상품관리 프로그램을 TCP를 이용하여 개발하려고 한다.

아래 조건에 맞게 서버와 클라이언트를 직접 개발하시오.

[실행내용]

1. 클라이언트를 실행하면 콘솔에서 다음과 같이 [상품 목록]과 메뉴가 출력된다.

```
[상품 목록]
---------------------------------------------------------------------
no    name                              price          stock
---------------------------------------------------------------------

---------------------------------------------------------------------
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택:
```

2. 선택에서 1을 입력하고 Enter 키를 누르면 다음과 같이 상품 생성을 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택: 1
[상품 생성]
상품 이름: Television
상품 가격: 3000000
상품 재고: 10
```

3. 상품 이름, 상품 가격, 상품 재고까지 모두 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다.

```
[상품 목록]
---------------------------------------------------------------------
no    name                              price          stock
---------------------------------------------------------------------
1     Television                          3000000       10

---------------------------------------------------------------------
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택:
```

4. 선택에서 2를 입력하고 Enter 키를 누르면 다음과 같이 상품 수정을 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택: 2

[상품 수정]
상품 번호: 1
이름 변경: SmartTV
가격 변경: 3500000
재고 변경: 20
```

5. 상품 번호, 이름 변경, 가격 변경, 재고 변경까지 모두 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다.

```
[상품 목록]
-------------------------------------------------------------------
no    name                          price          stock
-------------------------------------------------------------------
1     SmartTV                       3500000        20

-------------------------------------------------------------------
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택:
```

6. 선택에서 3을 입력하고 Enter 키를 누르면 다음과 같이 상품 삭제를 위한 정보를 입력할 수 있다.

```
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택: 3

[상품 삭제]
상품 번호: 1
```

7. 삭제할 상품 번호를 입력하고 Enter 키를 누르면 다시 [상품 목록]으로 되돌아간다. 목록에서는 삭제된 상품이 보이지 않아야 한다.

```
[상품 목록]
-------------------------------------------------------------------
no    name                          price          stock
-------------------------------------------------------------------
-------------------------------------------------------------------
메뉴: 1.Create | 2.Update | 3.Delete | 4.Exit
선택:
```

8. 마지막으로 선택에서 4를 입력하고 Enter 키를 누르면 클라이언트 프로그램이 종료된다.

[요약]

클라이언트 요청부터 서버 응답까지의 전 과정을 요약하면 다음과 같다.

❶ 선택에서 메뉴 번호를 입력한다.

❷ 필요한 정보를 키보드로 추가 입력받는다.

❸ 메뉴 번호와 입력된 데이터를 JSON 형식으로 만들고, 서버로 처리 요청을 한다.

❹ 서버는 요청 JSON을 해석하고 처리한다(동시 요청 처리를 위한 스레드풀 적용).

❺ 서버는 처리 결과를 JSON 형식으로 만들고 클라이언트로 응답을 보낸다.

❻ 클라이언트는 응답 JSON을 해석하고 status가 success일 경우 다시 목록과 메뉴를 보여준다.

[실행 조건]

1. 클라이언트가 서버로 보내는 요청 JSON은 다음과 같은 구조로 작성한다. menu는 입력된 메뉴 번호이며, data는 서버에서 요청을 처리하는 데 필요한 데이터이다.

```
{
  "menu": 메뉴 번호,
  "data": { … }
}
```

2. 서버가 클라이언트로 보내는 응답 JSON은 다음과 같은 구조로 작성한다. status는 처리 상태이므로 "success" 또는 "fail"로 작성하고, data는 클라이언트로 전달하고자 하는 데이터를 넣는다. 상품 목록을 보낼 경우 data는 배열 형태가 된다.

```
{
  "status": "success" 또는 "fail",
  "data": { … } 또는 [ … ]
}
```

3. 상품 정보를 담을 때 사용하는 Product 클래스는 다음과 같이 설계한다. 그리고 서버는 상품을 List〈Product〉 컬렉션으로 메모리에 저장해서 관리한다.

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Product {
  private int no;
  private String name;
  private int price;
  private int stock;
}
```

[작성할 파일]

이 문제를 위해 작성해야 할 소스 파일은 다음 3가지이다.

❶ 서버: ProductServer.java (ProductServer 클래스와 SocketClient 중첩 클래스 선언)

❷ 클라이언트: ProductClient.java (ProductClient 클래스 선언)

❸ 공통: Product.java (Product 클래스 선언)