecked(); }
```

2.Coupling

CartController	Cart	Data Coupling	Phụ thuộc trực tiếp vào Cart để lấy danh sách CartMedia và quản lý giỏ hàng.	Sử dụng Dependency Injection để giảm phụ thuộc trực tiếp.

Truyền đối tượng Cart vào lớp CartController thay vì để CartController tự khởi tạo đối tượng Cart.

```
//Constructor nhận Cart như tham số
2 usages   20215458-phuongnm
public CartController(Cart cart) {
    this.cart = cart;
    this.cartManager = new CartManager(cart);
    this.cartValidator = new CartValidator();
    this.addMediaStrategy = new SimpleAddMediaStrategy(cart);
    this.removeMediaStrategy = new SimpleRemoveMediaStrategy(cart);
    this.checkAvailabilityStrategy = new SimpleCheckAvailabilityStrategy();
}
```

Cập nhật HomeScreenController

```
1 usage   20215458-phuongnm +1
private void init() {
    mediaRepository = new MediaRepositoryImpl();
    cart = Cart.getCart(); // Lấy Cart từ Cart singleton
    cartController = new CartController(cart); // Truyền Cart vào CartController
}
```

Cập nhật CartScreenController

```
20215458-phuongnm +1
@Override
public <T> void initData(T... data) {
    // To do
    cartFlowpane.setVgap(10);
    cart = Cart.getCart(); // Lấy Cart từ Cart singleton
    cartController = new CartController(cart); // Truyền Cart vào CartController
    List<CartMedia> cartMedia = cart.getListCartMedia();
    for(CartMedia cartMediaItem : cartMedia) {
```

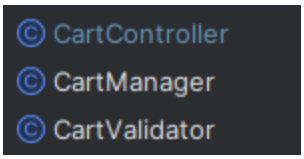
CartController	CartMedia	Data Coupling	Gọi phương thức và thuộc tính của CartMedia (như isChecked, getQuantity, getPrice).	Duy trì trạng thái hiện tại, nhưng đảm bảo CartMedia được kiểm soát rõ ràng hơn thông qua các interface cụ thể.

```
//Implements Interface
20215459-phuongvv +1 *
public class CartMedia implements ICartMedia {
    4 usages
    private Media media;
    4 usages
    private int quantity;
    3 usages
    private int price;
    3 usages
    private boolean checked = true;
```

3.Solid

SRP	Vi phạm	CartController quản lý cả trạng thái giỏ hàng, kiểm tra sản phẩm, xử lý logic đặt hàng.	Phân chia thành các lớp riêng biệt, ví dụ: CartManager (quản lý giỏ hàng), CartValidator (xác minh sản phẩm).

Tách CartController làm các lớp riêng biệt, CartController là lớp điều khiển, chịu trách nhiệm xử lý các yêu cầu từ người dùng hoặc hệ thống. Nó sử dụng CartManager, CartValidator như một công cụ để thực hiện các thao tác trên giỏ hàng



OCP	Vi phạm	Nếu thêm loại giỏ hàng hoặc logic xử lý mới, cần thay đổi code của CartController.	Sử dụng Strategy Pattern cho các logic xử lý, giúp mở rộng mà không cần sửa đổi lớp.

Cập nhật CartController

```
    }
    // Có thể thêm phương thức để thay đổi chiến lược khi cần thiết
    no usages  20215458-phuongnm
    public void setAddMediaStrategy(AddMediaStrategy strategy) {
        this.addMediaStrategy = strategy;
    }

    no usages  20215458-phuongnm
    public void setRemoveMediaStrategy(RemoveMediaStrategy strategy) {
        this.removeMediaStrategy = strategy;
    }

    no usages  20215458-phuongnm
    public void setCheckAvailabilityStrategy(CheckAvailabilityStrategy strategy) {
        this.checkAvailabilityStrategy = strategy;
    }
}
```