

(주의1) 프로젝트이름 형식 : 반_학번_이름 (예: A_20190000_김인하)

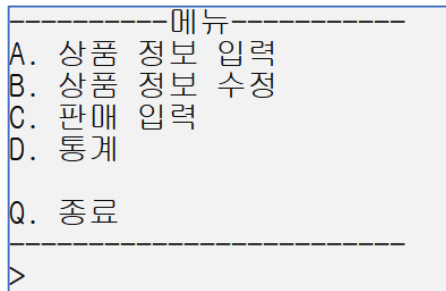
(주의2) 반복문 제어를 위한 변수를 제외하고 a, b, c와 같은 단순한 변수명 사용 금지.

(주의3) 모두 작성하지 못해도 됩니다. 제시한 내용을 순서대로 작성하세요.

(주의4) 잘못 입력 했을 때를 가정하지 않고, 올바르게 입력했음을 가정하고 진행한다.

(주의5) 화면 예제에서 노란색은 사용자 입력 / 초록색은 출력을 뜻한다.

프로그램 기본 화면 및 내용



The screenshot shows a menu interface with a title bar labeled '메뉴' (Menu). Below the title bar, there are five menu items: A. 상품 정보 입력 (Product Information Input), B. 상품 정보 수정 (Product Information Modification), C. 판매 입력 (Sales Input), D. 통계 (Statistics), and Q. 종료 (End). The items are listed vertically, separated by horizontal lines. At the bottom of the menu, there is a green prompt character '>'.

- A. 상품명, 단가, 초기수량(재고수량), 할인율을 입력하고 상품 정보 생성
- B. 상품의 현재 추가 재고수량(1)을 입력하거나, 할인율(2)을 수정한다.
- C. 등록된 상품을 판매하여 판매금액, 현재 재고수량을 조정
- D. 상품별 총 판매 수량, 총 판매액을 출력한다.
- Q. 프로그램을 종료한다.

각 입력 번호에 따라서 해당 항목을 실행한다.

Q 입력 시 프로그램을 종료한다.

그 외에는 계속 위 메뉴를 보여주어 정확한 입력을 받도록 한다.

[1]. 화면구성

메뉴	
A.	상품 정보 입력
B.	상품 정보 수정
C.	판매 입력
D.	통계
Q. 종료	

>

모든 작업이 끝나면 위 메뉴를 보여주고 '>'다음에 위 메뉴 번호를 입력 받아서 작업할 수 있도록 코드를 구성한다.

모든 메뉴는 작업이 끝나면 위 메뉴로 돌아간다.

[2] 클래스 Product 생성

- 프로퍼티

- 상품번호 Number (private set/public get)
- 상품명 Name (private set/public get)
- 재고수량 StockCount (private set/public get)
- 개당가격(단가) UnitPrice (public set/public get)
- 할인율 DiscountRate (public set/public get)

생성자 : 위 프로퍼티 항목을 매개변수로 받아 입력

- 메소드 (총 2개)

- 재고수량 증가 : 아래 [7]-4 에서 사용할

기본) 매개변수: 추가재고 수량, 반환값: 없음

1) 추가재고 수량을 현재 재고수량에 추가한다.

- 판매 : 아래 [9]-3 에서 사용할

기본)매개변수: 판매수량, 반환값: 총판매 금액

1) 판매수량을 현재 재고수량과 비교한다.

2)재고수량이 판매수량 이상 있으면 재고수량을 판매수량 만큼 줄이고

3)판매수량에 개당가격(단가)를 곱하여 총 판매금액을 계산하고 이를 반환한다.

- 단, 판매수량이 30개 이상이면 할인율을 적용한다.

- 총 판매금액은 정수형으로 반환해야 함

4)재고수량이 판매수량보다 적은 경우 "재고수량 없음" 으로 Exception을 발생시킨다.

[3] 클래스 Sale 생성

- 프로퍼티

- 판매번호 Number (private set/public get)
- 상품번호 ProdNumber (private set/public get)
- 판매일자 Date (private set/public get)
- 판매수량 Count (public set/public get)
- 총판매금액 Price (public set/public get)

생성자 : 위 프로퍼티 항목을 매개변수로 받아 입력

[4] 'Q' 입력 시

1. 현재 프로그램을 종료한다.

[5] 공통자료구조 생성

1. Product와 Sale 인스턴스를 저장할 수 있는 상품목록 / 판매목록 List 를 생성하여, Program에서 같이 관리할 수 있도록 한다.

[6] 'A' 입력 시

```
-----메뉴-----
A. 상품 정보 입력
B. 상품 정보 수정
C. 판매 입력
D. 통계
Q. 종료
-----
>a
상품명: 유리잔
단가: 1000
초기수량: 200
30개 이상 구매시 할인율(예: 10% -> 0.1 입력): 0.15
상품번호: 7 이름: 유리잔 단가: 1000 재고수량 200 할인율: 15%
```

1. 상품정보는 '상품명', '단가', '초기수량', '30개 이상 구매 시 할인율' 4가지를 받는다.
2. (1)에서 받은 정보와 상품번호를 가지고 Product 클래스로 인스턴스 생성한다.
 - 상품번호는 각자 알아서 룰을 정해서 할 것.
3. 생성한 인스턴스는 [5]에서 생성한 List에 추가한다.
4. 입력한 정보로 생성한 정보를 확인용으로 마지막에 출력한다.

[7] 'B' 입력 시 (1)

```
-----메뉴-----
A. 상품 정보 입력
B. 상품 정보 수정
C. 판매 입력
D. 통계
Q. 종료
-----
>b
상품명: 바지
현재 재고: 310
30개 이상 구매시 할인율: 20%
추가 입고 수량 입력(1) 할인율 수정(2): 1
추가 재고 수량: 20
현재 재고: 330
```

1. '상품명'을 입력 받아, 상품목록에서 해당 Product 인스턴스를 검색하고 '현재 재고', '30개 이상 구매시 할인율'을 출력한다.
2. (1) 수행 후 '추가 입고 수량 입력'을 하고 싶으면 '1'을 입력한다.
3. (2) 수행 후 '추가 재고 수량'을 입력한다.
4. 해당 Product의 '재고수량 증가' 메소드를 이용하여 재고수량을 증가한다.
5. 마지막엔 반영한 '현재 재고'를 출력한다.

[8] 'B' 입력 시 (2)

```
-----메뉴-----
A. 상품 정보 입력
B. 상품 정보 수정
C. 판매 입력
D. 통계

Q. 종료

>b
상품명: 바지
현재 재고: 330
30개 이상 구매시 할인율: 20%
추가 입고 수량 입력(1) 할인율 수정(2): 2
수정 할인율(예: 10% -> 0.1 입력): 0.3
변경 30개 이상 구매시 할인율: 30%
```

1. '상품명'을 입력 받아, 상품목록에서 해당 Product 인스턴스를 검색하고 '현재 재고', '30개 이상 구매시 할인율'을 출력한다.
2. (1) 수행 후 '할인율 수정'을 하고 싶으면 '2'를 입력한다.
3. (2) 수행 후 '수정 할인율'을 입력한다.
4. 해당 Product의 할인율 프로퍼티를 할인율을 변경한다.
5. 마지막엔 반영한 '변경 30개 이상 구매시 할인율'를 출력한다.

[9] 'C' 입력 시

```
-----메뉴-----
A. 상품 정보 입력
B. 상품 정보 수정
C. 판매 입력
D. 통계

Q. 종료

>c
상품명: 유리잔
현재 재고: 190
30개 이상 구매시 할인율: 15%
판매 수량: 40
판매번호: 6 상품명: 유리잔 판매일자: 2020-06-23 판매수량: 40 판매금액: 34000
```

1. '상품명'을 입력 받아, 상품목록에서 해당 Product 인스턴스를 검색하고 '현재 재고', '30개 이상 구매시 할인율'을 출력한다.
2. (1) 수행 후 '판매수량'을 현재 재고 내에서 입력한다.
3. 해당 Product 인스턴스의 재고수량을 판매수량 만큼 감소시키고, 판매수량에 따른 판매금액을 계산한다.
Product 클래스의 '판매' 메소드 이용
4. (1~3)에서 받은 정보와 계산한 결과, 판매번호를 가지고 Sale 클래스로 인스턴스 생성한다.
판매번호는 각자 알아서 룰을 정해서 할 것.
5. 생성한 인스턴스는 [5]에서 생성한 List에 추가한다.
6. 입력한 정보로 생성한 정보를 확인용으로 출력한다.

[10] 'D' 입력 시

-----메뉴-----				
A. 상품 정보 입력				
B. 상품 정보 수정				
C. 판매 입력				
D. 통계				
Q. 종료				

>d				
총수량:	0개	총판매액:	0원	[바지]
총수량:	100개	총판매액:	1020000원	[꽃지블라우스]
총수량:	0개	총판매액:	0원	[머리띠]
총수량:	20개	총판매액:	1100원	[그루]
총수량:	100개	총판매액:	1000원	[종이컵]
총수량:	0개	총판매액:	0원	[a]
총수량:	10개	총판매액:	100원	[장은]
총수량:	50개	총판매액:	44000원	[유리잔]

1. 모든 상품의 판매 '총수량'과 '총판매액', '상품명'을 출력한다.

2. 출력형식은 위와 동일하게 할 필요 없음

파일처리

사용하는 자료 파일은 C:드라이브에 Fin학번 폴더에 만든다. (예: C:\WFin20190000)

[11] [6]번을 완료한 후 아래와 같은 내용으로 'product.txt' 파일을 상품당 하나씩 저장(추가)한다.

```
000000,바지,010000,000300,00.20
000001,꽃지블라우스,012000,001100,00.15
000002,머리띠,001500,000100,00.11
000003,그루,000110,001180,00.05
000004,종이컵,000010,009900,00.00
```

000000,바지,010000,000300,00.20

(1) (2) (3) (4) (5)

(1) 상품번호 : 6자리로 표현

(2) 상품명 : 상품명

(3) 단가 : 6자리로 표현

(4) 현재재고수량 : 6자리로 표현

(5) 할인율 : 00.00 으로 표현 (code ex) (3.1456).ToString("00.00");

[12] [7]번을 완료한 후 아래와 같은 내용으로 'product.txt' 파일의 재고수량을 수정한다.

```
000000,바지,010000,000330,00.30
000001,꽃지블라우스,012000,001100,00.15
```

[13] [8]번을 완료한 후 아래와 같은 내용으로 'product.txt' 파일의 할인률을 수정한다.

```
000000,바지,010000,000330,00.30
000001,꽃지블라우스,012000,001100,00.15
```

[14] [9]번을 완료한 후 아래와 같은 내용으로 'sale.txt' 파일을 판매건당 하나씩 저장(추가)한다.

```
000000,000003,20200622,00000010,0000001100
000001,000004,20200622,00000100,0000001000
000002,000001,20200622,00000100,0001020000
```

000000,000003,20200622,00000010,0000001100

(1) (2) (3) (4) (5)

(1) 판매번호 : 6자리로 표현

(2) 상품번호 : 6자리로 표현

(3) 판매연월일 : yyyyMMdd로 표현

(4) 판매수량 : 8자리로 표현

(5) 총판매액 : 10자리로

[15] 프로그램 시작시 [5]에 'product.txt'와 'sale.txt'를 읽어서 복구할 수 있도록 한다.