

```
//main()에서는 order() 외에는 호출하지 마시오.  
//변수선언은 꼭 private 하지 않아도 됩니다.  
//메소드 위에 주석을 달아지만,  
//메소드를 수정 및 삭제하지 않는 한에서 여러분이 흐름상  
//맞다고 생각하는 대로 코드 작성해도 됩니다.
```

```
//결과 : Seller는 Buyer가 산 과일 개수 만큼  
//      과일 개수 줄어들과 수입이 증가되며,  
//      Buyer는 보유 금액이 감소하게 된다.
```

```
public class MainTest {
```

```
    public static void main(String[] args) {
```

```
        Seller seller01 = new Seller(10, 12);  
        Seller seller02 = new Seller(15, 18);  
        //Seller(보유사과개수, 보유바나나개수) 객체 2개 생성 - ok
```

```
        //Buyer(과일 살수 있는 총 보유 금액) 객체 3개 생성 - ok  
        Buyer buyer01 = new Buyer(2000);  
        Buyer buyer02 = new Buyer(1500);  
        Buyer buyer03 = new Buyer(1000);
```

```
        //Buyer 형 모든 객체의 order 메소드 호출  
        buyer01.order(seller01, 10, 12);  
        buyer02.order(seller02, 10, 10);  
        buyer03.order(seller02, 5, 8);
```

```
        //위에 만든 모든 객체의 객체 변수 값 출력하기
```

```

        System.out.println("=====seller=====
");

        System.out.println("=====seller01=====
==");

        System.out.println("seller01.appleCount      :
"+seller01.appleCount);

        System.out.println("seller01.bananaCount      :
"+seller01.bananaCount);

        System.out.println("seller01.sellerMoney      :
"+seller01.sellerMoney);


        System.out.println("=====seller02=====
==");

        System.out.println("seller02.appleCount      :
"+seller02.appleCount);

        System.out.println("seller02.bananaCount      :
"+seller02.bananaCount);

        System.out.println("seller02.sellerMoney      :
"+seller02.sellerMoney);


        System.out.println("=====buyer=====
");

        System.out.println("buyer01.buyerMoney      :
"+buyer01.buyerMoney);

        System.out.println("buyer02.buyerMoney      :
"+buyer02.buyerMoney);

        System.out.println("buyer03.buyerMoney      :
"+buyer03.buyerMoney);

    }

}

```

```
class Apple {  
    //사과 1개당 가격 70원  
    static int price;  
    static {  
        price = 70;  
    }  
}
```

```
class Banana {  
    //바나나 1개당 가격 50원  
    static int price;  
    static {  
        price = 50;  
    }  
}
```

```
class Seller {  
  
    //사과개수 변수 선언  
    int appleCount;  
    //바나나개수 변수 선언  
    int bananaCount;  
    //수입 변수 선언()  
    int sellerMoney;  
  
    Seller(int appleCount, int bananaCount){  
        this.appleCount = appleCount;  
        this.bananaCount = bananaCount;  
    }  
}
```

```

    }

    //주문 개수만큼 과일을 파는 메소드구현(과일 개수가 줄겠지?),
    //과일 총 판매금액을 리턴해준다.
    public int sell(int appleCount, int bananaCount) {
        this.appleCount -= appleCount;
        this.bananaCount -= bananaCount;

        int total = (appleCount*Apple.price) +
(bananaCount*Banana.price);

        return total;
    }

    //구매자로 부터 받은 돈을 내 수입으로 포함하는 메소드
    public void takeMoney(int total) {
        sellerMoney += total;
    }
}

class Buyer {
    int buyerMoney;

    //구매자가 보유하고 있는 금액 변수 선언 - ok

    Buyer(int buyerMoney){
        this.buyerMoney = buyerMoney;
    }

    //원하는 Seller 객체에게 사고 싶은
    //원하는 Seller 객체에게 사고 싶은 사과 개수,
    //바나나 개수를 판매하라고 하고
    //구매액을 받아오면 바로 금액을 지급한다.

```

```
//지급하는 행위는 giveMoney()을 호출한다.

public void order(Seller seller, int appleCount, int bananaCount) {
    int total = seller.sell(appleCount, bananaCount);
    giveMoney(seller, total);
}

//나의 보유 금액을 줄이고
//Seller에게 돈을 준다.
public void giveMoney(Seller seller, int total) {
    this.buyerMoney -= total;
    seller.takeMoney(total);
}
}
```