

실전 객체지향 프로그래밍

02. 쇼핑몰 프로그램 개발

■ 프로젝트의 개요 : 주요 기능

- **상품 등록** : 상품은 사전에 등록 절차를 거쳐 관리자가 등록해야 한다. 등록 메서드로 일괄 등록할 수 있게 구현한다. 등록된 상품은 데이터베이스와 같이 데이터를 저장하고 공유할 수 있는 저장소가 필요하다. 배열을 이용하여 저장한다.
- **사용자 등록 및 로그인** : 개인별로 사용자를 등록하고 로그인한 후 쇼핑몰을 이용할 수 있다. 사용자 등록 메서드로 일괄 등록하는 것으로 구현한다. 사용자 정보 역시 상품과 마찬가지로 배열에 저장한다. 로그인도 별도의 비밀번호 없이 사용자 목록 화면에서 사용자를 선택하여 자동 로그인하는 것으로 구현한다.
- **상품 목록 및 정보 출력** : 쇼핑몰의 메인 화면이 된다. 등록된 상품의 세부 정보를 나열하고 장바구니로 선택할 수 있게 한다.
- **장바구니** : 상품 목록 화면에서 선택된 상품들을 임시 보관하는 기능으로, `java.util.ArrayList` 클래스를 사용하여 관리한다
- **계산** : 장바구니에 등록된 상품들을 일괄 계산하는 기능으로, 로그인하면 선택한 사용자의 결제 정보를 함께 보여 준다. 이전 화면으로 돌아가 상품을 추가하는 것도 가능하다. 실제 결제 처리 과정은 없으며, 결제되었다는 메시지를 출력한 후 프로그램을 종료한다.

02. 쇼핑몰 프로그램 개발

■ 프로젝트의 개요 : 주요 기술 요소

표 5-1 프로젝트에 적용된 주요 기술

번호	사용 기술	설명
1	자바 기본 문법	변수 선언, 출력문, for 문, switch 문, 배열
2	자바 라이브러리	ArrayList, Scanner, Enum, Integer
3	클래스 기본 구조	클래스 선언, 생성자, 객체 생성, 인스턴스 변수, 클래스 변수, 멤버 변수, 메서드
4	객체지향 개념	추상 클래스, 인터페이스, 상속, 메서드 오버라이딩

02. 쇼핑몰 프로그램 개발

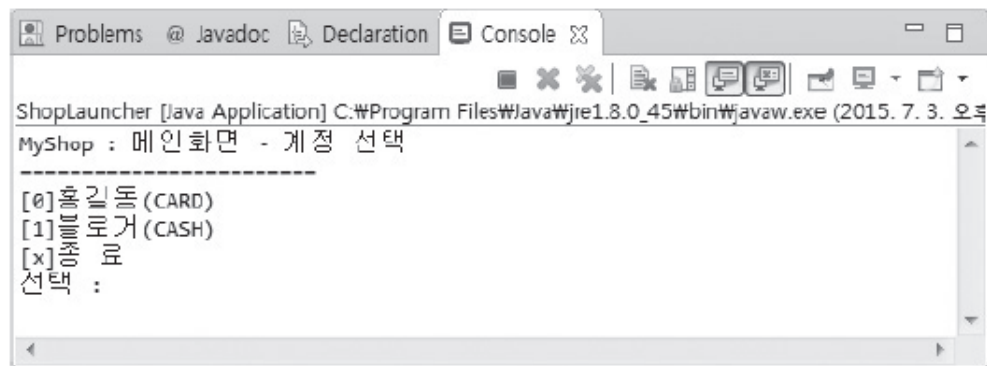
■ 프로젝트의 개요 : 특이 사항

- 콘솔 화면에 텍스트로 구현된 화면을 사용.
- 데이터베이스에 정보를 저장하지 않기 때문에 실행 내용이 저장되지 않고, 프로그램을 실행할 때마다 기본값으로 설정.

02. 쇼핑몰 프로그램 개발

■ 프로젝트의 개요 : 실행 결과 화면

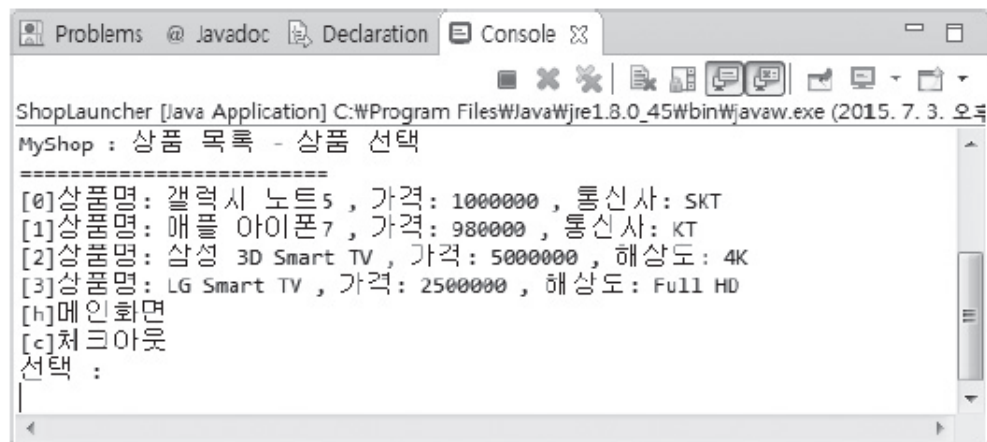
그림 5-2 사용자 선택



```
ShopLauncher [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (2015. 7. 3. 오후 1:14:14)
MyShop : 메인화면 - 계정 선택
-----
[0]홍길동 (CARD)
[1]블로거 (CASH)
[x]종료
선택 :
```

사용자 이름 앞에 표시된 번호를 입력하고 Enter 를 누르면 상품 목록을 출력

그림 5-3 상품 목록 출력



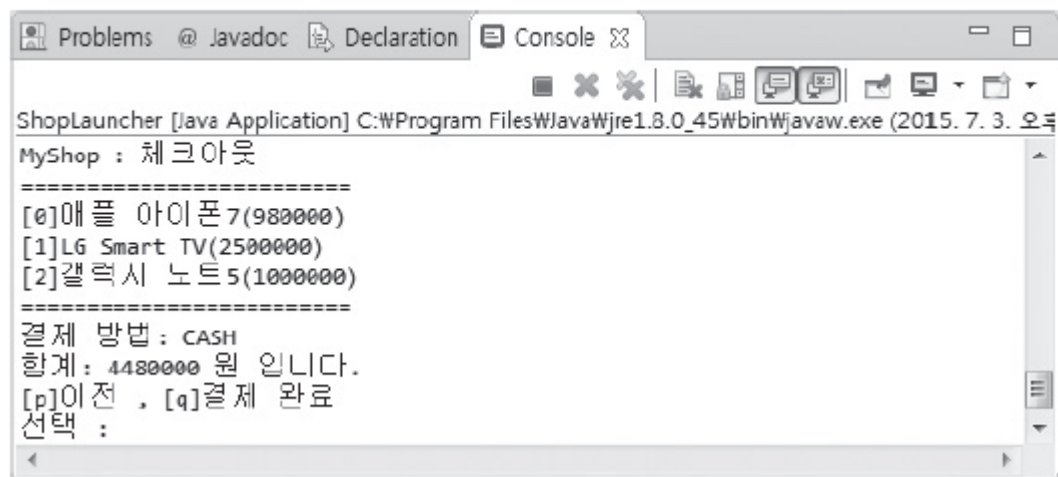
```
ShopLauncher [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (2015. 7. 3. 오후 1:14:14)
MyShop : 상품 목록 - 상품 선택
-----
[0]상품명: 갤럭시 노트5 , 가격: 1000000 , 통신사: SKT
[1]상품명: 애플 아이폰7 , 가격: 980000 , 통신사: KT
[2]상품명: 삼성 3D Smart TV , 가격: 5000000 , 해상도: 4K
[3]상품명: LG Smart TV , 가격: 2500000 , 해상도: Full HD
[h]메인화면
[c]처크아웃
선택 :
```

02. 쇼핑몰 프로그램 개발

■ 프로젝트의 개요 : 실행 결과 화면

체크아웃을 선택하면 지금까지 선택한 모든 상품의 정보가 결제 정보와 함께 출력

그림 5-4 체크아웃



The screenshot shows a Java application window titled 'ShopLauncher [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (2015. 7. 3. 오후 1:00)'. The console output displays the checkout process for 'MyShop'.

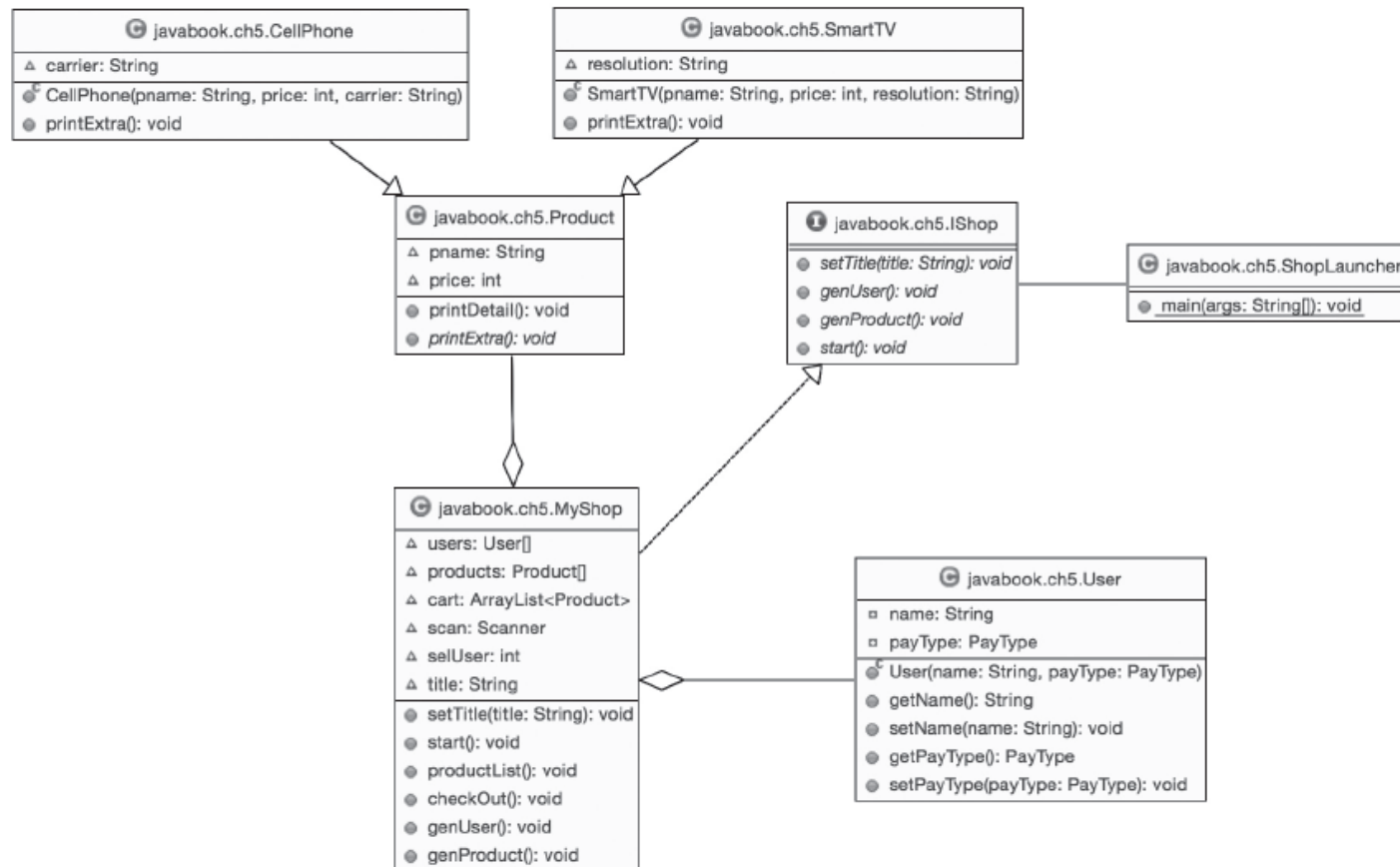
```
ShopLauncher [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (2015. 7. 3. 오후 1:00)
MyShop : 체크아웃
=====
[0]애플 아이폰7(980000)
[1]LG Smart TV(2500000)
[2]갤럭시 노트5(1000000)
=====
결제 방법 : CASH
합계 : 4480000 원 입니다.
[p]이전 , [q]결제 완료
선택 :
```

02. 쇼핑몰 프로그램 개발

■ 클래스 구조 설계

- 클래스 간 세모 화살표 : 상속 관계
- 다이아몬드 화살표 : 집합 관계

그림 5-5 클래스 다이어그램



02. 쇼핑몰 프로그램 개발

■ 클래스 구조 설계

표 5-2 쇼핑몰 프로그램의 클래스 목록

번호	클래스	설명
1	IShop	쇼핑몰의 기본 구조를 정의한 인터페이스
2	Product	상품의 기본 구조와 정보 출력 기능이 있는 추상 클래스
3	MyShop	쇼핑몰의 메인 클래스로, IShop 인터페이스를 구현하는 여러 쇼핑몰 클래스가 있다.
4	CellPhone	Product 추상 클래스를 구현한 휴대폰 클래스
5	SmartTV	Product 추상 클래스를 구현한 스마트TV 클래스
6	User	쇼핑몰 사용자 정의를 위한 클래스
7	ShopLauncher	쇼핑몰 프로그램의 실행을 위한 런처 프로그램

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : IShop 인터페이스

- IShop 인터페이스는 쇼핑몰의 기본 구조를 정의한다.

표 5-3 IShop 인터페이스 구성

번호	메서드	설명
1	setTitle(String title)	쇼핑몰 제목(이름)을 설정한다.
2	genUser()	쇼핑몰을 이용할 수 있는 예제 사용자들을 생성한다.
3	genProduct()	쇼핑몰의 예제 상품을 생성한다.
4	start()	쇼핑몰 프로그램을 실행하는 시작 메서드이다.

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : IShop 인터페이스

예제 5-8 IShop 인터페이스 구현하기

IShop.java

```
01 package javabook.ch5;
02
03 // 쇼핑몰 규격을 정의한 인터페이스, 모든 쇼핑몰은 IShop 인터페이스를 구현해야 한다.
04 public interface IShop {
05     public abstract void setTitle(String title);
06     public abstract void genUser();
07     public abstract void genProduct();
08     public abstract void start();
09 }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : Product 추상 클래스

- 상품에 대한 기본적인 구조와 기능을 담고 있다.
- 모든 상품이 가져야 할 공통 내용만 정의하고, 각 상품별 구체적인 내용은 해당 클래스에서 확장한다.
- 멤버 변수로는 상품 이름 지정에 필요한 String pname과 가격을 저장하는 int price가 있다.

표 5-4 Product 추상 클래스 구성

번호	메서드	설명
1	printDetail()	상품의 기본 정보인 상품 이름과 가격을 출력한다. 상세 정보를 출력하려고 printExtra() 메서드를 호출한다.
2	printExtra()	추상 메서드로, 실제 구현은 서브 클래스에서 해야 한다. printDetail()에서 자동으로 호출하므로, printExtra()에 구현된 내용도 자동으로 함께 출력된다.

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : Product 추상 클래스

예제 5-9 Product 추상 클래스 생성하기

Product.java

```
01 package javabook.ch5;
02
03 // 쇼핑몰 상품 정의를 위한 추상 클래스,
04 // 다양한 상품 종류를 지원하려고 기본 정보 설정과 출력만 추상 클래스에서 지정하고
05 // 추가 정보는 서브 클래스에서 구현하도록 유도한다.
06 public abstract class Product {
07     // 상품 이름과 가격 변수
08     String pname;
09     int price;
10
11     // 상품 이름과 가격을 출력하는 기본 정보 출력 메서드
```

02. 쇼핑몰 프로그램 개발

```
12     public void printDetail() {  
13         System.out.print("상품명: " + pname + " , ");  
14         System.out.print("가격: " + price + " , ");  
15         printExtra();  
16     }  
17  
18     public abstract void printExtra();  
19 }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : User 클래스

- User 클래스는 쇼핑몰 사용자의 기본적인 정보를 설정하고 제공하는 기능을 한다.

표 5-5 User 클래스 구성

번호	메서드	설명
1	User(String name, PayType payType)	생성자로 이름과 결제 정보를 받아 설정한다.
2	String getName()	사용자의 이름 정보를 리턴한다.
3	void setName(String name)	사용자의 이름 정보를 설정한다.
4	PayType getPayType()	사용자의 결제 정보를 리턴한다.
5	void setPayType(PayType payType)	사용자의 결제 정보를 설정한다.

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : User 클래스

예제 5-10 User 클래스 구현하기

User.java

```
01 package javabook.ch5;
02
03 public class User {
04     // 사용자 이름
05     private String name;
06
07     // 결제 유형 - enum
08     private PayType payType;
09
10     // 이름과 결제 정보를 파라미터로 가지는 생성자
11     public User(String name, PayType payType) {
12         this.name = name;
13         this.payType = payType;
14     }
```

02. 쇼핑몰 프로그램 개발

```
15
16     public String getName() {
17         return name;
18     }
19
20     public void setName(String name) {
21         this.name = name;
22     }
23
24     public PayType getPayType() {
25         return payType;
26     }
27
28     public void setPayType(PayType payType) {
29         this.payType = payType;
30     }
31 }
```


02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : User 클래스

▶ PayType enum 클래스

```
if(user.getPayType.equals("현금"))
```

[문제점]

- 사용자의 결제 유형이 무엇인지 알려면 equals를 써야 하는 불편함
- 문자열의 철자가 잘못되어도 컴파일 오류가 발생하지 않기 때문에 프로그램 실행 과정에서 런타임 오류가 발생
- 다국어 처리에도 어려움
- 코드에서 '현금' 문자열을 분산해서 사용하므로 나중에 이를 'CASH' 등으로 변경

[해결방법 : enum 타입 사용]

- enum은 상수와 같이 여러 데이터 유형을 그룹으로 지정하여 사용 가능
- 이름 또는 인덱스값으로 타입을 병행하여 사용할 수도 있어 다양한 프로그램 환경에 쉽게 대응 가능

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰과 상품, 사용자의 구조화 : User 클래스

예제 5-11 결제 정보 타입 지정하기

PayType.java

```
01 package javabook.ch5;
02
03 // 결제 유형을 지정하는 enum 타입으로 CASH(현금), CARD(카드) 두 가지를 지원한다.
04 public enum PayType {CASH, CARD}
```

02. 쇼핑몰 프로그램 개발

■ 상품 클래스 구체화 : CellPhone 클래스

- CellPhone 클래스는 Product 클래스를 상속받는 클래스로, 휴대폰의 제품 정보를 등록하는 데 필요.
- 기본 Product 클래스 정보 외에 휴대폰 정보만 추가하여 출력하려고 printExtra() 메서드를 오버라이딩해서 구현.

표 5-6 CellPhone 클래스 구성

번호	메서드	설명
1	CellPhone(String pname, int price, String carrier)	생성자로, 상품 이름과 가격, 통신사 정보를 받아서 처리한다.
2	void printExtra()	인터페이스에 정의된 추상 메서드를 오버라이딩한다. 추가 정보인 통신사 정보를 출력한다.

02. 쇼핑몰 프로그램 개발

■ 상품 클래스 구체화 : CellPhone 클래스

예제 5-12 Product 클래스를 상속받는 CellPhone 클래스 구현하기

CellPhone.java

```
01 package javabook.ch5;
02
03 // Product 추상 클래스를 상속받아 구현한 CellPhone 클래스
04 public class CellPhone extends Product {
05     String carrier;    // 통신사
06
07     public CellPhone(String pname, int price, String carrier) {
08         this.pname = pname;
09         this.price = price;
10         this.carrier = carrier;
11     }
12
13     @Override
14     public void printExtra() {
15         System.out.println("통신사: " + carrier);
16     }
17 }
```

02. 쇼핑몰 프로그램 개발

■ 상품 클래스 구체화 : SmartTV 클래스

- Product 클래스를 상속받는 클래스로, 스마트TV의 제품 정보를 등록하는 데 필요
- 기본 Product 클래스 정보 외에 스마트TV정보만 추가하여 출력하려고 printExtra() 메서드를 오버라이딩해서 구현

표 5-7 SmartTV 클래스 구성

번호	메서드	설명
1	SmartTV(String pname, int price, String resolution)	생성자로 상품 이름과 가격, 해상도 정보를 받아 처리한다.
2	void printExtra()	인터페이스에 정의된 추상 메서드를 오버라이딩한다. 추가 정보인 해상도 정보를 출력한다.

02. 쇼핑몰 프로그램 개발

■ 상품 클래스 구체화 : SmartTV 클래스

예제 5-13 Product 클래스를 상속받는 SmartTV 클래스 구현하기

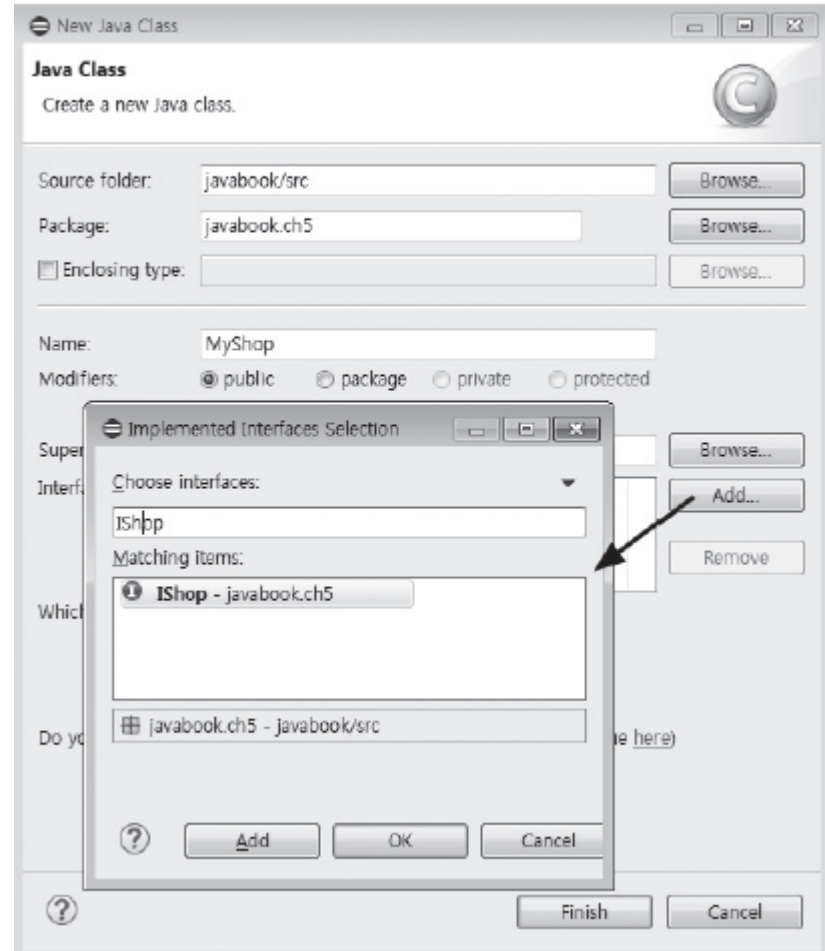
SmartTV.java

```
01 package javabook.ch5;
02
03 // Product 추상 클래스를 상속받아 구현한 SmartTV 클래스
04 public class SmartTV extends Product {
05     String resolution;    // 해상도
06
07     public SmartTV(String pname, int price, String resolution) {
08         this.pname = pname;
09         this.price = price;
10         this.resolution = resolution;
11     }
12
13     @Override
14     public void printExtra() {
15         System.out.println("해상도: " + resolution);
16     }
17 }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현

그림 5-6 인터페이스 선택



02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현

표 5-8 MyShop 클래스 구성

번호	메서드	설명
1	setTitle(String title)	쇼핑몰 제목(이름)을 설정한다.
2	genUser()	쇼핑몰을 이용할 수 있는 예제 사용자들을 생성한다.
3	genProduct()	쇼핑몰의 예제 상품을 생성한다.
4	start()	쇼핑몰 프로그램을 실행하는 메인 시작 메서드이다.
5	void productList()	상품 목록을 보여 주고 장바구니에 담을 수 있도록 하는 메서드이다.
6	void checkOut()	결제 진행을 위한 메서드로, 장바구니에 선택된 상품과 결제 금액, 결제 수단을 보여 주고 처리하는 메서드이다.

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : 클래스 선언부 구현

예제 5-14 IShop 인터페이스를 구현한 쇼핑몰 클래스

MyShop.java

```
01 // IShop 인터페이스를 구현한 MyShop 클래스
02 public class MyShop implements IShop {
03     // 등록 회원 정보 배열
04     User[] users = new User[2];
05
06     // 등록 상품 정보 배열
07     Product[] products = new Product[4];
08
09     // 상품을 추가할 수 있는 장바구니
10     ArrayList<Product> cart = new ArrayList<Product>();
11
12     // 키보드 입력으로 문자열을 입력받는 Scanner 객체 생성
13     Scanner scan = new Scanner(System.in);
14
15     // 선택한 사용자의 index 보관
16     int selUser;
17
18     // 쇼핑몰 이름
19     String title;
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : genUser() 메서드 구현

예제 5-15 사용자 정보를 생성하는 genUser() 메서드

MyShop.java

```
01      public void genUser() {  
02          User user = new User("홍길동", PayType.CARD);  
03          users[0] = user;  
04          user = new User("블로거", PayType.CASH);  
05          users[1] = user;  
06      }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : genProduct() 메서드 구현

예제 5-16 상품 정보를 생성하는 genProduct() 메서드

MyShop.java

```
01      public void genProduct() {  
02          CellPhone cp = new CellPhone("갤럭시 노트5", 1000000, "SKT");  
03          products[0] = cp;  
04          cp = new CellPhone("애플 아이폰7", 980000, "KT");  
05          products[1] = cp;  
06          SmartTV st = new SmartTV("삼성 3D Smart TV", 5000000, "4K");  
07          products[2] = st;  
08          st = new SmartTV("LG Smart TV", 2500000, "Full HD");  
09          products[3] = st;  
10      }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : start() 메서드 구현

예제 5-17 쇼핑몰 프로그램의 메인 기능인 start() 메서드

MyShop.java

```
01      public void start() {
02          System.out.println(title + " : 메인 화면 - 계정 선택");
03          System.out.println("=====");
04          int i = 0;
05
06          // 등록된 사용자 정보 출력
07          for(User u : users) {
08              System.out.printf("[%d]%(%)\n", i++, u.getName(), u.getPayType());
09          }
10
11          System.out.println("[x]종 료");
12          System.out.print("선택 : ");
13          String sel = scan.next();
14          System.out.println("## " + sel + "선택 ##");
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : start() 메서드 구현

```
15
16     // 선택한 메뉴에 따라 처리
17     switch(sel) {
18         case "x": System.exit(0);break;
19         default:
20             selUser = Integer.parseInt(sel);
21             productList();
22     }
23 }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : productList() 메서드 구현

예제 5-18 상품 목록을 출력하는 productList() 메서드

MyShop.java

```
01      public void productList() {
02          System.out.println(title + " : 상품 목록 - 상품 선택");
03          System.out.println("=====");
04          int i = 0;
05
06          // 등록된 상품 정보 출력
07          for(Product p : products) {
08              System.out.print "[" + i + " ]");
09              p.printDetail();
10              i++;
```

02. 쇼핑몰 프로그램 개발

```
11         }
12         System.out.println("[h]메인화면");
13         System.out.println("[c]체크아웃");
14         System.out.print("선택 : ");
15         String sel = scan.next();
16         System.out.println("## " + sel + "선택 ##");
17
18         // 선택한 메뉴에 따라 처리
19         switch(sel) {
20             case "h": start();break;
21             case "c": checkOut();break;
22             default:
23                 int psel = Integer.parseInt(sel);
24                 cart.add(products[psel]);
25                 productList();
26         }
27     }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : checkOut() 메서드 구현

예제 5-19 상품을 결제하는 checkOut() 메서드

MyShop.java

```
01      public void checkOut() {
02          System.out.println(title + " : 체크아웃");
03          System.out.println("=====");
04          int total = 0;
05          int i = 0;
06
07          // 장바구니에 등록된 상품 정보 출력
08          for(Product p : cart) {
09              System.out.printf("[%d]%(%)\n", i++, p.pname, p.price);
10              total = total + p.price;
11          }
12          System.out.println("=====");
13
14          // 선택한 사용자의 결제 방법 출력
```


02. 쇼핑몰 프로그램 개발

```
15         System.out.println("결제 방법: " + users[selUser].getPayType());
16
17         // 합계 출력
18         System.out.println("합계: " + total + " 원 입니다.");
19         System.out.println("[p]이전 , [q]결제 완료");
20         System.out.print("선택 : ");
21         String sel = scan.next();
22
23         // 선택된 메뉴에 따라 처리
24         switch(sel) {
25             case "q":
26                 System.out.println("## 결제가 완료 되었습니다. 종료합니다 ##");
27                 System.exit(0);break;
28             case "p": productList();break;
29             default:
30                 checkOut();
31         }
32     }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : checkOut() 메서드 구현

예제 5-20 MyShop 클래스의 전체 코드

MyShop.java

```
01 package javabook.ch5;
02
03 import java.util.ArrayList;
04 import java.util.Scanner;
05
06 // IShop 인터페이스를 구현한 MyShop 클래스
07 public class MyShop implements IShop {
08     // 등록 회원 정보 배열
09     User[] users = new User[2];
10
11     // 등록 상품 정보 배열
12     Product[] products = new Product[4];
13
14     // 상품을 추가할 수 있는 장바구니
15     ArrayList<Product> cart = new ArrayList<Product>();
```

02. 쇼핑몰 프로그램 개발

```
16
17 // 키보드 입력으로 문자열 입력받는 Scanner 객체 생성
18 Scanner scan = new Scanner(System.in);
19
20 // 선택한 사용자 index 보관
21 int selUser;
22
23 // 쇼핑몰 이름
24 String title;
25
26 public void setTitle(String title) {
27     this.title = title;
28 }
29
30 /**
31  * 프로그램 메인 시작 메서드
32  */
```

02. 쇼핑몰 프로그램 개발

```
33     public void start() {
34         System.out.println(title + " : 메인 화면 - 계정 선택");
35         System.out.println("=====");
36         int i = 0;
37
38         // 등록된 사용자 정보 출력
39         for(User u : users) {
40             System.out.printf("[%d]%(%)\n", i++, u.getName(), u.getPayType());
41         }
42
43         System.out.println("[x]종 료");
44         System.out.print("선택 : ");
45         String sel = scan.next();
46         System.out.println("## " + sel + "선택 ##");
47
48         // 선택한 메뉴에 따라 처리
49         switch(sel) {
```

02. 쇼핑몰 프로그램 개발

```
50         case "x": System.exit(0);break;
51         default:
52             selUser = Integer.parseInt(sel);
53             productList();
54     }
55 }
56
57 /**
58  * 상품 목록을 보고 상품을 선택해 장바구니에 넣는 메서드
59  */
60 public void productList() {
61     System.out.println(title + " : 상품 목록 - 상품 선택");
62     System.out.println("=====");
63     int i = 0;
64
65     // 등록된 상품 정보 출력
66     for(Product p : products) {
67         System.out.print "[" + i + " ]");
```

02. 쇼핑몰 프로그램 개발

```
68         p.printDetail();
69         i++;
70     }
71     System.out.println("[h]메인화면");
72     System.out.println("[c]체크아웃");
73     System.out.print("선택 : ");
74     String sel = scan.next();
75     System.out.println("## " + sel + "선택 ##");
76
77     // 선택한 메뉴에 따라 처리
78     switch(sel) {
79         case "h": start();break;
80         case "c": checkOut();break;
81         default:
82             int psel = Integer.parseInt(sel);
83             cart.add(products[psel]);
84             productList();
85     }
86 }
```

02. 쇼핑몰 프로그램 개발

```
87
88     /**
89     * 결제 진행을 하는 체크아웃 처리 메서드
90     */
91     public void checkOut() {
92         System.out.println(title + " : 체크아웃");
93         System.out.println("=====");
94         int total = 0;
95         int i = 0;
96
97         // 장바구니에 등록된 상품 정보 출력
98         for(Product p : cart) {
99             System.out.printf("[%d]%(%)\n", i++, p.pname, p.price);
100             total = total + p.price;
101         }
102         System.out.println("=====");
103     }
```

02. 쇼핑몰 프로그램 개발

```
104         // 선택한 사용자의 결제 방법 출력
105         System.out.println("결제 방법: " + users[selUser].getPayType());
106
107         // 합계 출력
108         System.out.println("합계: " + total + " 원 입니다.");
109         System.out.println("[p]이전 , [q]결제 완료");
110         System.out.print("선택 : ");
111         String sel = scan.next();
112
113         // 선택한 메뉴에 따라 처리
114         switch(sel) {
115             case "q":
116                 System.out.println("## 결제가 완료 되었습니다. 종료합니다 ##");
117                 System.exit(0);break;
118             case "p": productList();break;
119             default:
120                 checkOut();
121         }
122     }
```


02. 쇼핑몰 프로그램 개발

```
123
124     /**
125     * 프로그램에서 사용하는 예제 사용자 등록 메서드
126     */
127     public void genUser() {
128         User user = new User("홍길동", PayType.CARD);
129         users[0] = user;
130         user = new User("블로거", PayType.CASH);
131         users[1] = user;
132     }
133
134     /**
135     * 프로그램에서 사용하는 예제 상품 등록 메서드
136     */
```

02. 쇼핑몰 프로그램 개발

```
137     public void genProduct() {  
138         CellPhone cp = new CellPhone("갤럭시 노트5", 1000000, "SKT");  
139         products[0] = cp;  
140         cp = new CellPhone("애플 아이폰7", 980000, "KT");  
141         products[1] = cp;  
142         SmartTV st = new SmartTV("삼성 3D Smart TV", 5000000, "4K");  
143         products[2] = st;  
144         st = new SmartTV("LG Smart TV", 2500000, "Full HD");  
145         products[3] = st;  
146     }  
147 }
```

02. 쇼핑몰 프로그램 개발

■ 쇼핑몰 메인 클래스 구현 : 프로그램 실행 런처 클래스 구현

예제 5-21 쇼핑몰 프로그램을 실행하는 런처 클래스 구현하기

ShopLauncher.java

```
01 package javabook.ch5;
02
03 // 쇼핑몰을 실행하는 런처 프로그램,
04 // IShop 인터페이스를 구현하는 다른 쇼핑몰 클래스도 이곳에서 실행 가능하다.
05 public class ShopLauncher {
06
07     // 프로그램 메인
08     public static void main(String[] args) {
09         IShop shop = new MyShop();
10         shop.setTitle("MyShop");
11         shop.genUser();
12         shop.genProduct();
13         shop.start();
14     }
15 }
```