Contents

[Danh mục hình ảnh 1](#_Toc54597891)

[Bài tập 4.1: Sửa phương thức getTotalCost của lớp com.oms.bean.Order theo đúng yêu cầu đặt ra của hệ thống thương mại điện tử. Để đơn giản, khách hàng được xem là ở nội thành Hà Nội, nếu địa chỉ khách hàng có chứa xâu (không phân biệt hoa thường) “HN” hoặc “Hà Nội”. Tương tự, khách hàng được xem là ở nội thành TP. HCM nếu địa chỉ khách hàng có chứa xâu (không phân biệt hoa thường) “HCM” hoặc “Hồ Chí Minh”. 2](#_Toc54597892)

[Bài tập 4.2: Thiết kế các test case cho getTotalCost theo phương pháp kiểm thử hộp đen và kiểm thử hộp trắng. 2](#_Toc54597893)

[Bài tập 4.3: Tham khảo ví dụ mẫu được cung cấp, bạn cần cài đặt 3 lớp: (1) lớp TotalCostWhiteBoxTest ứng với các test case thiết kế cho getTotalCost theo phương pháp kiểm thử hộp đen, (2) lớp TotalCostBlackBoxTest ứng với các test case thiết kế cho getTotalCost theo phương pháp kiểm thử hộp trắng, và (3) lớp TotalCostTestSuite chạy các test case của 2 lớp trên. Bạn cần tự tạo 1 package có tên phù hợp và đặt các lớp này vào package đó. Lớp TotalCostBlackBoxTest và TotalCostWhiteBoxTest cần sử dụng kỹ thuật kiểm thử tham số hóa (tham khảo lớp ParameterizedBookApiTest). 2](#_Toc54597894)

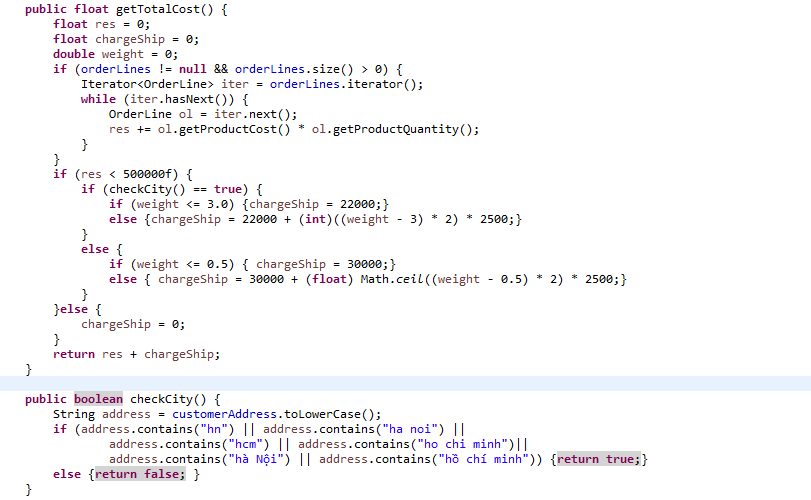
[Bài tập 4.4: Tự thiết kế kịch bản kiểm thử cho một module tùy chọn khác. 2](#_Toc54597895)

# Danh mục hình ảnh

### Bài tập 4.1: Sửa phương thức getTotalCost của lớp com.oms.bean.Order theo đúng yêu cầu đặt ra của hệ thống thương mại điện tử. Để đơn giản, khách hàng được xem là ở nội thành Hà Nội, nếu địa chỉ khách hàng có chứa xâu (không phân biệt hoa thường) “HN” hoặc “Hà Nội”. Tương tự, khách hàng được xem là ở nội thành TP. HCM nếu địa chỉ khách hàng có chứa xâu (không phân biệt hoa thường) “HCM” hoặc “Hồ Chí Minh”

Sửa phương thức getTotalCost():

* Thêm trường : weight vào OrderLine



Bài tập 4.2: Thiết kế các test case cho getTotalCost theo phương pháp kiểm thử hộp đen và kiểm thử hộp trắng

Sinh các Test case: Do hàm getTotalCost() phụ thuộc vào 3 tham số: address, price và weight

Kiểm tra theo kiểu hộp đen (không biết các thành phần trong đó):

(weight, price, address, ketQuaMongDoi) =

(5,600000,”Hai Phong”, 600000),

(3,300000,”Ha noi”,322000),

(5,300000,”Hcm”,332000),

(0.02, 400000,”Hai Duong”,430000),

(2,200000,”Hai Duong”,237500),

(30,700000,”Ca Mau”,700000);

Kiểm tra theo kiểu hộp trắng:

(weight, price, address, ketQuaMongDoi) =

(null, null, null, 0),

(5, null, “”, 0),

(null, null, “”, 0),

(5, 600000,”Hai Phong”, 600000),

(3, 300000,”Ha noi”,322000),

(5, 300000,”Hcm”,332000),

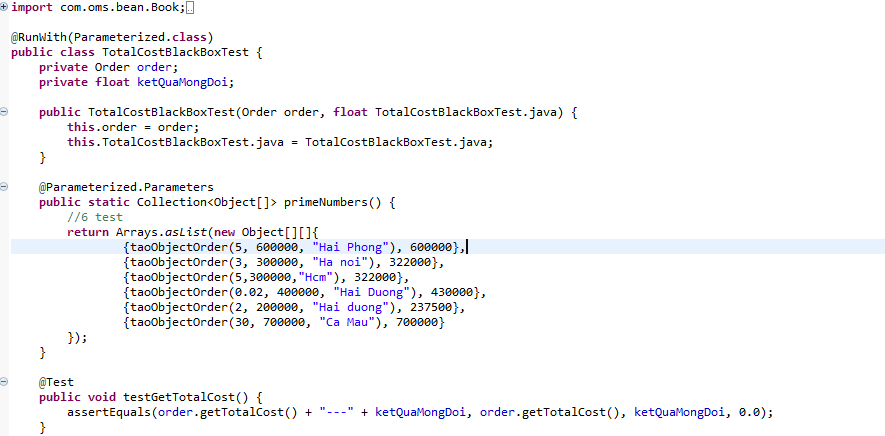
(0.1, 400000,”Hai Duong”,430000),

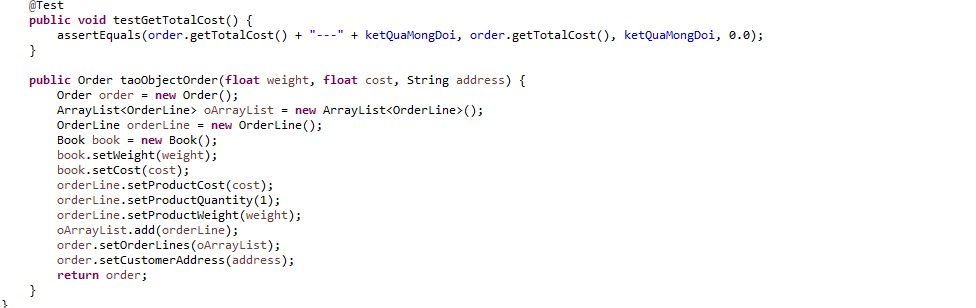
(1,200000,”Hai Duong”, 237500),

(30,700000,”Ca Mau”,700000);

## Bài tập 4.3: Tham khảo ví dụ mẫu được cung cấp, bạn cần cài đặt 3 lớp: (1) lớp TotalCostWhiteBoxTest ứng với các test case thiết kế cho getTotalCost theo phương pháp kiểm thử hộp đen, (2) lớp TotalCostBlackBoxTest ứng với các test case thiết kế cho getTotalCost theo phương pháp kiểm thử hộp trắng, và (3) lớp TotalCostTestSuite chạy các test case của 2 lớp trên. Bạn cần tự tạo 1 package có tên phù hợp và đặt các lớp này vào package đó. Lớp TotalCostBlackBoxTest và TotalCostWhiteBoxTest cần sử dụng kỹ thuật kiểm thử tham số hóa (tham khảo lớp ParameterizedBookApiTest).

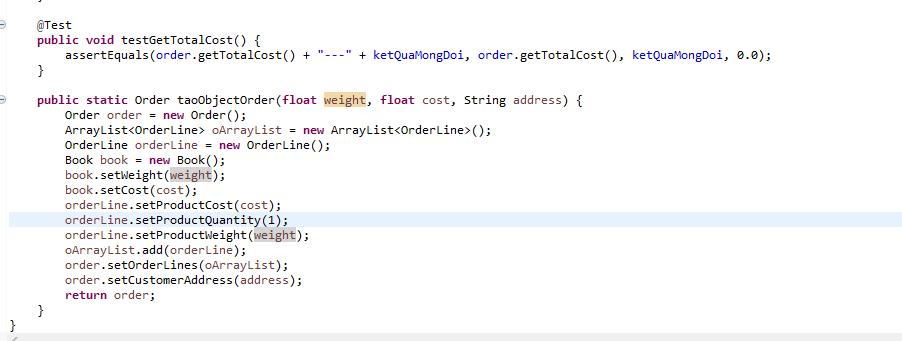
* TotalCostBlackBoxTest:



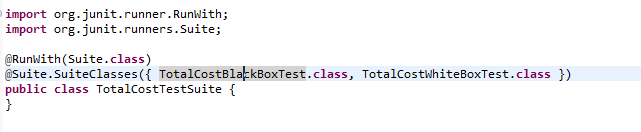


* TotalCostWhiteBoxTest:





* TotalCodeTestSuite:



## Bài tập 4.4: Tự thiết kế kịch bản kiểm thử cho một module tùy chọn khác.