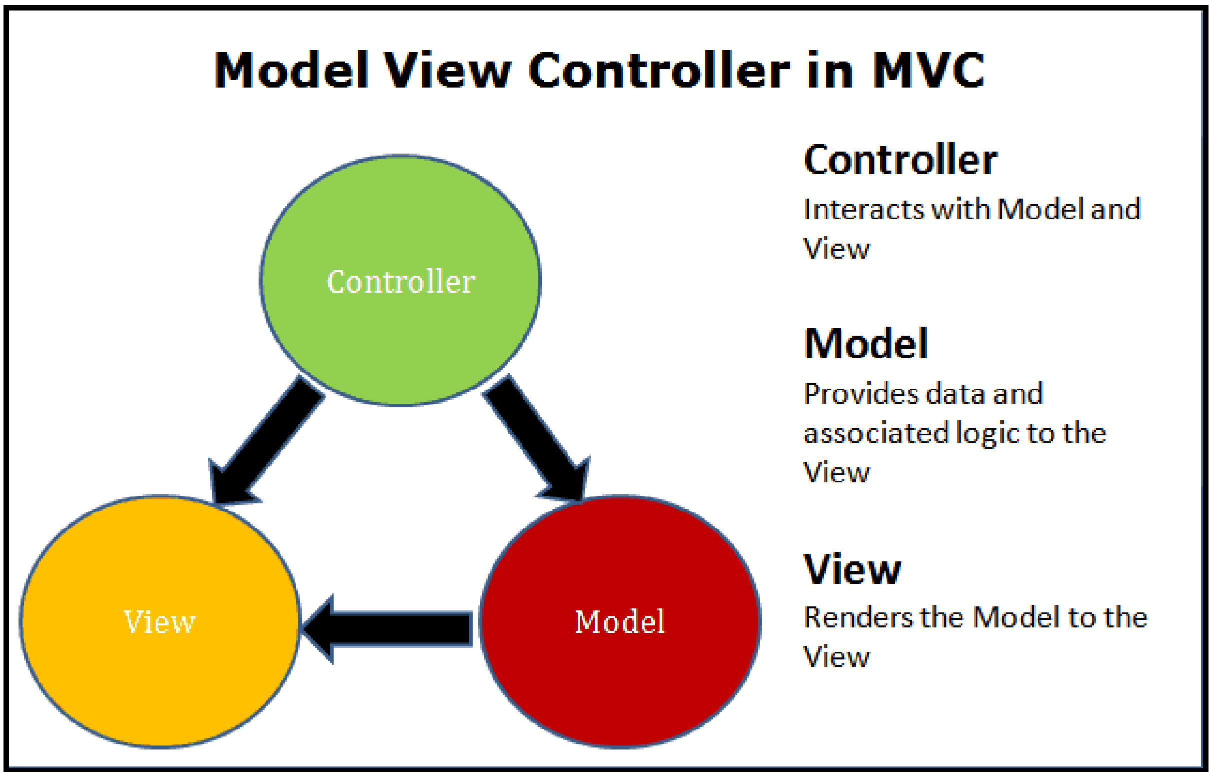
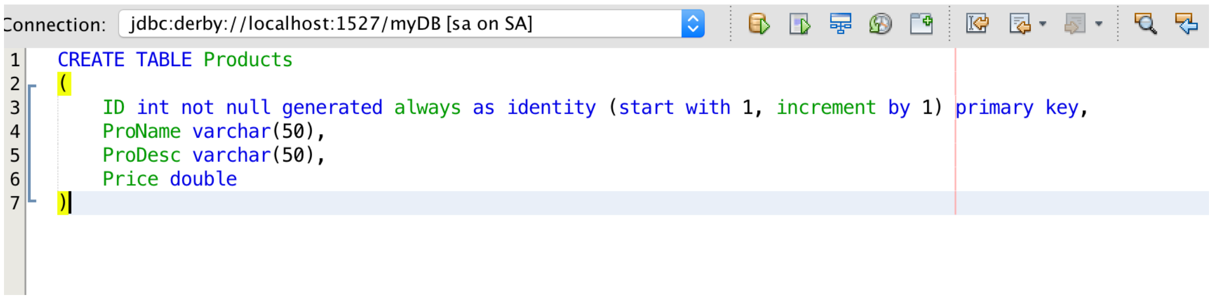
MVC CRUD demo

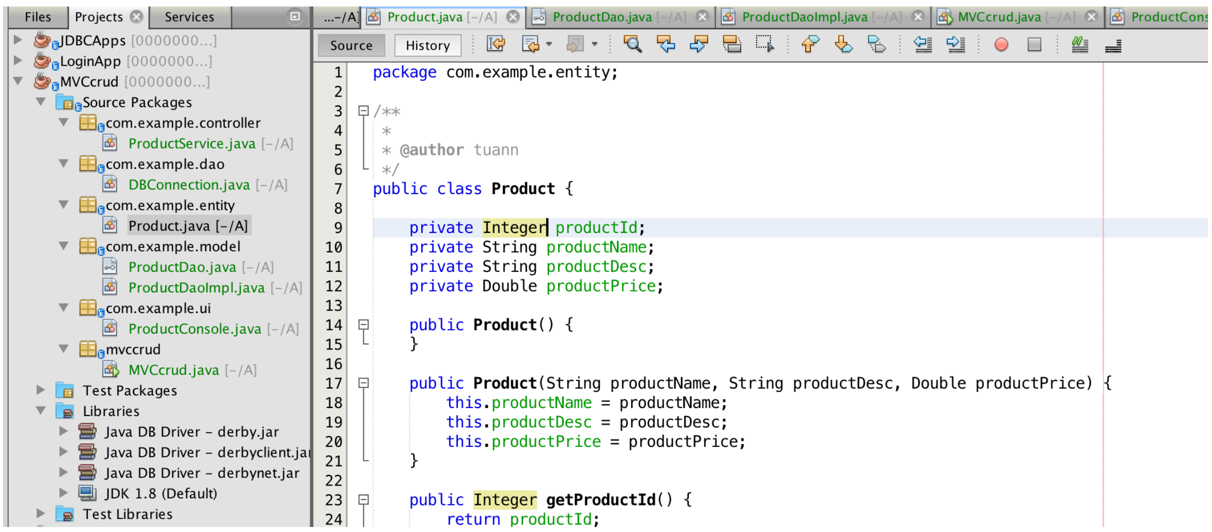

Model View Controller in MVC
Controller
Model
View
Controller
Interacts with Model and View
Model
Provides data and associated logic to the View
View
Renders the Model to the View

1.Tạo database với tên là MyDB với Table có tên Products cấu trúc như sau:



```
CREATE TABLE Products
(
    ID int not null generated always as identity (start with 1, increment by 1) primary key,
    ProName varchar(50),
    ProDesc varchar(50),
    Price double
)
```

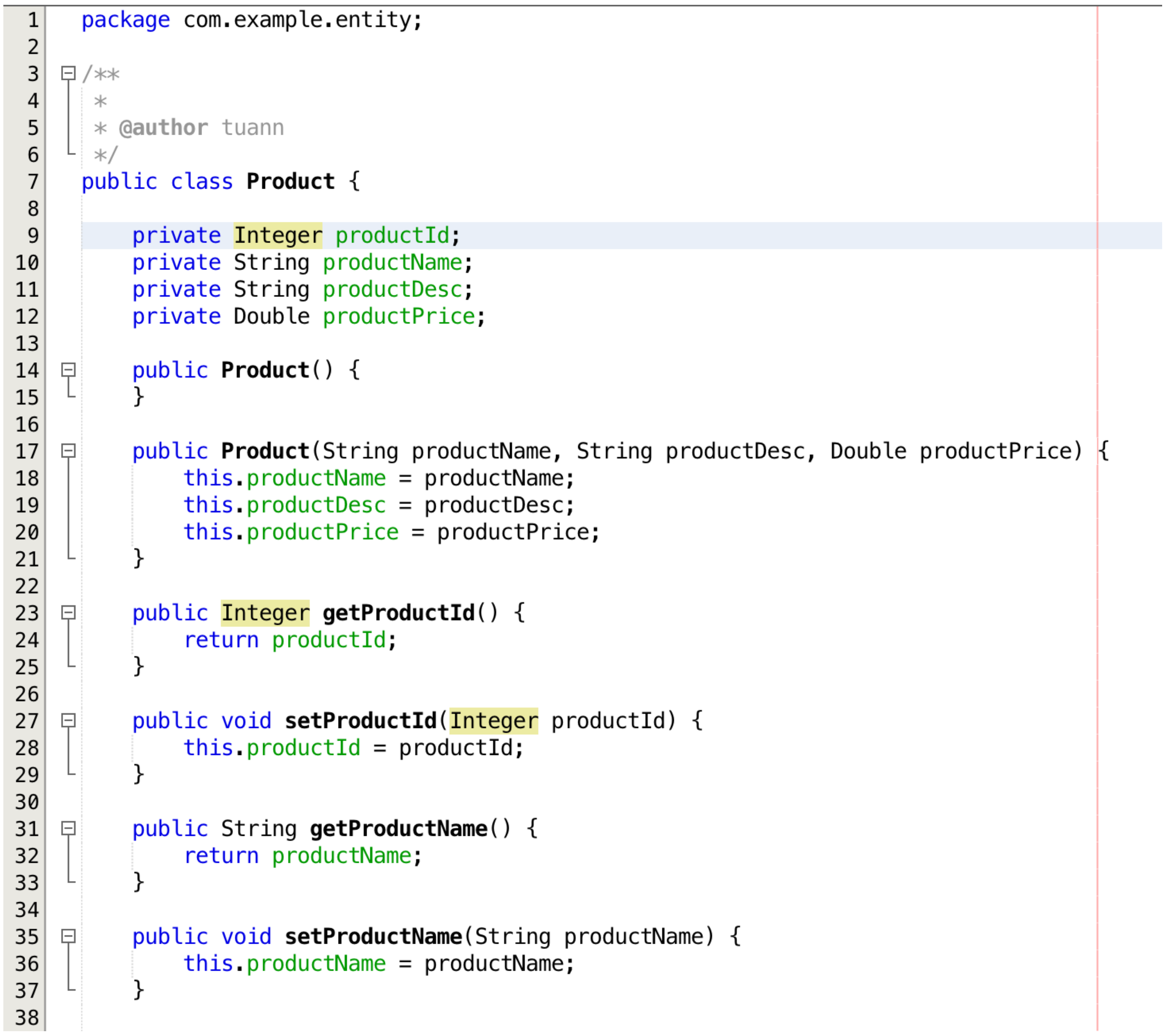
2. Tạo Project có cấu trúc như sau:



```
JDBCApps [0000000...]
LoginApp [0000000...]
MVCcrud [0000000...]
    Source Packages
        com.example.controller
            ProductService.java [-/A]
        com.example.dao
            DBConnection.java [-/A]
        com.example.entity
            Product.java [-/A]
        com.example.model
            ProductDao.java [-/A]
            ProductDaoImpl.java [-/A]
        com.example.ui
            ProductConsole.java [-/A]
        mvccrud
            MVCcrud.java [-/A]
    Test Packages
    Libraries
        Java DB Driver - derby.jar
        Java DB Driver - derbyclient.jar
        Java DB Driver - derbynet.jar
        JDK 1.8 (Default)
    Test Libraries
```

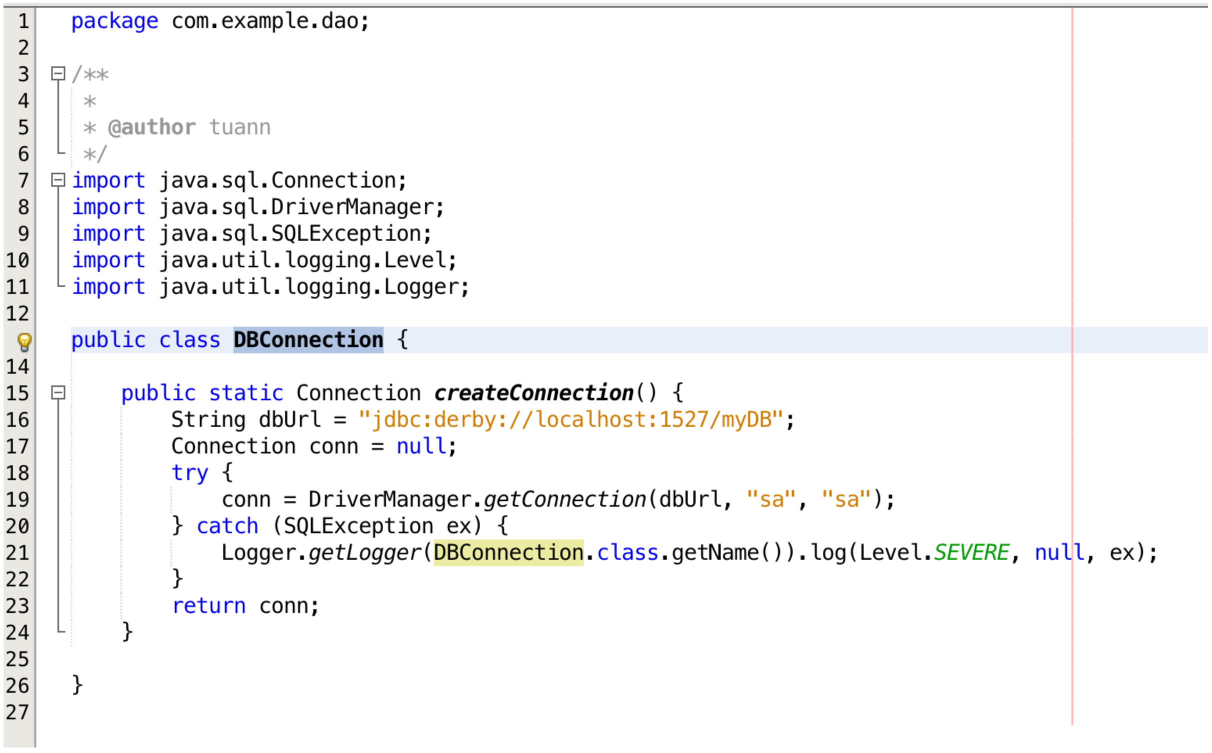
```java
package com.example.entity;

/**
 *
 * @author tuann
 */
public class Product {

    private Integer productId;
    private String productName;
    private String productDesc;
    private Double productPrice;

    public Product() {
    }

    public Product(String productName, String productDesc, Double productPrice) {
        this.productName = productName;
        this.productDesc = productDesc;
        this.productPrice = productPrice;
    }

    public Integer getProductId() {
        return productId;
```

3. Product entity(thực thể này sẽ mapping đến table trong CSDL):



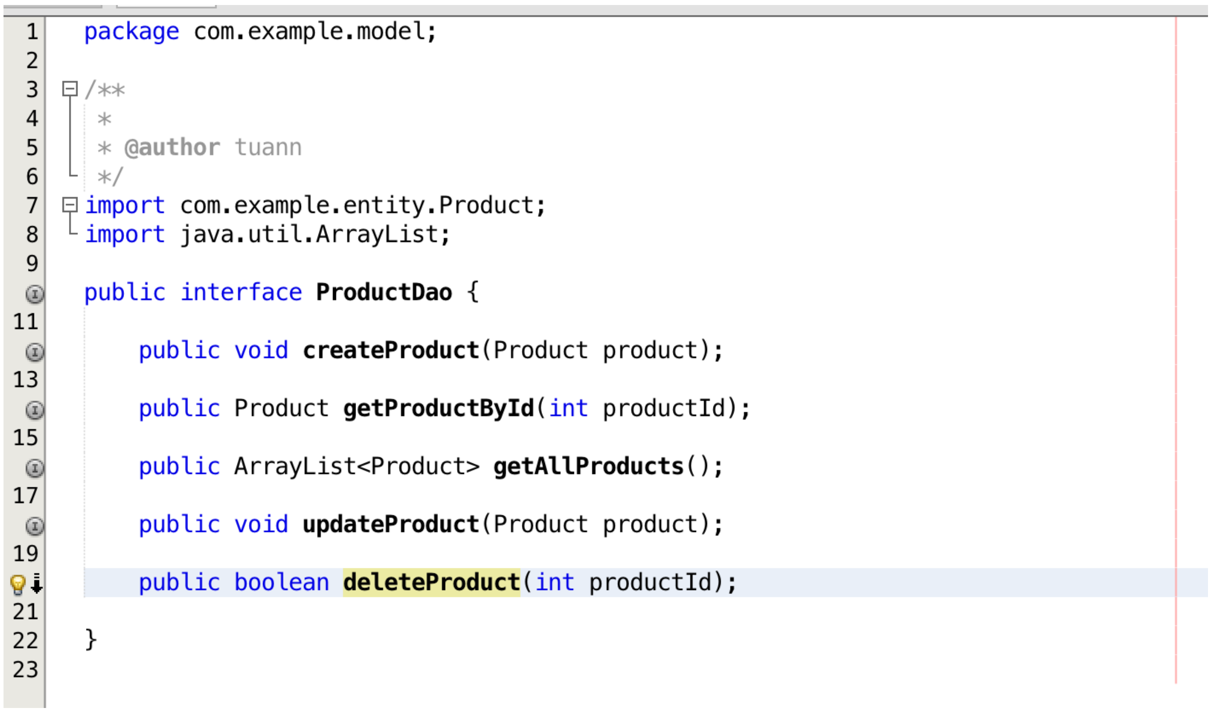
```java
package com.example.entity;

/**
 *
 * @author tuann
 */
public class Product {

    private Integer productId;
    private String productName;
    private String productDesc;
    private Double productPrice;

    public Product() {
    }

    public Product(String productName, String productDesc, Double productPrice) {
        this.productName = productName;
        this.productDesc = productDesc;
        this.productPrice = productPrice;
    }

    public Integer getProductId() {
        return productId;
    }

    public void setProductId(Integer productId) {
        this.productId = productId;
    }

    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

```

4. DBConnection class(lớp kết nối đến CSDL):



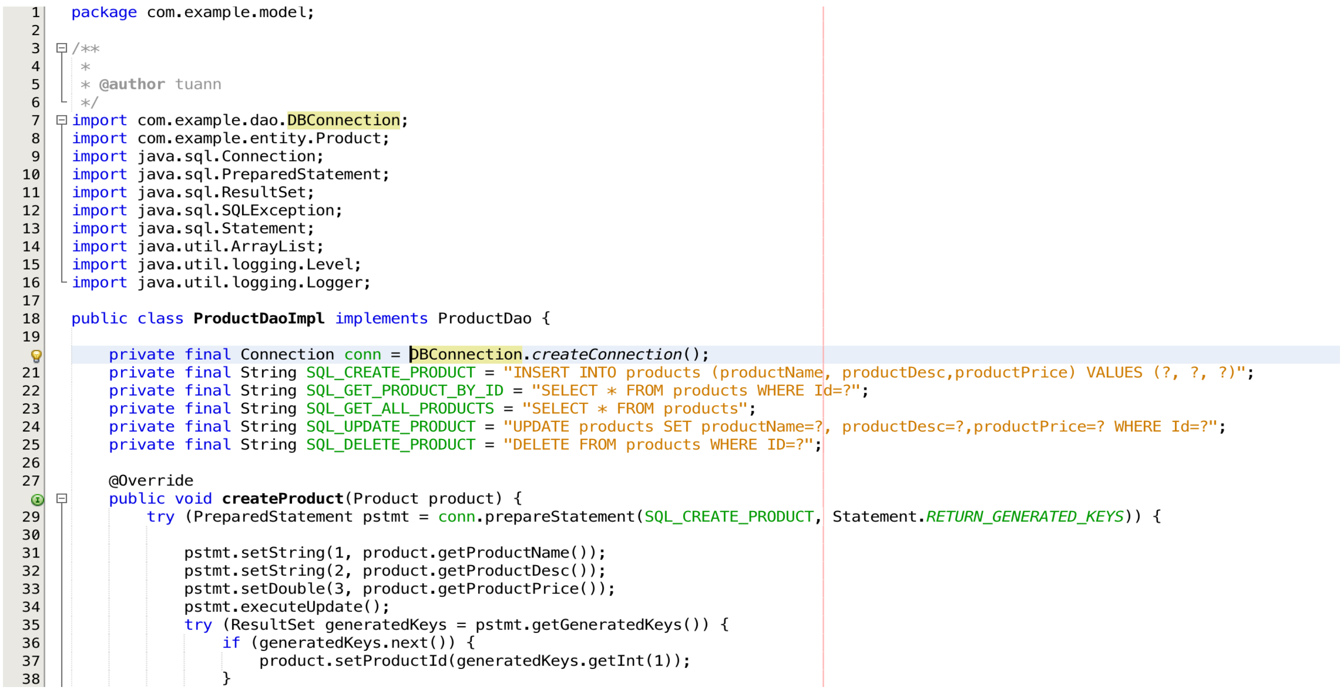
```java
package com.example.dao;

/**
 *
 * @author tuann
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class DBConnection {

    public static Connection createConnection() {
        String dbUrl = "jdbc:derby://localhost:1527/myDB";
        Connection conn = null;
        try {
            conn = DriverManager.getConnection(dbUrl, "sa", "sa");
        } catch (SQLException ex) {
            Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE, null, ex);
        }
        return conn;
    }

}
```

5.ProductDao interface(interface này mô tả các phương thức sẽ triển khai trong lớp Biz)

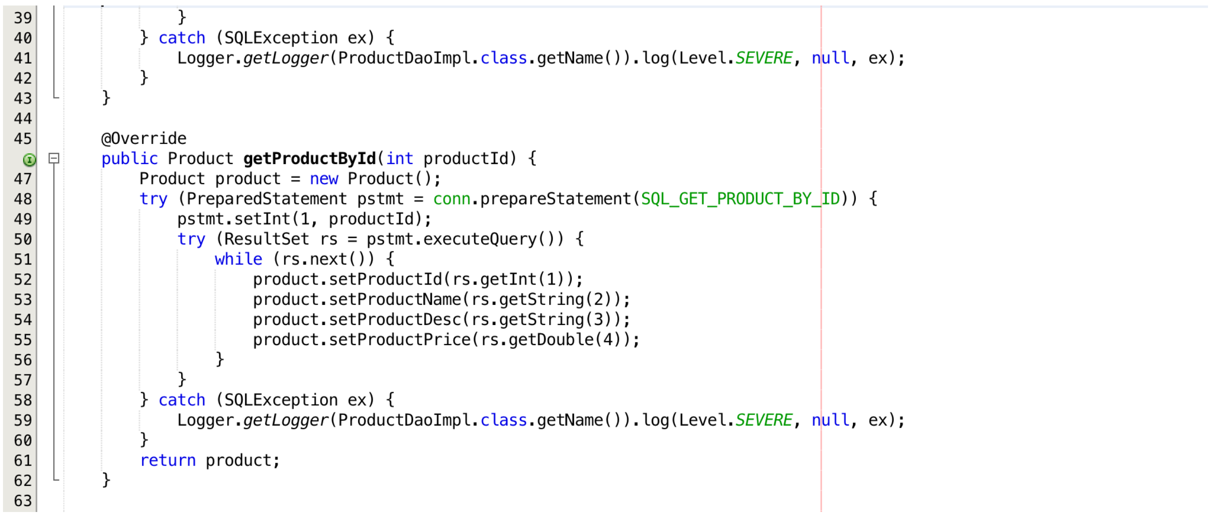


```java
package com.example.model;

/**
 *
 * @author tuann
 */
import com.example.entity.Product;
import java.util.ArrayList;

public interface ProductDao {

    public void createProduct(Product product);

    public Product getProductById(int productId);

    public ArrayList<Product> getAllProducts();

    public void updateProduct(Product product);

    public boolean deleteProduct(int productId);

}
```

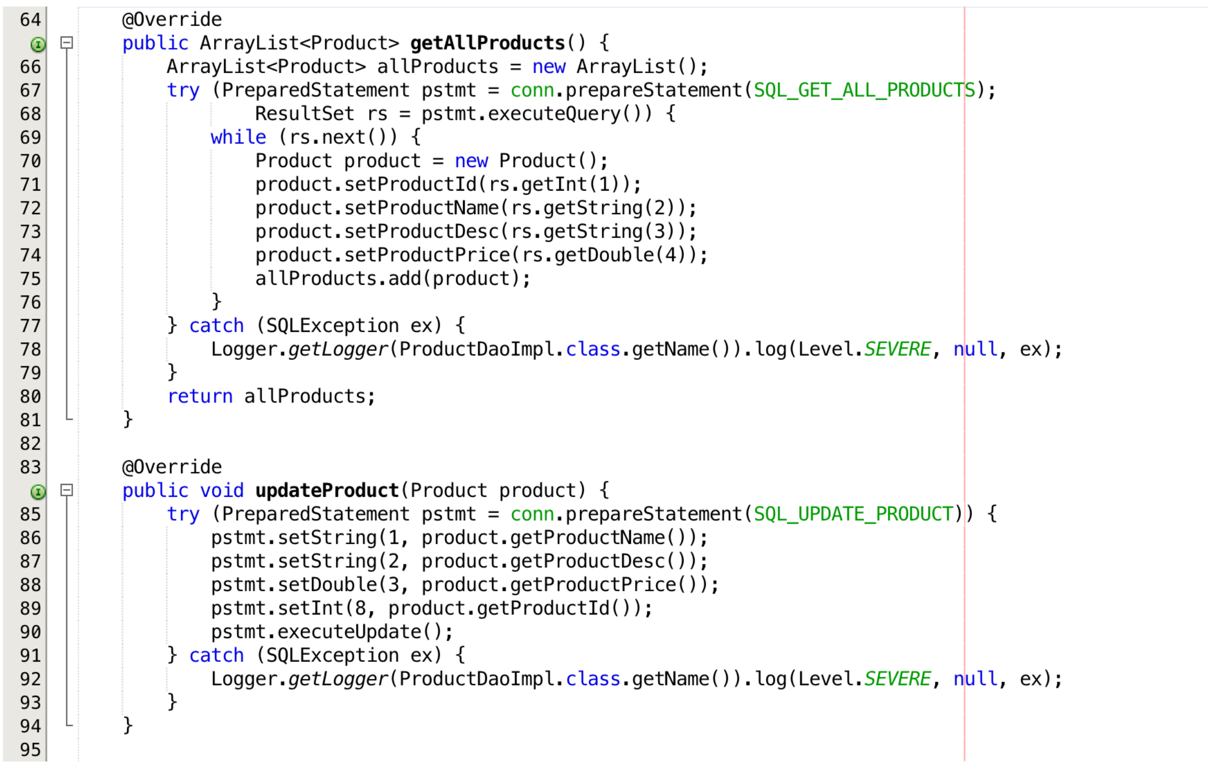
6. ProductDaoImpl class(implements các phương thức mô tả trong interface)



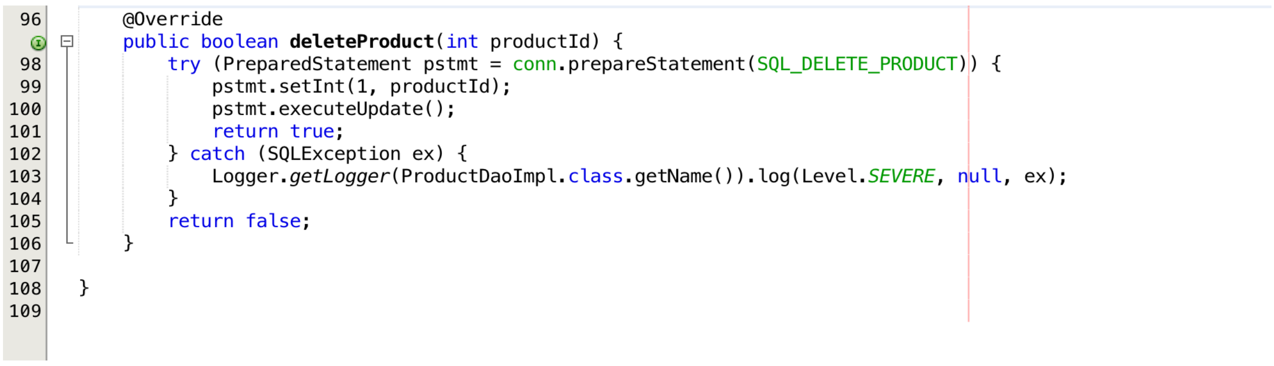
```java
package com.example.model;

/**
 *
 * @author tuann
 */
import com.example.dao.DBConnection;
import com.example.entity.Product;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ProductDaoImpl implements ProductDao {

    private final Connection conn = DBConnection.createConnection();
    private final String SQL_CREATE_PRODUCT = "INSERT INTO products (productName, productDesc,productPrice) VALUES (?, ?, ?)";
    private final String SQL_GET_PRODUCT_BY_ID = "SELECT * FROM products WHERE Id=?";
    private final String SQL_GET_ALL_PRODUCTS = "SELECT * FROM products";
    private final String SQL_UPDATE_PRODUCT = "UPDATE products SET productName=?, productDesc=?,productPrice=? WHERE Id=?";
    private final String SQL_DELETE_PRODUCT = "DELETE FROM products WHERE ID=?";

    @Override
    public void createProduct(Product product) {
        try (PreparedStatement pstmt = conn.prepareStatement(SQL_CREATE_PRODUCT, Statement.RETURN_GENERATED_KEYS)) {

            pstmt.setString(1, product.getProductName());
            pstmt.setString(2, product.getProductDesc());
            pstmt.setDouble(3, product.getProductPrice());
            pstmt.executeUpdate();
            try (ResultSet generatedKeys = pstmt.getGeneratedKeys()) {
                if (generatedKeys.next()) {
                    product.setProductId(generatedKeys.getInt(1));
                }
```



```java
            }
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    public Product getProductById(int productId) {
        Product product = new Product();
        try (PreparedStatement pstmt = conn.prepareStatement(SQL_GET_PRODUCT_BY_ID)) {
            pstmt.setInt(1, productId);
            try (ResultSet rs = pstmt.executeQuery()) {
                while (rs.next()) {
                    product.setProductId(rs.getInt(1));
                    product.setProductName(rs.getString(2));
                    product.setProductDesc(rs.getString(3));
                    product.setProductPrice(rs.getDouble(4));
                }
            }
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return product;
    }

```

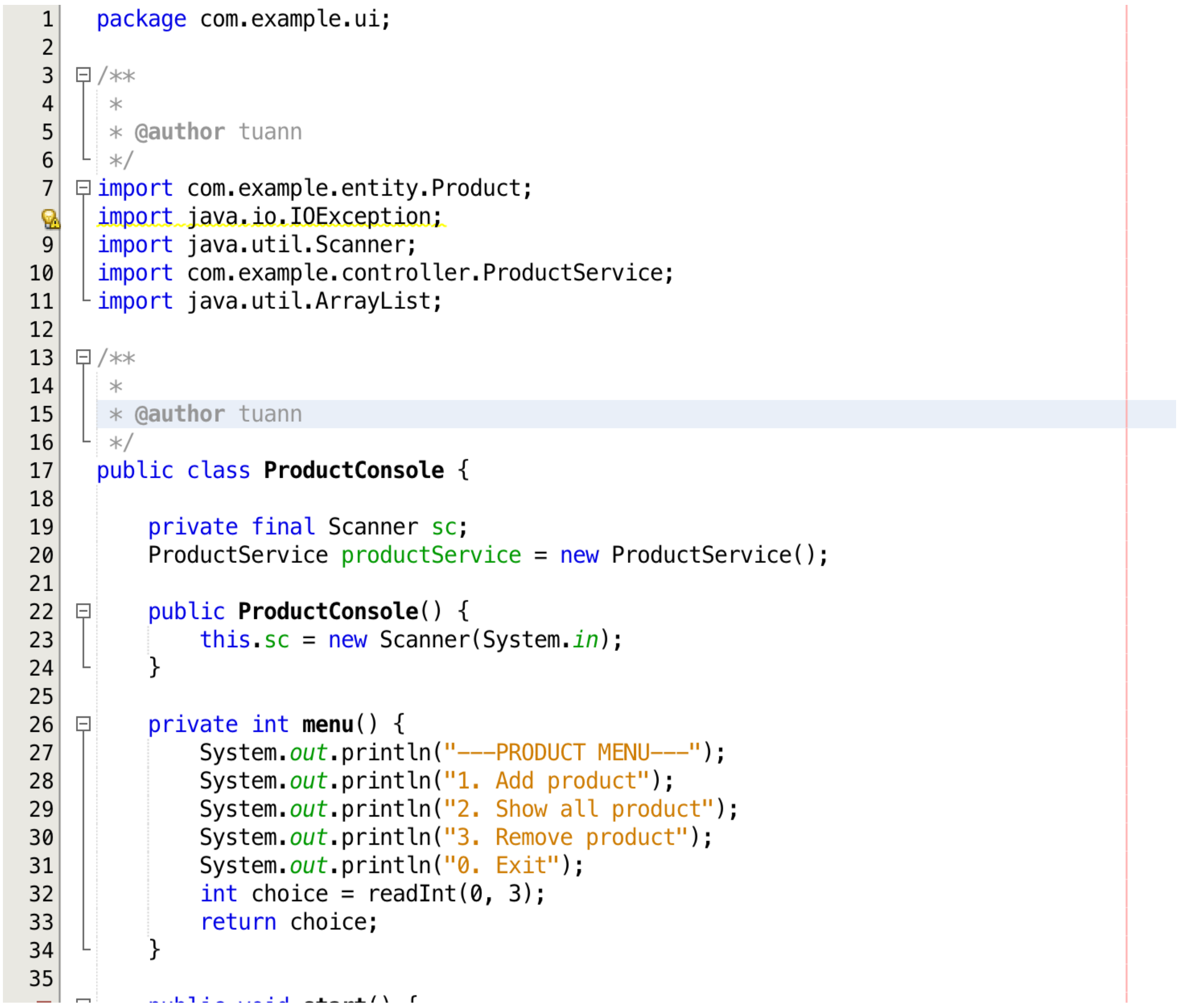


```java
    @Override
    public ArrayList<Product> getAllProducts() {
        ArrayList<Product> allProducts = new ArrayList();
        try (PreparedStatement pstmt = conn.prepareStatement(SQL_GET_ALL_PRODUCTS);
                ResultSet rs = pstmt.executeQuery()) {
            while (rs.next()) {
                Product product = new Product();
                product.setProductId(rs.getInt(1));
                product.setProductName(rs.getString(2));
                product.setProductDesc(rs.getString(3));
                product.setProductPrice(rs.getDouble(4));
                allProducts.add(product);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return allProducts;
    }

    @Override
    public void updateProduct(Product product) {
        try (PreparedStatement pstmt = conn.prepareStatement(SQL_UPDATE_PRODUCT)) {
            pstmt.setString(1, product.getProductName());
            pstmt.setString(2, product.getProductDesc());
            pstmt.setDouble(3, product.getProductPrice());
            pstmt.setInt(8, product.getProductId());
            pstmt.executeUpdate();
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

```

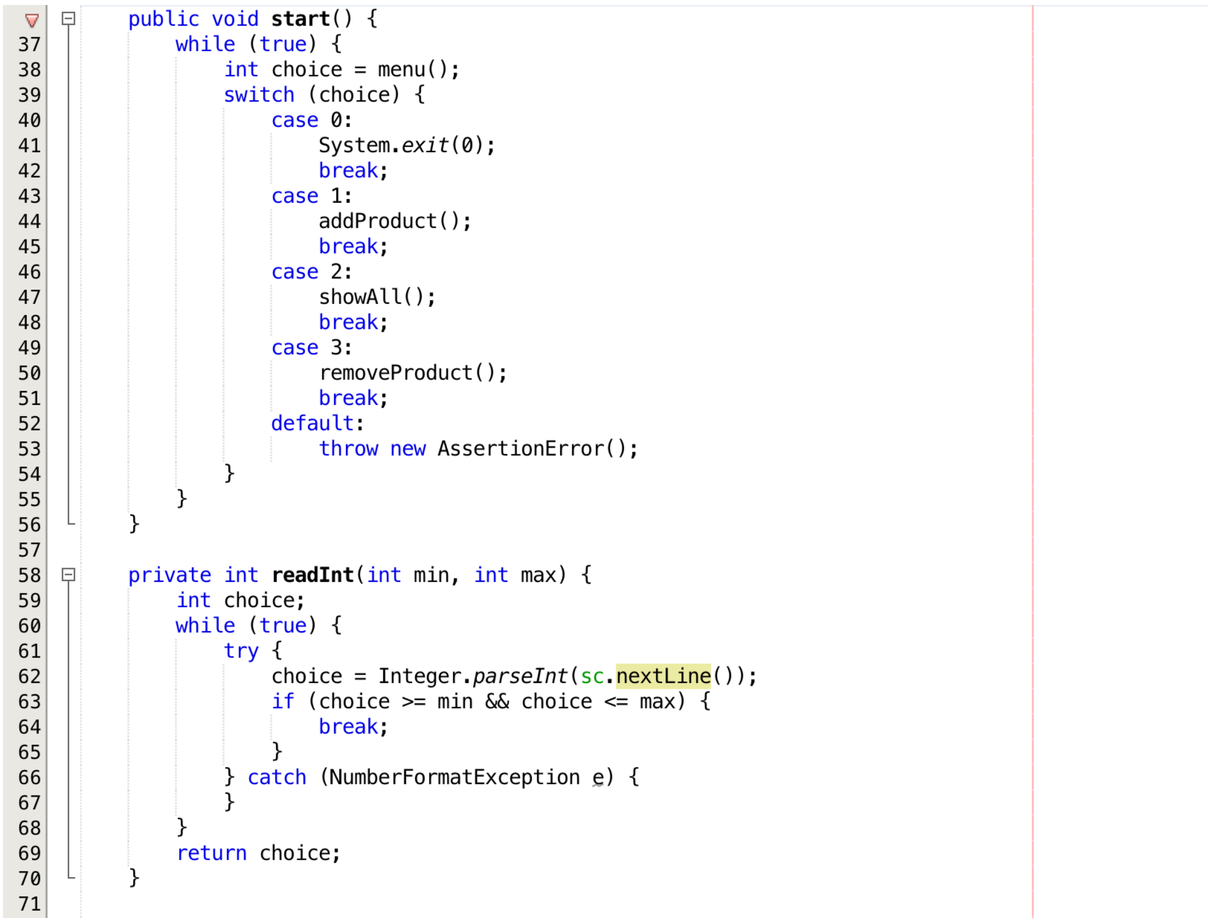


```java
    @Override
    public boolean deleteProduct(int productId) {
        try (PreparedStatement pstmt = conn.prepareStatement(SQL_DELETE_PRODUCT)) {
            pstmt.setInt(1, productId);
            pstmt.executeUpdate();
            return true;
        } catch (SQLException ex) {
            Logger.getLogger(ProductDaoImpl.class.getName()).log(Level.SEVERE, null, ex);
        }
        return false;
    }

}

```

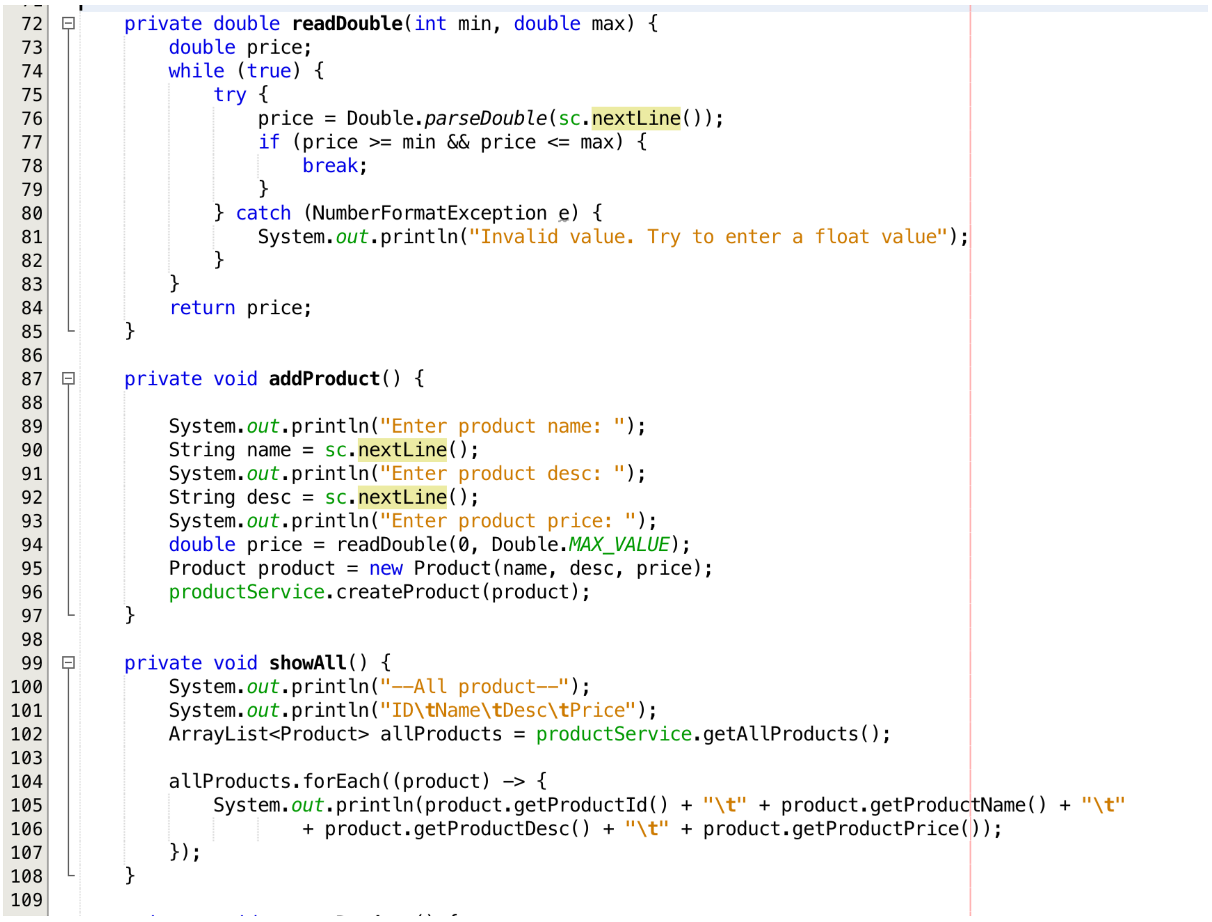
7. ProductConsole class(Lớp “view” đóng vai trò tương tác với người dùng)



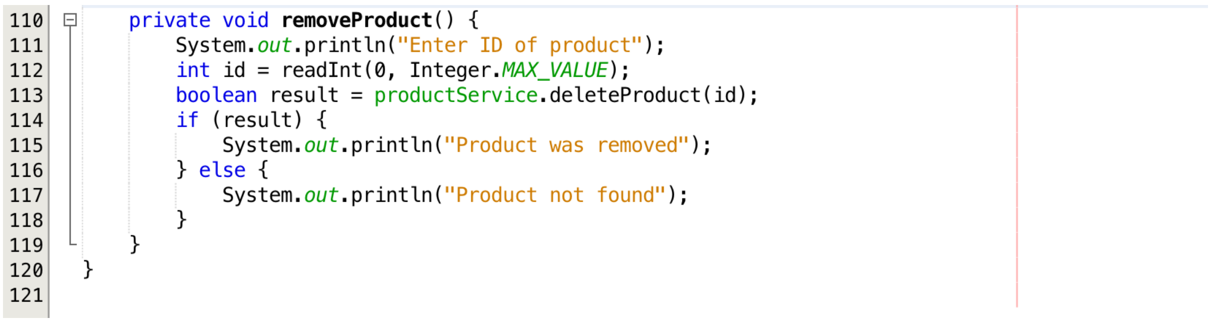
```java
package com.example.ui;

/**
 *
 * @author tuann
 */
import com.example.entity.Product;
import java.io.IOException;
import java.util.Scanner;
import com.example.controller.ProductService;
import java.util.ArrayList;

/**
 *
 * @author tuann
 */
public class ProductConsole {

    private final Scanner sc;
    ProductService productService = new ProductService();

    public ProductConsole() {
        this.sc = new Scanner(System.in);
    }

    private int menu() {
        System.out.println("---PRODUCT MENU---");
        System.out.println("1. Add product");
        System.out.println("2. Show all product");
        System.out.println("3. Remove product");
        System.out.println("0. Exit");
        int choice = readInt(0, 3);
        return choice;
    }

    public void start() {
```



```java
    public void start() {
        while (true) {
            int choice = menu();
            switch (choice) {
                case 0:
                    System.exit(0);
                    break;
                case 1:
                    addProduct();
                    break;
                case 2:
                    showAll();
                    break;
                case 3:
                    removeProduct();
                    break;
                default:
                    throw new AssertionError();
            }
        }
    }

    private int readInt(int min, int max) {
        int choice;
        while (true) {
            try {
                choice = Integer.parseInt(sc.nextLine());
                if (choice >= min && choice <= max) {
                    break;
                }
            } catch (NumberFormatException e) {
            }
        }
        return choice;
    }

```

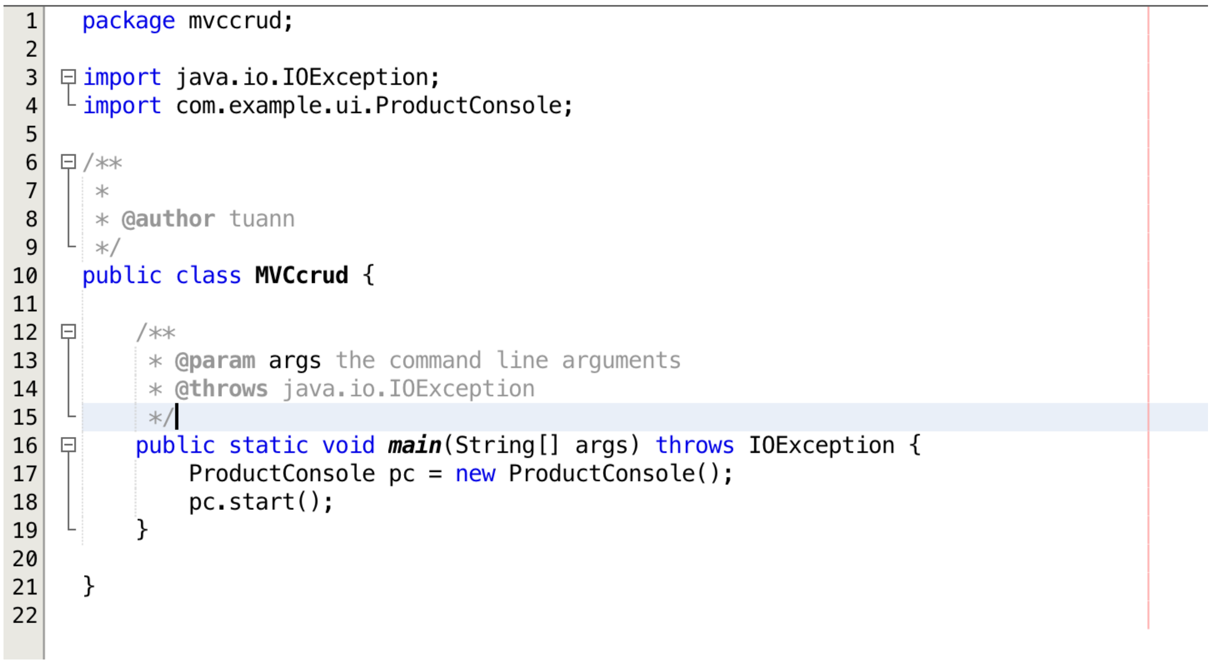


```java
    private double readDouble(int min, double max) {
        double price;
        while (true) {
            try {
                price = Double.parseDouble(sc.nextLine());
                if (price >= min && price <= max) {
                    break;
                }
            } catch (NumberFormatException e) {
                System.out.println("Invalid value. Try to enter a float value");
            }
        }
        return price;
    }

    private void addProduct() {

        System.out.println("Enter product name: ");
        String name = sc.nextLine();
        System.out.println("Enter product desc: ");
        String desc = sc.nextLine();
        System.out.println("Enter product price: ");
        double price = readDouble(0, Double.MAX_VALUE);
        Product product = new Product(name, desc, price);
        productService.createProduct(product);
    }

    private void showAll() {
        System.out.println("--All product--");
        System.out.println("ID\tName\tDesc\tPrice");
        ArrayList<Product> allProducts = productService.getAllProducts();

        allProducts.forEach((product) -> {
            System.out.println(product.getProductId() + "\t" + product.getProductName() + "\t"
                    + product.getProductDesc() + "\t" + product.getProductPrice());
        });
    }

```



```java
    private void removeProduct() {
        System.out.println("Enter ID of product");
        int id = readInt(0, Integer.MAX_VALUE);
        boolean result = productService.deleteProduct(id);
        if (result) {
            System.out.println("Product was removed");
        } else {
            System.out.println("Product not found");
        }
    }
}

```

8. MVCcrud -> run application



```java
package mvccrud;

import java.io.IOException;
import com.example.ui.ProductConsole;

/**
 *
 * @author tuann
 */
public class MVCcrud {

    /**
     * @param args the command line arguments
     * @throws java.io.IOException
     */
    public static void main(String[] args) throws IOException {
        ProductConsole pc = new ProductConsole();
        pc.start();
    }

}

```