CS 1083

Assignment #1

Author: Yulong Wang

Id: 3713596

1. Source Code

Product.java:

```java
/**
 * The Product class stored in Stock
 * @author Yulong Wang
 * @date 2021/09/15
 */
public class Product{
    /**
     * The initial id number, used to generate id for new products.
     */
    private static int totalId = 10001;
    /**
     * The id of product
     */
    private final int id;
    /**
     * The name of product
     */
    private String name;
    /**
     * The number of available stock.
     */
    private int numStock;
    /**
     *    The number of ordered number of product.
     */
    private int numOrdered;
    /**
     * The price of the product, round to 2 decimals.
     */
    private double price;

    /**
     * @param name The name of the product
     * @param price The price of the product
     * @param numStock The number of the product in stock.
     */
    public Product(String name, double price, int numStock) throws
IllegalArgumentException{
        if(price < 0 || numStock < 0){
            throw new IllegalArgumentException();
        }
        this.id = totalId++;
        this.name = name;
        this.numStock = numStock;
        this.numOrdered = 0;
        this.price = Math.round(price * 100.0) / 100.0;
    }
```

```java
/**
 * Receive the order and update numOrdered and numStock.
 * @return the number of stock after order completed.
 */
public int orderReceived(){
    this.numStock += this.numOrdered;
    this.numOrdered = 0;
    return this.numStock;
}

/**
 * Get the total value of the product.
 * @return double The total value of the product.
 */
public double getTotalValue(){
    return Math.round((this.numStock * this.price) * 100.0) / 100.0;
}

/**
 * Sell ONE product.
 * @return {@link Boolean} If sell succeed.
 */
public Boolean sellProduct(){
    if(this.numStock > 0){
        this.numStock -= 1;
        return true;
    }else{
        return false;
    }
}

/**
 * Get the id of the product.
 * @return int Id of the product.
 */
public int getId(){
    return this.id;
}

/**
 * Get the number of product currently in stock.
 * @return int Number of stock.
 */
public int getNumStock(){
    return this.numStock;
}

/**
 * Get the name of the product.
 * @return String name of product.
 */
```

```
    public String getName(){
        return this.name;
    }

    /**
     * Get the number of ordered product.
     * @return int Number of product ordered.
     */
    public int getNumOrdered(){
        return this.numOrdered;
    }

    /**
     * Get the price of the product.
     * @return double price.
     */
    public double getPrice(){
        return this.price;
    }

    /**
     * Set the number of ordered products.
     * @param numOrdered The number ordered.
     * @return Boolean If change applied.
     */
    public Boolean setNumOrdered(int numOrdered){
        if(numOrdered < 1){
            return false;
        }
        this.numOrdered = numOrdered;
        return true;
    }

    /**
     * Change the price of the product.
     * @param price
     * @return Boolean If changed applied.
     */
    public Boolean setPrice(double price){
        if(price < 0){
            return false;
        }
        this.price = Math.round(price * 100.0) / 100.0;
        return true;
    }

    /**
     * print summary of the product to the terminal.
     */
    public void summary(){
        System.out.printf("%-20s(id: %s)     qty: %-5s   %-5s ordered   $ %.2f \n",
                this.name,
```

```java
                this.id,
                this.numStock,
                this.numOrdered,
                this.price
        );
    }
}
```

Stock.java:

```java
/**
 * A Stock class that stores multiple Products.
 * @author Yulong Wang
 * @date 2021/09/15
 */
public class Stock{
    /**
     The max number of products can be stored in the stock
     */
    private int maxNumOfProducts;
    /**
     Current number of product in the stock
     */
    private int numOfProducts;
    /**
     The list stores the product objects.
     */
    private Product[] products;

    /**
     * init the Stock
     * @param maxNumOfProducts the max number of products this stock can store
     */
    public Stock(int maxNumOfProducts) throws IllegalArgumentException{
        if(maxNumOfProducts < 0){
            throw new IllegalArgumentException();
        }
        this.maxNumOfProducts = maxNumOfProducts;
        this.numOfProducts = 0;
        this.products = new Product[maxNumOfProducts];
    }

    /**
     * find a given product in the stock.
     * @param product the target product.
     * @return The index of the given product in the products array. If product not
found, return -1.
     */
    private int getProductIndex(Product product){
        for(int i = 0; i < this.maxNumOfProducts; i++){
            if(this.products[i] != null && this.products[i].getId() ==
product.getId()){
```

```java
                return i;
            }
        }
        return -1;
    }

    /**
     * Add a product to stock.
     * @param product The product need to be added.
     * @return {@link Boolean} Where the operation is successful.
     */
    public Boolean addProduct(Product product){
        if (this.numOfProducts > this.maxNumOfProducts){
            return false;
        }
        for(int i = 0; i < this.maxNumOfProducts; i++){
            if(this.products[i] == null){
                this.products[i] = product;
                return true;
            }
        }
        return false;
    }

    /**
     * Sell a product in the stock
     * @param product The product need to be sold.
     * @return {@link Boolean} If selling succeed.
     */
    public Boolean sellProduct(Product product){
        return product.sellProduct();
    }

    /**
     * Remove a product from the stock.
     * @param product The product need to be removed.
     * @return {@link Boolean} If operation succeed.
     */
    public Boolean removeProduct(Product product){
        int index = this.getProductIndex(product);
        if(index > -1){
            this.products[index] = null;
            return true;
        }
        return false;
    }

    /**
     * Check if a product is low in stock (stock < 5).
     * @param product The product need to be checked.
     * @return {@link Boolean} If the product need to be refilled.
     */
    public Boolean fillProductStock(Product product){
        int index = this.getProductIndex(product);
        if(index > 0 && this.products[index].getNumStock() < 5){
            product.setNumOrdered(10);
            return true;
        }
        return false;
    }

    /**
     * Get the total value of all product in the stock.
     * @return double The total value.
     */
    public String getTotalValue(){
        double totalValue = 0;
        for(int i = 0; i < this.maxNumOfProducts; i++){
            if(this.products[i] != null){
                totalValue += this.products[i].getTotalValue();
            }
        }
        return "$ " + Math.round(totalValue * 100.0) / 100.0;
    }

    /**
     * print summary of all products to the terminal.
     */
    public void summary(){
        for(int i=0; i<this.maxNumOfProducts; i++) {
            Product product = this.products[i];
            if (product != null) {
                product.summary();
            }
        }
    }
}
```

StockAndProductDriver.java:

```
/**
 * @author Yulong Wang
 * @date 2021/09/15
 */
public class StockAndProductDriver{
    public static void main(String[] args){
//         1. Create some products.
        System.out.println("1. Create some products.");
        Product p1 = new Product("iphone13", 699.00, 10);
        System.out.print("p1: ");
        p1.summary();
        Product p2 = new Product("ipad", 299, 3);
        System.out.print("p2: ");
        p2.summary();
        Product p3 = new Product("mac book pro", 1699.991, 6);
        System.out.print("p3: ");
        p3.summary();
        Product p4 = new Product("apple watch", 399.99, 8);
        System.out.print("p4: ");
        p4.summary();
        Product p5 = new Product("air pods", 199.123, 20);
        System.out.print("p5: ");
        p5.summary();
//        Try meaningless init
        try{
            System.out.print("Create product with negative price: ");
            Product wrongPrice = new Product("apple ball", -10, 10);
        }catch (IllegalArgumentException e){
            System.out.println(e);
        }

        try{
            System.out.print("Create product with negative stock: ");
            Product wrongStock = new Product("apple ball", 10, -10);
        }catch (IllegalArgumentException e){
            System.out.println(e);
        }

System.out.println("========================================================");
//         2. Create a Stock and add Product to the stock.
        System.out.println("2. Create a Stock and add Product to the stock.");
        Stock stock = new Stock(4);
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> create a Stock with 4 max number of products");
        System.out.println("-> adding p1: " + stock.addProduct(p1));
        System.out.println("-> adding p2: " + stock.addProduct(p2));
        System.out.println("-> adding p3: " + stock.addProduct(p3));
        System.out.println("-> adding p4: " + stock.addProduct(p4));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        3. Add more than max number of products allowed.
        System.out.println("3. Add more than max number of products allowed.");
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> adding p5: " + stock.addProduct(p5));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
System.out.println("========================================================");
//        4. Remove a product.
        System.out.println("4. Remove a product.");
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> remove p1: " + stock.removeProduct(p1));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        5. Add another product.
        System.out.println("5. Add another product.");
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> add p5: " + stock.addProduct(p5));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        6. Remove a product does not exist;
        System.out.println("6. Remove a product does not exist.");
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> remove p1: " + stock.removeProduct(p1));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        7. Sell a product.
        System.out.println("7. Sell a product.");
        System.out.println("-------------------------BEFORE------------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> sell p2: " + stock.sellProduct(p2));
        System.out.println("-------------------------AFTER------------------------ ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        8. Sell a product that is no more available.
        System.out.println("8. Sell a product that is no more available.");
        System.out.println("-------------------------BEFORE-----------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
//        Empty the stock.
        System.out.println("-> empty p5 stock");
        for(int i = p5.getNumStock(); i > 0; i--){
            stock.sellProduct(p5);
        }
        System.out.println("-------------------------AFTER----------------------- ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> sell p5: " + stock.sellProduct(p5));
        System.out.println("-------------------------AFTER----------------------- ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        9. Order new product running low.
        System.out.println("9. Order new product running low.");
        System.out.println("-------------------------BEFORE-----------------------
");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> order p2: " + stock.fillProductStock(p2));
        System.out.println("-------------------------AFTER----------------------- ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");
        System.out.println("-> receive order of p2: " + p2.orderReceived());
        System.out.println("-------------------------AFTER----------------------- ");
        stock.summary();
        System.out.println("------------------------------------------------------ ");

System.out.println("========================================================");
//        10. Display the total value of all products.
        System.out.println("10. Display the total value of all products.");
        System.out.println(stock.getTotalValue());
    }
}
```

2. Sample Output

```
1. Create some products.
p1: iphone13            (id: 10001)     qty: 10      0     ordered   $ 699.00
p2: ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
p3: mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
p4: apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
p5: air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
Create product with negative price: java.lang.IllegalArgumentException
Create product with negative stock: java.lang.IllegalArgumentException
========================================================
2. Create a Stock and add Product to the stock.
-------------------------BEFORE------------------------
------------------------------------------------------
-> create a Stock with 4 max number of products
-> adding p1: true
-> adding p2: true
-> adding p3: true
-> adding p4: true
-------------------------AFTER------------------------
iphone13            (id: 10001)     qty: 10      0     ordered   $ 699.00
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
3. Add more than max number of products allowed.
-------------------------BEFORE------------------------
iphone13            (id: 10001)     qty: 10      0     ordered   $ 699.00
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> adding p5: false
-------------------------AFTER------------------------
iphone13            (id: 10001)     qty: 10      0     ordered   $ 699.00
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
4. Remove a product.
-------------------------BEFORE------------------------
iphone13            (id: 10001)     qty: 10      0     ordered   $ 699.00
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> remove p1: true
-------------------------AFTER------------------------
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
5. Add another product.
-------------------------BEFORE------------------------
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> add p5: true
-------------------------AFTER------------------------
```

```
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
6. Remove a product does not exist.
-------------------------BEFORE------------------------
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> remove p1: false
-------------------------AFTER------------------------
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
7. Sell a product.
-------------------------BEFORE------------------------
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 3       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> sell p2: true
-------------------------AFTER------------------------
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
8. Sell a product that is no more available.
-------------------------BEFORE------------------------
air pods            (id: 10005)     qty: 20      0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> empty p5 stock
-------------------------AFTER------------------------
air pods            (id: 10005)     qty: 0       0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> sell p5: false
-------------------------AFTER------------------------
```

```
air pods            (id: 10005)     qty: 0       0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
9. Order new product running low.
-------------------------BEFORE------------------------
air pods            (id: 10005)     qty: 0       0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> order p2: true
-------------------------AFTER------------------------
air pods            (id: 10005)     qty: 0       0     ordered   $ 199.12
ipad                (id: 10002)     qty: 2       10    ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
-> receive order of p2: 12
-------------------------AFTER------------------------
air pods            (id: 10005)     qty: 0       0     ordered   $ 199.12
ipad                (id: 10002)     qty: 12      0     ordered   $ 299.00
mac book pro        (id: 10003)     qty: 6       0     ordered   $ 1699.99
apple watch         (id: 10004)     qty: 8       0     ordered   $ 399.99
------------------------------------------------------
========================================================
10. Display the total value of all products.
$ 16987.86
```