

Projet Zéro

Lien Github :

https://github.com/Florianng/oc_project-0.git

1.Ajout de deux produits à la liste

```
Repository
public class ProductRepository {

    private static Map<Integer, Product> products = new HashMap<>();

    private static void generateProductData() {
        int id=0;
        products.put(++id, new Product(id, quantity: 10, price: 92.50, name: "Echo Dot", description: "(2nd Generation) - Black"));
        products.put(++id, new Product(id, quantity: 20, price: 9.99, name: "Anker 3ft / 0.9m Nylon Braided", description: "Tangle-Free Micro USB Cable"));
        products.put(++id, new Product(id, quantity: 30, price: 69.99, name: "JVC HAFXBR Headphone", description: "Riptidz, In-Ear"));
        products.put(++id, new Product(id, quantity: 40, price: 32.50, name: "VTech CS6114 DECT 6.0", description: "Cordless Phone"));
        products.put(++id, new Product(id, quantity: 50, price: 895.00, name: "NOKIA OEM BL-5J", description: "Cell Phone"));
        // Add product 1
        products.put(++id, new Product(id, quantity: 60, price: 49.00, name: "Google Home mini", description: "Black color"));
        // Add product 2
        products.put(++id, new Product(id, quantity: 70, price: 649.95, name: "Amp YAMAHA RX-1000", description: "Titan color"));
    }
}
```

Ajout des deux lignes de code sur Intelli J.

Nous vérifions bien l'ajout de ces produits sur le navigateur.

Liste des produits

Produit	Description du produit	Détails
Amp YAMAHA RX-1000	Titan color	ID-7
Anker 3ft / 0.9m Nylon Braided	Tangle-Free Micro USB Cable	ID-2
Echo Dot	(2nd Generation) - Black	ID-1
Google Home mini	Black color	ID-6
JVC HAFXBR Headphone	Riptidz, In-Ear	ID-3
NOKIA OEM BL-5J	Cell Phone	ID-5
VTech CS6114 DECT 6.0	Cordless Phone	ID-4

Valeur globale de l'inventaire : 123456.78

Produit

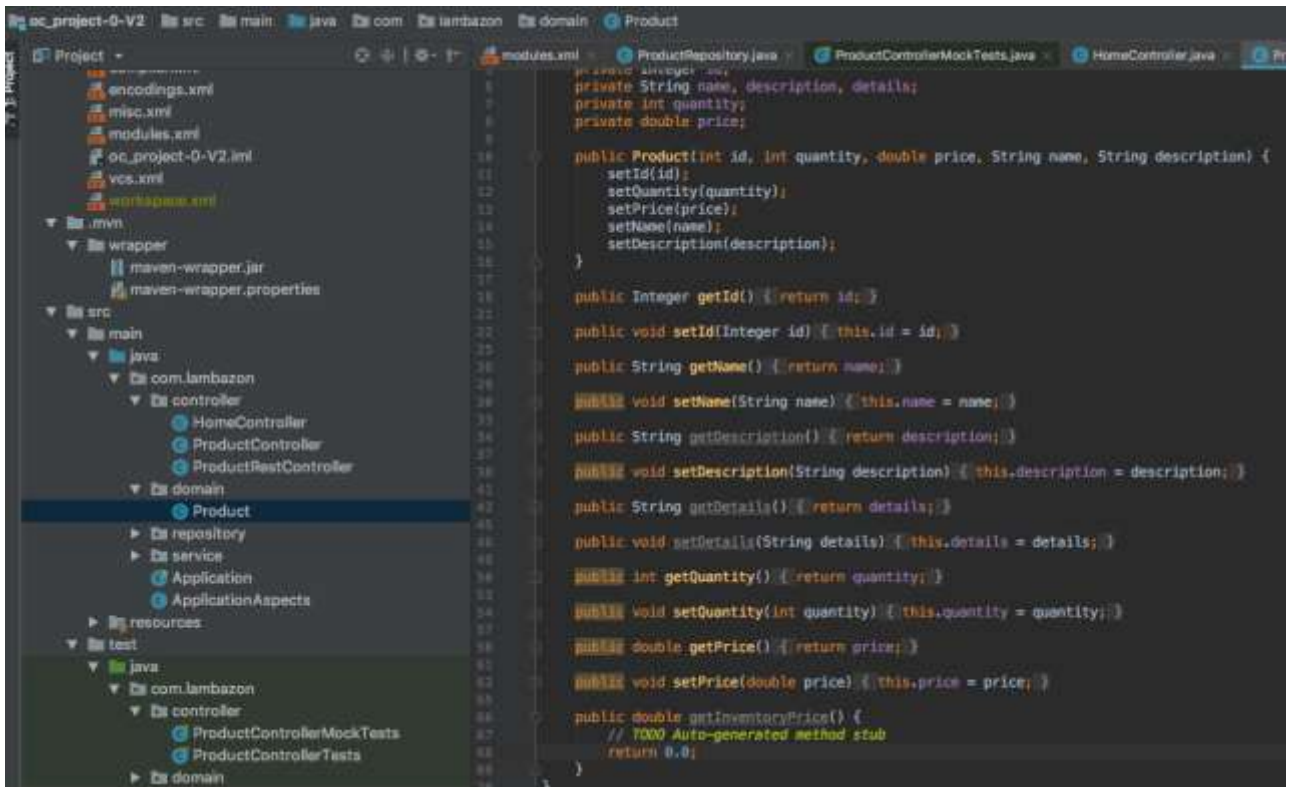
Détails

Identifiant	1
Prénom	Arp. Khatoun (01/10/2000)
Description du produit	Chapeau
Quantité	10
Prix	4000,00
Montant total	40000
Retour à la liste	

Ecran avec l'ajout de deux articles à la liste.

Détails du produit OK

2. Correction du bug valeur stock



```

1  private Integer id;
2  private String name, description, details;
3  private int quantity;
4  private double price;
5
6
7
8
9
10
11 public Product(int id, int quantity, double price, String name, String description) {
12     setId(id);
13     setQuantity(quantity);
14     setPrice(price);
15     setName(name);
16     setDescription(description);
17 }
18
19 public Integer getId() { return id; }
20
21 public void setId(Integer id) { this.id = id; }
22
23 public String getName() { return name; }
24
25 public void setName(String name) { this.name = name; }
26
27 public String getDescription() { return description; }
28
29 public void setDescription(String description) { this.description = description; }
30
31 public String getDetails() { return details; }
32
33 public void setDetails(String details) { this.details = details; }
34
35 public int getQuantity() { return quantity; }
36
37 public void setQuantity(int quantity) { this.quantity = quantity; }
38
39 public double getPrice() { return price; }
40
41 public void setPrice(double price) { this.price = price; }
42
43 public double getInventoryPrice() {
44     // TODO Auto-generated method stub - Valeur total produit = Quantité produit * Prix Produit
45     return price;
46 }

```

Ajout calcul stock produit

```
public double getInventoryPrice() { this.price= this.quantity * this.price;
```

```
// TODO Auto-generated method stub - quantity total product = Quantity product * Price
```

Product

```
return price;
```

```
}
```

Produit

Details

Identifiant

Prénom

Description du produit

Quantité

Prix

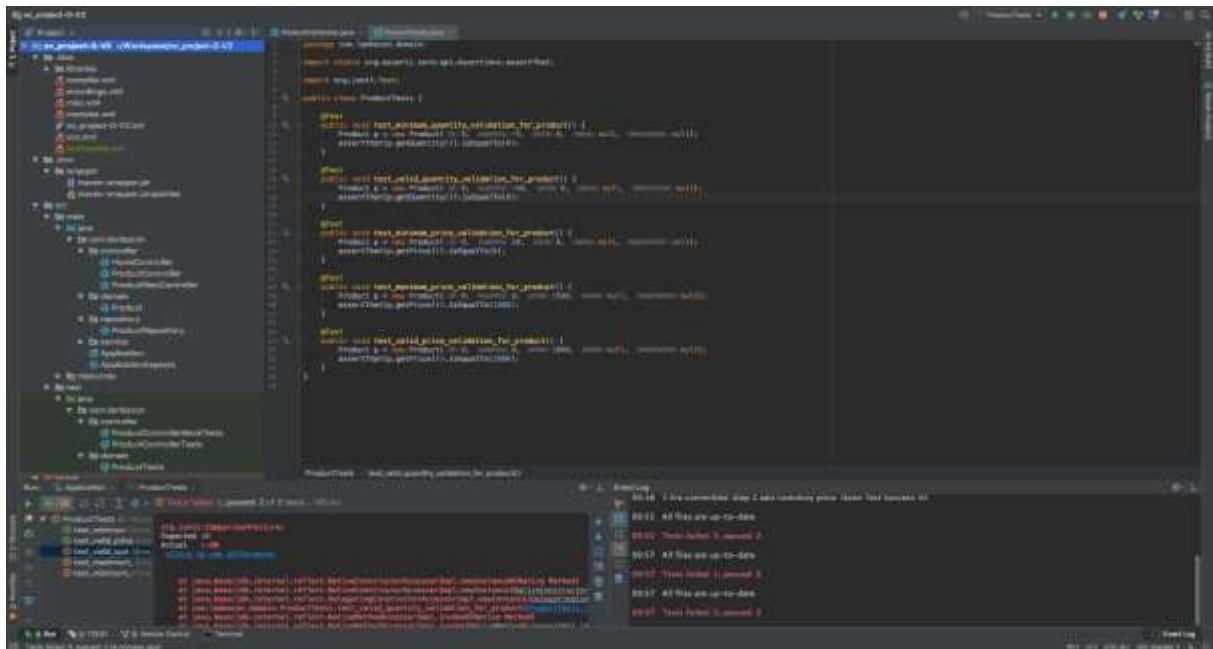
Montant total

[Retour à la liste](#)

Nous constatons que les calculs sont corrects sur les fiches produits sur le navigateur.

3. Tests validation produits

Nous pouvons constater que deux tests sont OK et trois autres sont « KO »



KO

@Test

```
public void test_minimum_quantity_validation_for_product() {
```

```
    Product p = new Product(0, -5, 0, null, null);
```

```
    assertThat(p.getQuantity()).isEqualTo(0);
```

```
}
```

KO

@Test

```
public void test_valid_quantity_validation_for_product() {
```

```
    Product p = new Product(0, -50, 0, null, null);
```

```
    assertThat(p.getQuantity()).isEqualTo(0);
```

```
}
```

OK

@Test

```
public void test_minimum_price_validation_for_product() {  
    Product p = new Product(0, 50, 0, null, null);  
    assertThat(p.getPrice()).isEqualTo(0);  
}
```

KO

@Test

```
public void test_maximum_price_validation_for_product() {  
    Product p = new Product(0, 0, 1500, null, null);  
    assertThat(p.getPrice()).isEqualTo(1000);  
}
```

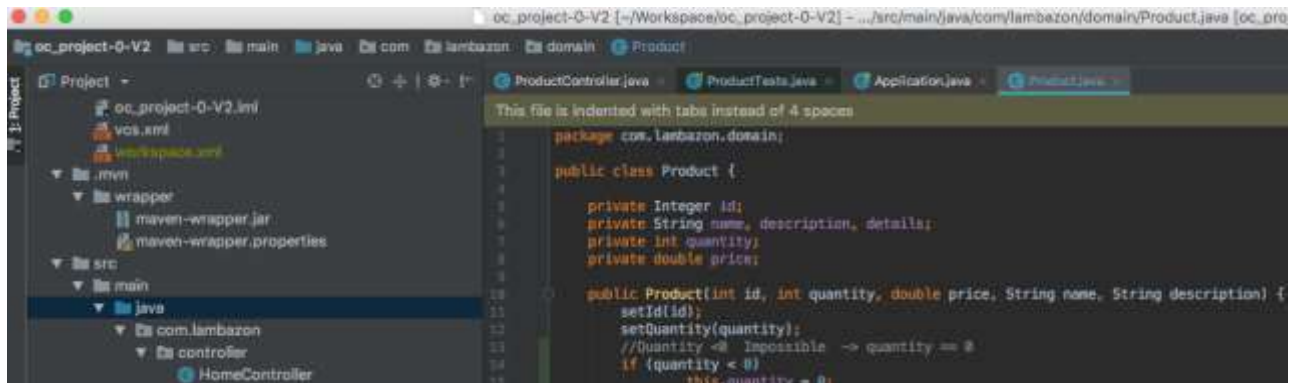
OK

@Test

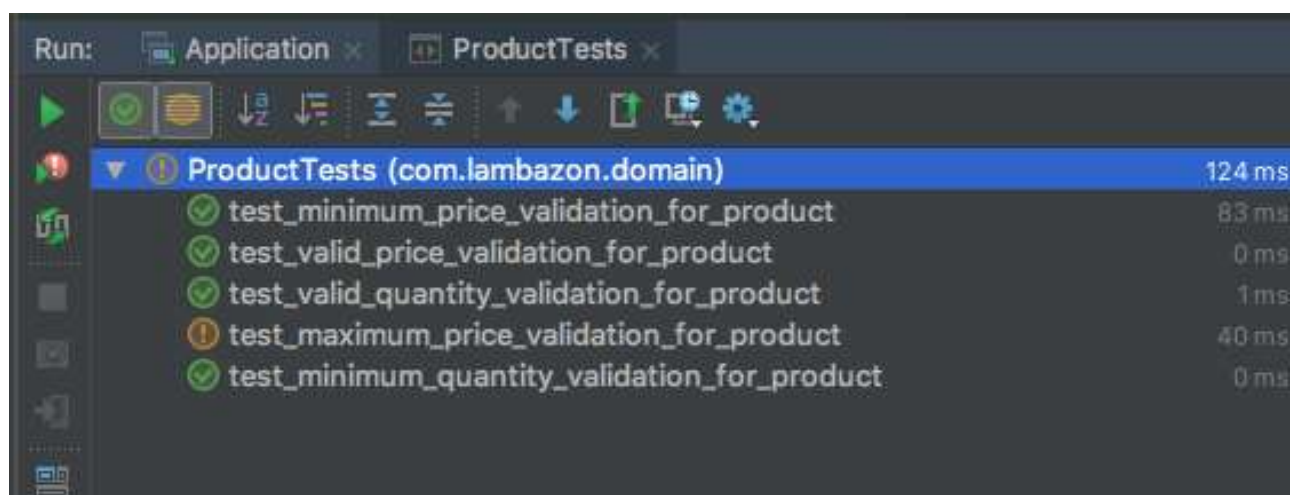
```
public void test_valid_price_validation_for_product() {  
    Product p = new Product(0, 0, 1000, null, null);  
    assertThat(p.getPrice()).isEqualTo(1000);  
}  
}
```

Modifions le fichier producttests.java

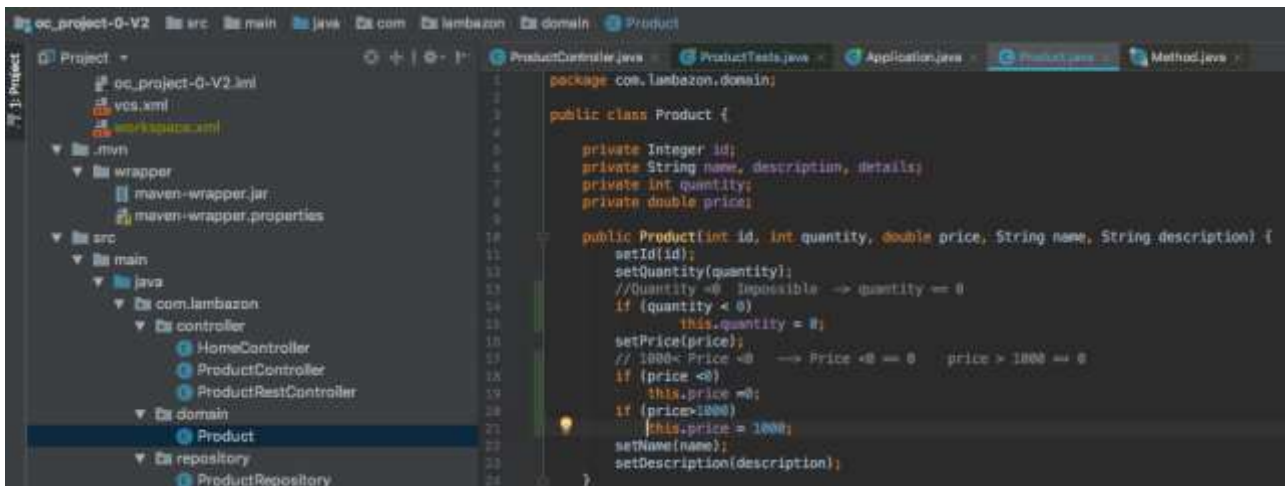
```
if (quantity < 0)
    this.quantity = 0;
```



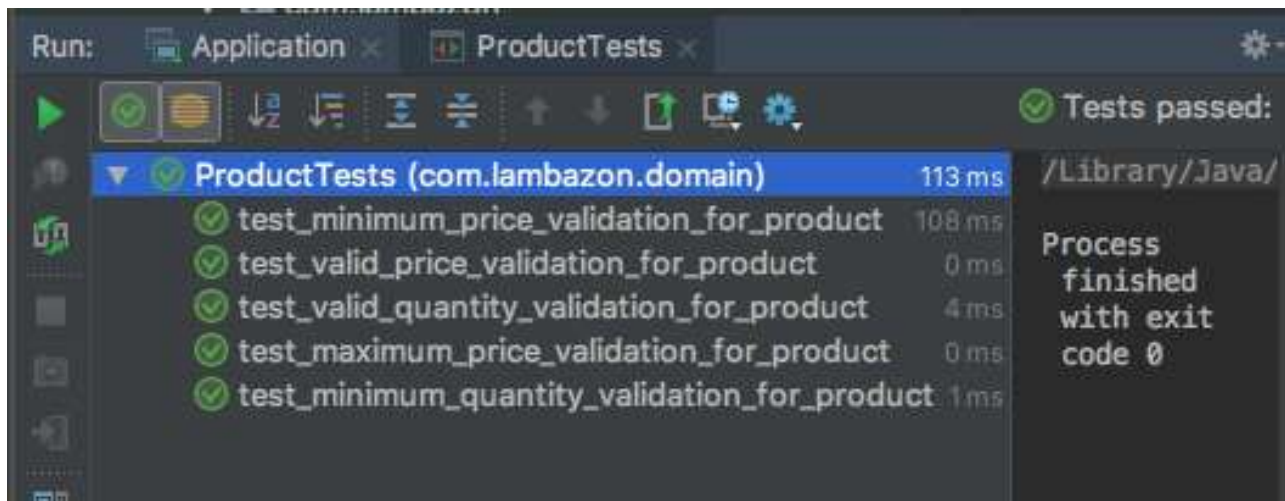
Test sur les quantités négatives



Nous avons donc plus qu'un test KO

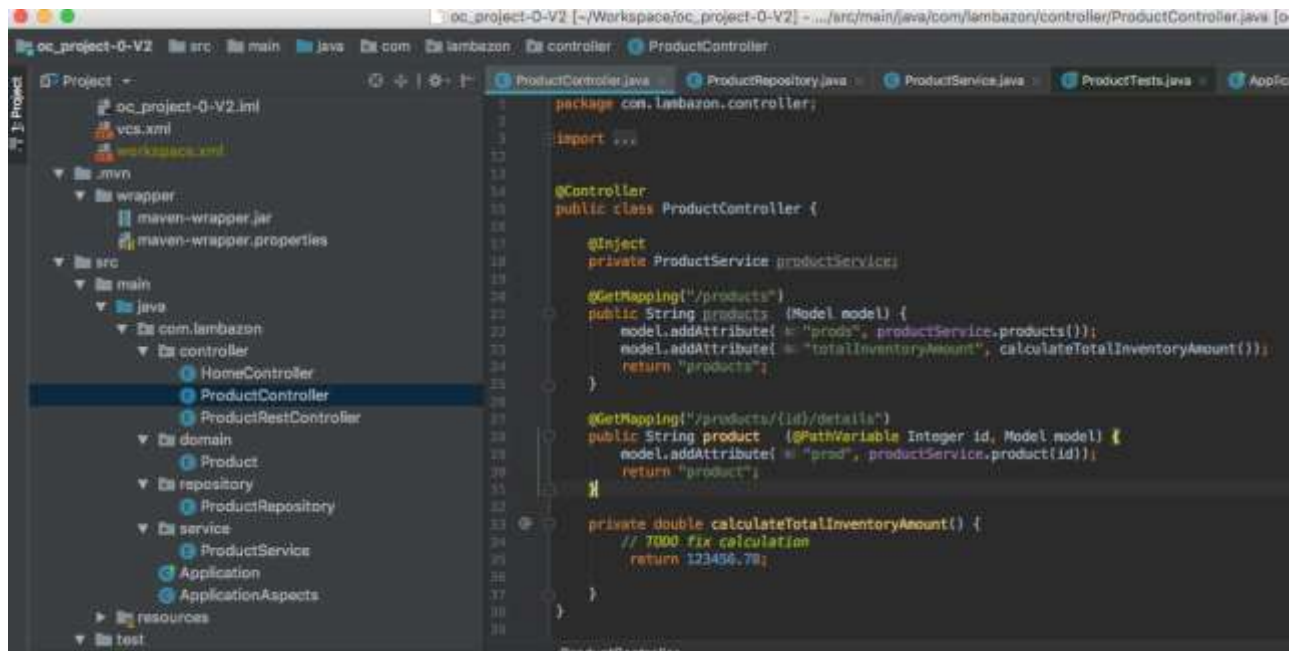


Autre modification du code



Ensemble des tests OK

4. Calcul valeur totale du stock



Productcontroller

```
Private double calculateTotalInventoryAmount() {
    // TODO: fix calculation - return 123456.78;
    double totalInventoryAmount = 0.0; //initialisation
    for (Product p : productService.products()) { // Loop Use productService
        totalInventoryAmount = p.getInventoryPrice() + totalInventoryAmount; // sum
    }
    return totalInventoryAmount;
}
```

Vérifions ce calcul sur le navigateur :

Liste des produits

Produit	Description du produit	Détails
Amp YAMAHA RX-1000	Titan color	ID-7
Anker 3ft / 0.9m Nylon Braided	Tangle-Free Micro USB Cable	ID-2
Echo Dot	(2nd Generation) - Black	ID-1
Google Home mini	Black color	ID-6
JVC HAFXBR Headphone	Riptidz, In-Ear	ID-3
NOKIA OEM BL-5J	Cell Phone	ID-5
VTech CS6114 DECT 6.0	Cordless Phone	ID-4

Valeur globale de l'inventaire : 97711.0

Les données du navigateur sont cohérentes. Le calcul est correct.

Articles	Quantité		
N°1	70	649,95	45 496,5
N°2	20	9,99	199,8
N°3	10	92,50	925
N°4	60	49	2940
N°5	30	69,99	2099,7
N°6	50	895	44750
N°7	40	32,5	1300
	280	1798,93	97711