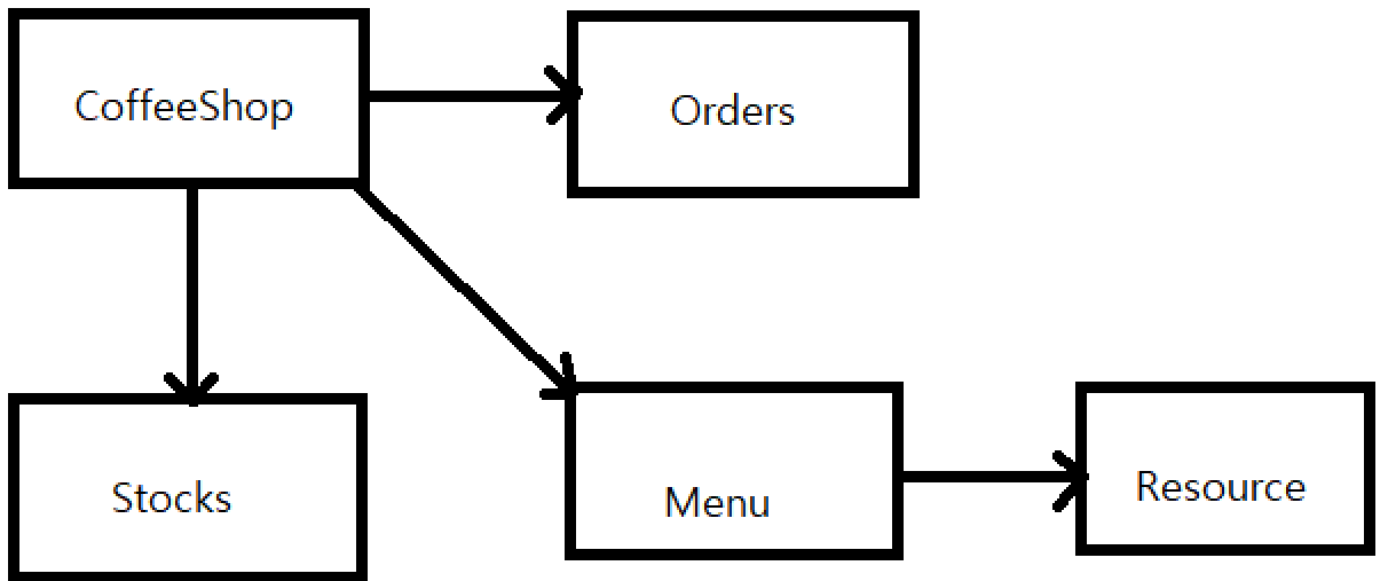


문제 1. 다음은 커피숍 주문 계산기를 만드는 프로그램이다. 아래의 순서대로 작성하라

<설계도>



CoffeeShop : 손님들의 주문을 받고 계산을 처리하는 클래스

Orders : 손님들의 주문에 대한 명세가 들어있는 클래스

Stocks : 커피숍의 재고가 들어있는 클래스

Menu : 커피숍의 메뉴 아이템과 가격이 들어있는 클래스

Resource : 소모되는 재료의 이름과 양이 들어있는 클래스

- CoffeeShopMain.java

```
public class CoffeeShopMain {  
    public static void main(String[] args) {  
        CoffeeShop coffee = new CoffeeShop();  
    }  
}
```

- CoffeeShop.java

```
import java.util.Scanner;  
  
public class CoffeeShop {  
    Scanner scan = new Scanner(System.in);  
    public CoffeeShop() {  
    }  
}
```

1) Main 문 실행 시 "어서오세요!구디 커피숍에 오신 것을 환영합니다" 라는 문구를 출력해라.

```
import java.util.Scanner;

public class CoffeeShop {

    Scanner scan = new Scanner(System.in);

    public CoffeeShop() {
        System.out.println("어서오세요 구디 커피숍에 오신것을 환영합니다.");
    }

}
```

실행결과)

```
<terminated> CoffeeShopMain [Java Application] C:\work\W
어서오세요 구디 커피숍에 오신것을 환영합니다.
```

2) 실행 시 아래와 같은 문구와 숫자를 넣어 출력하는 로직을 작성하라

```
public class CoffeeShop {

    Scanner scan = new Scanner(System.in);

    public CoffeeShop() {
        System.out.println("어서오세요 구디 커피숍에 오신것을 환영합니다.");
        System.out.println("어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)");
        int a = scan.nextInt();
        System.out.println(a);
    }

}
```

실행결과)

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
```

3) 위의 예제에서 숫자를 입력하면 각 숫자에 해당하는 출력을 하도록 만들어라.

```
public class CoffeeShop {

    Scanner scan = new Scanner(System.in);

    public CoffeeShop() {
        System.out.println("어서오세요 구디 커피숍에 오신것을 환영합니다.");
        System.out.println("어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)");
        int a = scan.nextInt();
        System.out.println(a);

        if(a == 1) {
            System.out.println("주문 메뉴를 선택하셨습니다");
            System.out.println("-----주문-----");
        }else if(a == 2) {
            System.out.println("재고 관리 메뉴를 선택하셨습니다");
            System.out.println("-----재고 관리-----");
        }else if(a == 3) {
            System.out.println("출력 메뉴를 선택하셨습니다");
            System.out.println("-----출력-----");
        }else if(a == 4) {
            System.out.println("시스템을 종료합니다");
        }
    }
}
```

실행결과)

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
2
2
재고 관리 메뉴를 선택하셨습니다
-----재고 관리-----
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
3
3
출력 메뉴를 선택하셨습니다
-----출력-----
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
4
4
시스템을 종료합니다
```

4) 로직을 아래와 같이 만들고 4번을 누르면 프로그램을 종료하는 로직을 작성하세요

```
public CoffeeShop() {  
  
    while(true) {  
        System.out.println();  
        System.out.println();  
        System.out.println("어서오세요 구디 커피숍에 오신것을 환영합니다.");  
        System.out.println("어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)");  
        int a = scan.nextInt();  
        System.out.println(a);  
  
        if(a == 1) {  
            System.out.println("주문 메뉴를 선택하셨습니다");  
            System.out.println("-----주문-----");  
            for (Menu menu : menus) {  
                System.out.println(menu);  
            }  
        }else if(a == 2) {  
            System.out.println("재고 관리 메뉴를 선택하셨습니다");  
            System.out.println("-----재고 관리-----");  
        }else if(a == 3) {  
            System.out.println("출력 메뉴를 선택하셨습니다");  
            System.out.println("-----출력-----");  
        }else if(a == 4) {  
            System.out.println("시스템을 종료합니다");  
            break;  
        }  
    }  
}
```

실행결과)

```
어서오세요 구디 커피숍에 오신것을 환영합니다.  
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)  
4  
4  
시스템을 종료합니다
```

5. Menu 클래스를 만들고 다음과 같이 작성하라

```
public class Menu {
    private String name;
    private int price;

    public Menu(String name, int price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public int getPrice() {
        return price;
    }

    public String toString() {
        return "[이름=" + name + ", 가격=" + price + "]";
    }
}
```

6. CoffeeShop 위쪽에 다음과 같은 필드를 만들고 생성자에 다음과 같은 메뉴를 추가하라

```
public class CoffeeShop {

    Scanner scan = new Scanner(System.in);
    ArrayList<Menu> menus = new ArrayList<Menu>();

    public CoffeeShop() {

        menus.add(new Menu("아메리카노", 2000));
        menus.add(new Menu("카페라떼", 2500));
        menus.add(new Menu("카푸치노", 3000));

        while(true) {
            System.out.println();
            System.out.println();
        }
    }
}
```

7. CoffeeShop 클래스에 setOrder 라는 메서드를 추가한다.

```
public void setOrder() {

}
```

8. setOrder에 menus에 있는 메뉴들의 목록을 출력하는 로직을 작성하라.

```
public void setOrder() {
    for (int i = 0; i < menus.size(); i++) {
        System.out.println(i+":"+menus.get(i));
    }
}
```

9. 입력창에서 1번을 누를 시 다음과 같은 화면이 나오도록 setOrder와 CoffeeShop의 생성자 함수를 활용하여 작성하라.

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
```

10. 다음과 같이 Orders 클래스를 만들고 총 금액을 구하는 로직을 다음과 같이 만들어라.

```
public class Orders {
    private ArrayList<Menu> menus = new ArrayList<Menu>();
    private ArrayList<Integer> counts = new ArrayList<Integer>();

    public void setMenu(Menu menu) {
        this.menus.add(menu);
    }
    public void setCount(Integer count) {
        this.counts.add(count);
    }

    // 총 금액을 구하는 로직
    public int getTotalPrice() {
        int total = 0;
        for (int i = 0; i < menus.size(); i++) {
            total += menus.get(i).getPrice() * counts.get(i);
        }
        return total;
    }
}
```

11. CoffeeShop의 setOrder를 아래 빈칸에 만들어 다음과 같은 내용이 출력되도록 만들어라

```
public void setOrder() {
    Orders orders = new Orders();
    for (int i = 0; i < menus.size(); i++) {
        System.out.println(i+": "+menus.get(i));
    }
    System.out.println("몇번 메뉴를 주문하시겠습니까?");
    int menuNum = scan.nextInt();
    System.out.println("수량은 얼마나 주문하시겠습니까?");
    int count = scan.nextInt();
    orders.setMenu(menus.get(menuNum));
    orders.setCount(count);
    System.out.println("총 금액은 "+orders.getTotalPrice()+" 입니다.");
}
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
0
수량은 얼마나 주문하시겠습니까?
2
총 금액은 4000 입니다.
```


12. 위의 내용을 더 개량하여 여러 개의 주문을 받을 수 있도록 만들어라.

```
public void setOrder() {
    Orders orders = new Orders();
    while(true) {
        for (int i = 0; i < menus.size(); i++) {
            System.out.println(i+":"+menus.get(i));
        }
        System.out.println("몇번 메뉴를 주문하시겠습니까?");
        int menuNum = scan.nextInt();
        System.out.println("수량은 얼마나 주문하시겠습니까?");
        int count = scan.nextInt();
        orders.setMenu(menus.get(menuNum));
        orders.setCount(count);
        System.out.println("종료하시겠습니까?(1.yes, 2.no)");
        int choose = scan.nextInt();
        if(choose == 1) {
            break;
        }
    }
    System.out.println("총 금액은 "+orders.getTotalPrice()+" 입니다.");
}
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
0
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
2
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
1
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
1
총 금액은 9000 입니다.
```


13. Stock 클래스를 다음과 같이 만들어라

```
public class Stocks {
    private String source;
    private int amount;

    public Stocks(String source, int amount) {
        this.source = source;
        this.amount = amount;
    }

    public String getSource() {
        return source;
    }

    public void plusAmount(int amount) {
        this.amount += amount;
    }

    public void minusAmount(int amount) {
        this.amount -= amount;
    }

    @Override
    public String toString() {
        return "[재료명=" + source + ", 양=" + amount + "]";
    }
}
```

14. CoffeeShop 생성자 함수와 필드에 다음과 같이 추가하라

```
public class CoffeeShop {

    Scanner scan = new Scanner(System.in);
    ArrayList<Menu> menus = new ArrayList<Menu>();
    ArrayList<Stocks> stocks = new ArrayList<Stocks>();

    public CoffeeShop() {

        // 메뉴
        menus.add(new Menu("아메리카노", 2000));
        menus.add(new Menu("카페라떼", 2500));
        menus.add(new Menu("카푸치노", 3000));

        // 재고
        stocks.add(new Stocks("커피콩", 50));
        stocks.add(new Stocks("시럽", 80));
        stocks.add(new Stocks("우유", 60));

        while(true) {
            System.out.println();
            System.out.println();
        }
    }
}
```

15. CoffeeShop 클래스에 getStocks라는 메서드를 추가한다.

```
public void getStocks() {  
  
}
```

16. 위의 getStock을 완성하여 다음과 같이 돌아가는 로직을 작성하라.(2번을 선택 시 재고 관련 로그가 떠야 함)

```
어서오세요 구디 커피숍에 오신것을 환영합니다.  
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)  
2  
2  
재고 관리 메뉴를 선택하셨습니다  
-----재고 관리-----  
[재료명=커피콩, 양=50]  
[재료명=시럽, 양=80]  
[재료명=우유, 양=60]
```

17. Resource라는 클래스를 아래와 같이 만들어라.

```
public class Resource {  
    // 재료명  
    private String name;  
    // 소모되는 양  
    private int consumeAmount;  
  
    public Resource(String name, int consumeAmount) {  
        this.name = name;  
        this.consumeAmount = consumeAmount;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getConsumeAmount() {  
        return consumeAmount;  
    }  
}
```

18. Menu 클래스에 다음과 같이 로직을 수정하고 CoffeeShop 클래스를 아래와 같이 작성한다.

```
public class Menu {
    private String name;
    private int price;
    private ArrayList<Resource> consumeList;

    public Menu(String name, int price, ArrayList<Resource> consumeList) {
        this.name = name;
        this.price = price;
        this.consumeList = consumeList;
    }

    public String getName() {
        return name;
    }

    public int getPrice() {
        return price;
    }

    public ArrayList<Resource> getConsumeList() {
        return consumeList;
    }

    public String toString() {
        return "[이름=" + name + ", 가격=" + price + "]";
    }
}
```

```
public class CoffeeShop {

    Scanner scan = new Scanner(System.in);
    ArrayList<Menu> menus = new ArrayList<Menu>();
    ArrayList<Stocks> stocks = new ArrayList<Stocks>();

    public CoffeeShop() {

        // 메뉴
        ArrayList<Resource> menu1Resource = new ArrayList<Resource>();
        menu1Resource.add(new Resource("커피콩", 2));

        menus.add(new Menu("아메리카노", 2000, menu1Resource));

        ArrayList<Resource> menu2Resource = new ArrayList<Resource>();
        menu2Resource.add(new Resource("커피콩", 2));
        menu2Resource.add(new Resource("우유", 2));

        menus.add(new Menu("카페라떼", 2500, menu2Resource));

        ArrayList<Resource> menu3Resource = new ArrayList<Resource>();
        menu2Resource.add(new Resource("커피콩", 2));
        menu2Resource.add(new Resource("시럽", 1));
        menus.add(new Menu("카푸치노", 3000, menu3Resource));

        // 재고
        stocks.add(new Stocks("커피콩", 50));
        stocks.add(new Stocks("시럽", 80));
        stocks.add(new Stocks("우유", 60));

        while(true) {
            System.out.println();
            System.out.println();
        }
    }
}
```

19. setOrder에 주문과 수량을 선택하는 순간 Stocks에 있던 재료가 소진되는 로직을 만들어라.(Menu에 있는 ConsumeList에 있는 Resource의 이름과 소모되는 양을 참조해서 Stocks에 있는 재료의 양을 소모시켜라)

*참고 : String 의 비교는 String == String 이 아니라 String.equals(String)으로 하는 편이 좋다.

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
0
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
2
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
1
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
1
총 금액은 9000 입니다.

어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
2
2
재고 관리 메뉴를 선택하셨습니다
-----재고 관리-----
[재료명=커피콩, 양=38]
[재료명=시럽, 양=78]
[재료명=우유, 양=56]
```

20. CoffeeShop의 getStocks의 메소드를 수정하여 다음과 같이 출력되도록 만들어라.

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
2
2
재고 관리 메뉴를 선택하셨습니다
-----재고 관리-----
0: [재료명=커피콩, 양=50]
1: [재료명=시럽, 양=80]
2: [재료명=우유, 양=60]
재고를 채우시겠습니까? (1.yes 2.no)
```

21. CoffeeShop의 getStock의 메소드를 다음과 같이 수정하여 재고를 채우는 로직을 작성하라.

```
public void getStocks() {
    while(true) {
        for (int i = 0; i < stocks.size(); i++) {
            System.out.println(i+":"+stocks.get(i));
        }
        System.out.println("재고를 채우시겠습니까? (1.yes 2.no) ");
        int chol = scan.nextInt();
        if(chol == 2)break;
        System.out.println("몇번째 재고를 채우시겠습니까?");
        int cho2 = scan.nextInt();
        System.out.println("얼마만큼 재고를 채우시겠습니까?");
        int cho3 = scan.nextInt();

        stocks.get(cho2).plusAmount(cho3);
    }
}
```

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
2
2
재고 관리 메뉴를 선택하셨습니다
-----재고 관리-----
0: [재료명=커피콩, 양=50]
1: [재료명=시럽, 양=80]
2: [재료명=우유, 양=60]
재고를 채우시겠습니까? (1.yes 2.no)
1
몇번째 재고를 채우시겠습니까?
0
얼마만큼 재고를 채우시겠습니까?
30
0: [재료명=커피콩, 양=80]
1: [재료명=시럽, 양=80]
2: [재료명=우유, 양=60]
재고를 채우시겠습니까? (1.yes 2.no)
2
```


22. 다음과 같은 필드를 CoffeeShop 클래스에 추가하라

```
public class CoffeeShop {  
  
    Scanner scan = new Scanner(System.in);  
    ArrayList<Menu> menus = new ArrayList<Menu>();  
    ArrayList<Stocks> stocks = new ArrayList<Stocks>();  
    private int totalEarn = 0;  
    private int orderCount = 0;  
}
```

23. CoffeeShop에 있는 setOrder 메소드 로직을 수정하여 주문 총액과 주문량을 각각 totalEarn과 orderCount 필드에 총 주문 금액과 카운트를 가산하라.

```
public void setOrder() {  
    Orders orders = new Orders();  
    while(true) {  
        for (int i = 0; i < menus.size(); i++) {  
            System.out.println(i+":"+menus.get(i));  
        }  
        System.out.println("몇번 메뉴를 주문하시겠습니까?");  
        int menuNum = scan.nextInt();  
        System.out.println("수량은 얼마나 주문하시겠습니까?");  
        int count = scan.nextInt();  
        orders.setMenu(menus.get(menuNum));  
        orders.setCount(count);  
  
        // 소모 로직  
        for (Resource consume : menus.get(menuNum).getConsumeList()) {  
            for (Stocks stock : stocks) {  
                if (consume.getName().equals(stock.getSource())) {  
                    stock.minusAmount(consume.getConsumeAmount() * count);  
                }  
            }  
        }  
  
        System.out.println("종료하시겠습니까? (1.yes, 2.no)");  
        int choose = scan.nextInt();  
        if (choose == 1) {  
            break;  
        }  
    }  
    System.out.println("총 금액은 "+orders.getTotalPrice()+" 입니다.");  
    this.totalEarn += orders.getTotalPrice();  
    this.orderCount++;  
}
```


24. 3번을 클릭 시 출력을 다음과 같이 나오도록 작성하라

```
어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
1
1
주문 메뉴를 선택하셨습니다
-----주문-----
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
0
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
2
0: [이름=아메리카노, 가격=2000]
1: [이름=카페라떼, 가격=2500]
2: [이름=카푸치노, 가격=3000]
몇번 메뉴를 주문하시겠습니까?
1
수량은 얼마나 주문하시겠습니까?
2
종료하시겠습니까? (1.yes, 2.no)
1
총 금액은 9000 입니다.

어서오세요 구디 커피숍에 오신것을 환영합니다.
어떤 메뉴를 사용하시겠습니까? (1.주문 2.재고 3.출력 4.종료)
3
3
출력 메뉴를 선택하셨습니다
-----출력-----
총 수입 : 9000
총 주문량 : 1
```